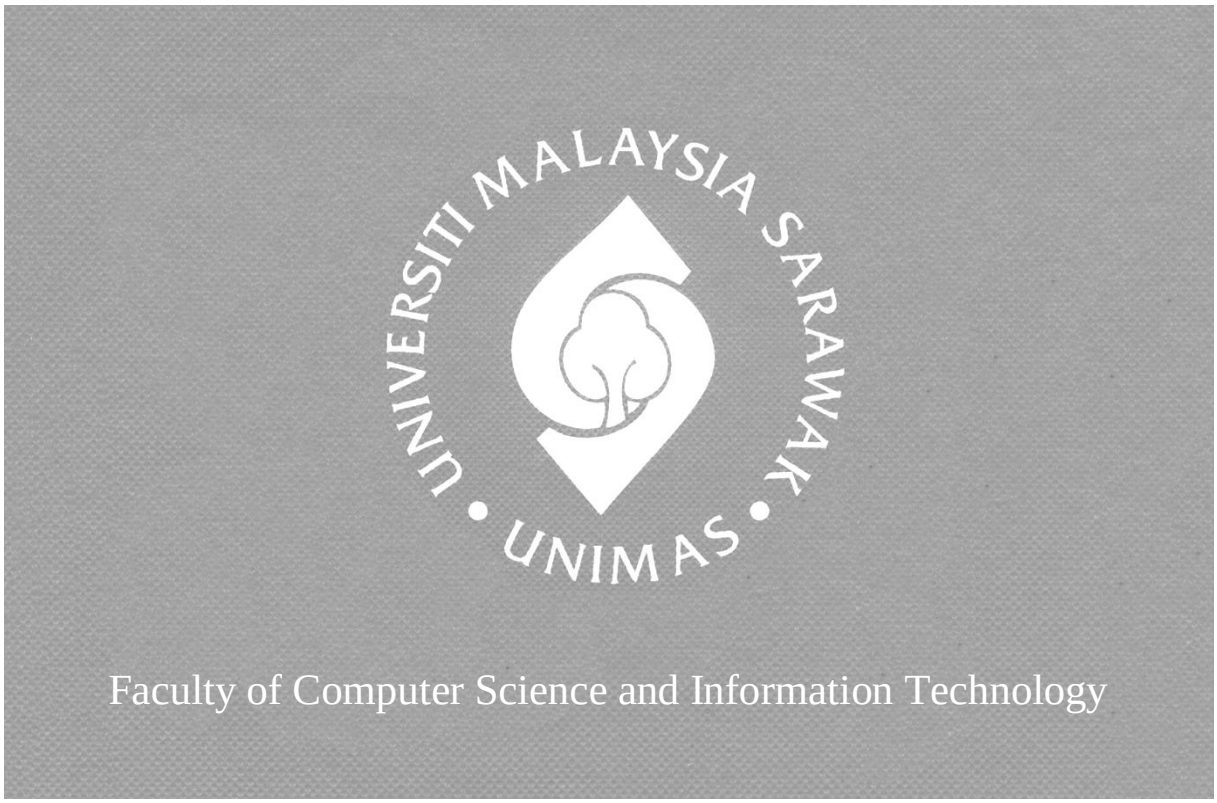


Phishdentity: Leverage Website Favicon to Offset Phishing Website

Jeffrey Choo Soon Fatt

Master of Science
(Computer Science)
2015

Phishdentity: Leverage Website Favicon to Offset Phishing Website

Jeffrey Choo Soon Fatt

A thesis submitted
in fulfillment of the requirements for the degree of
Master of Science

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF TERMS

AIWL	Automated Individual White-List
API	Application Program Interface
APWG	Anti-Phishing Working Group
CANTINA	Carnegie Mellon Anti-Phishing and Network Analysis Tool
CBIR	Content-Based Image Retrieval
DNS	Domain Name System
DOM	Document Object Model
FN	False Negative
FP	False Positive
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
IM	Instant Messaging
IP	Internet Protocol
ISP	Internet Service Provider
KNN	K-Nearest Neighbor
LR	Logistic Regression
LUI	Login User Interface
LDA	Linear Discriminant Analysis
MODI	Microsoft Office Document Imaging
NB	Naïve Bayes
NPO	Nonprofit Organization
OCR	Optical Character Recognition
RSA	Rivest-Shamir-Adleman
SIFT	Scale-Invariant Feature Transform
SLD	Second-Level Domain
SMS	Short Message Service
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TLD	Top-Level Domain

TN	True Negative
TP	True Positive
URL	Uniform Resource Locator
UT	Unique Term
W3	World Wide Web
W3C	World Wide Web Consortium
WHOIS	Who Is
WOT	Web Of Trust
XSS	Cross-Site Scripting

ABSTRACT

Phishing attack is a cybercrime which will result in severe financial losses to consumers and entrepreneurs. Typically, the phishers are fond of using fuzzy techniques during the creation of phishing websites. They obfuscate the victims by mimicking the appearance and content of the legitimate website. In addition, most of the websites are susceptible to the threat of phishing attacks, including financial institutions, social networks, e-commerce, airline websites and others. Phishers can easily earn the trust of the victim by impersonating as a consultant in the travel agency, booking flights and hotel reservations. Therefore, it is important to establish an intelligent gateway for browsers that can protect internet users from visiting malicious websites. In this thesis, we proposed an approach which is based on the website favicon to uncover the hidden identity of a website. We employ Google search by image engine to obtain the search results specific to the website favicon. Then, we perform feature extraction based on the search results to retrieve the website identity. Our identity retrieval technique involves an effective mathematical model in which it could be used to assist in retrieving the right identity from the many entries of the search results. In addition, we also proposed additional approach which is based on the URL to examine the legitimacy of a website. More precisely, we study the URL based on the lexical features, host-based features and domain features. Additional approach is very useful when the website under examination does not have a favicon. We have collected a total of 500 phishing websites from PhishTank and 500 of the legitimate websites from Alexa Top 500 Global Websites to verify the effectiveness of this approach. From the experimental results, our proposed technique has achieved 97.4% true positive with only 5.4% false positive. After combining with additional approach, our proposed technique is able to improve the false positives to 2.2%, while slightly reducing the accuracy of classifying phishing websites where we have achieved 97% true positive.

ABSTRAK

Serangan *phishing* adalah jenayah siber yang boleh mengakibatkan kerugian kewangan yang teruk kepada pengguna dan usahawan. Biasanya, *phisher* gemar menggunakan teknik kabur semasa penciptaan laman sesawang *phishing*. Mereka mengelirukan mangsa dengan meniru rupa dan kandungan laman sesawang yang sah. Selain itu, sebahagian besar daripada laman sesawang terdedah kepada ancaman serangan *phishing*, termasuk institusi kewangan, rangkaian sosial, e-dagang, laman sesawang syarikat penerbangan dan lain-lain. *Phisher* boleh mendapatkan kepercayaan mangsa dengan mudah melalui penyamaran sebagai seorang perunding di agensi pelancongan, tempahan penerbangan dan tempahan hotel. Oleh itu, adalah penting untuk menubuhkan sebuah gerbang pintar untuk pelayar melindungi pengguna internet melayari laman sesawang yang berniat jahat. Dalam tesis ini, kami mencadangkan satu pendekatan yang berasaskan *favicon* laman sesawang untuk membongkar identiti yang tersembunyi dalam sebuah laman sesawang. Kami menggunakan enjin carian Google dengan gambar untuk mendapatkan hasil carian khusus kepada *favicon* laman sesawang. Kemudian, kami melakukan pengekstrakan ciri berdasarkan kepada keputusan carian untuk mendapatkan identiti laman sesawang. Teknik identiti dapatan semula kami melibatkan model matematik di mana ia boleh digunakan untuk membantu dalam mendapatkan semula identiti yang betul daripada banyak penyertaan daripada hasil carian. Selain itu, kami juga telah mencadangkan ciri-ciri tambahan yang berasaskan URL untuk memeriksa kesahihan laman sesawang. Lebih tepat lagi, kita mengkaji URL berdasarkan ciri leksikal, ciri-ciri berasaskan tuan rumah dan ciri domain. Ciri-ciri tambahan adalah sangat berguna apabila laman sesawang di bawah pemeriksaan tidak mempunyai *favicon*. Kami telah mengumpulkan sejumlah 500 laman sesawang *phishing* dari PhishTank dan 500 laman sesawang yang sah dari Alexa Top 500 Global Website untuk mengesahkan keberkesanan pendekatan ini. Daripada keputusan ujikaji, teknik yang kami cadangkan dapat mencapai 97.4% positif benar dengan hanya 5.4% positif palsu. Selepas menggabungkan dengan ciri-ciri tambahan, teknik yang kami cadangkan dapat meningkatkan positif palsu kepada 2.2%, manakala mengurangkan dengan sedikit ketepatan mengklasifikasikan laman *phishing* di mana kita telah mencapai 97% positif benar.

LIST OF PUBLICATIONS

1. J. S. F. Choo, K. L. Chiew, and S. N. Sze. An initiative approach to prevent phishing attacks on tourism network. *Proceedings of the 12th Asia Pacific Forum for Graduate Students' Research in Tourism – Challenging Conventions in Research*, 2013.
2. J. S. F. Choo, K. L. Chiew, and S. N. Sze. Phishdentity: Leverage website favicon to offset polymorphic phishing website. *Proceedings of the 9th International Conference on Availability, Reliability and Security (ARES)*, pages 114 – 119, 2014.
3. J. S. F. Choo, K. L. Chiew and S. N. Sze. Phishdentity: Leverage website favicon to offset phishing websites. *International Journal of Computers and Security*, 2015. [Submitted for review]

ACKNOWLEDGEMENTS

This thesis would not have been possible without the assistance from many people who have given their support directly or indirectly. First and foremost, I would like to express my sincere gratitude to my main supervisor, Dr. Chiew Kang Leng for his expertise, support, and encouragement throughout the study. His patient instruction encouraged me to think in a more profound and pervasive way. I also want to give special thanks to my co-supervisor, Dr. Sze San Nah who always inspired me to undertake a thorough and in-depth analysis on the research. I would also like to thank my colleagues for so many great and nice discussions, which made my life here full of laughter. Last but not least, I am deeply grateful to my parents and siblings for their endless love, understanding, support and encouragement, which has accompanied me throughout my life.

The funding for this project is made possible through the research grant obtained from the Ministry of Higher Education, Malaysia under the Long Term Research Grant Scheme 2011 [LRGS grant no: JPT.S (BPKI)2000/09/01/015Jld.4(67)].

Chapter 1

Introduction

The advancement of information technology has brought new ways of interacting with people. In particular, the internet has opened up many opportunities for businesses to create new sales. However, this convenience also attracts the people with bad intention to take advantage of the shortcomings of the internet for illegal activities. Without proper protection, internet users will be exposed to different types of online fraud, such as phishing. As a result, internet users may suffer financial loss or disclosure of personal information to fraudsters. Eventually, internet users will lose confidence in using the internet and it will inhibit the growth in online business.

1.1. Background of Phishing

Phishing is defined as an act to deceive the recipients through a legitimate-looking email or a website in order to earn the trust and confidence of internet users to divulge their personal and financial information [9, 20, and 29]. With the advancement of information technology, many business agencies (i.e., tourism, hotel, airline, etc.) are able to take advantage of e-commerce, electronic payment, and social networking technologies into their business in order to increase sales. But, this creates an opportunity for phishers to masquerade into different types of services such as financial institutions, social networking, and e-commerce websites to gain illegal profits. Anti-Phishing Working Group (APWG) reported a total of 128,378 unique phishing websites detected in its second quarter of 2014 phishing activity trends report [1]. The report showed evidence that phishing activities are on the rise which revealed that existing anti-phishing technology is unable to fight phishing attacks efficiently.

The most common way to launch a phishing attack is through a combination of internet content from multiple sources or domain name confusion. Phishing websites can be done relatively quickly and requires little effort. This is because phishers simply clone the entire

website with some modifications in the input tags to collect personal information. The time of this process can be shortened by utilizing phishing kits [5], which are available on the black market. In addition, advancement of information technology helps phishers to develop more sophisticated phishing techniques to avert phishing detectors. Figure 1.1 shows an example of a phishing website masquerading as PayPal. There are two flaws identified in the address bar (as indicated by the red line box in Figure 1.1):

- The domain name is completely different than the original.
- It obfuscates the URL with HTTPS as part of the URL.

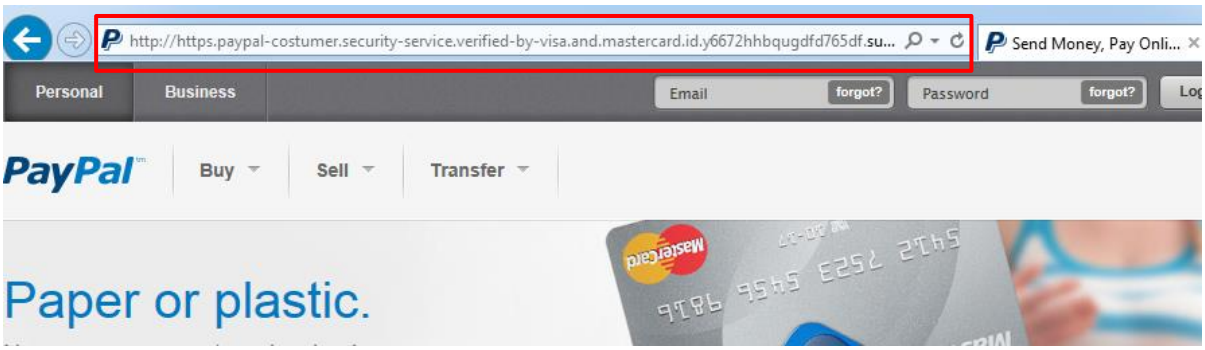


Figure 1.1: Example of a phishing website.

Phishers are highly in favor of visual deception techniques to fool potential victims. It manipulates the perception of the victim through common images shown on the website. A sophisticated phishing websites usually will adopt iconic images of the targeted website including the website favicons with distinctive contents. More specifically, phishers have the ability to reproduce similar targeted websites by adjusting the structure of HTML, images, flash objects, or some other dynamic components. In other words, sophisticated phishing attacks can cause phishing detectors to produce false negative results. This course of action is known as a phishing page polymorphism.

1.2. Motivation

Since the advent of phishing in cybercrime over the past decade, it has caused tremendous financial losses to internet consumers and online businesses [2 – 3]. It also tarnishes the reputation and trust of the targeted company. If there is no countermeasure to prevent the occurrence of phishing attacks, then there will be more internet consumers become the

victims of phishing. Eventually, phishing will affect the economic growth of a country. Government and non-profit anti-phishing organizations provide many channels to help educate internet users when browsing the internet, especially when dealing with money transactions on the internet. However, internet users are still falling into the trap of phishing. In other words, alternative approaches need to be developed to detect and prevent phishing attacks. As a result, the computational approach is introduced to study the behavior of phishing and propose countermeasures to detect phishing activity. While there are many anti-phishing solutions available in the market, but none of the solutions are 100 percent effective without incurring false alarms to the targeted company. In addition, advances in information technology have provided opportunities for phishers to study the mechanism of anti-phishing and produce high profile phishing attacks [4].

Phishing activity trends have not diminished despite many efforts from different bodies. The severity of these attacks can be seen from the reports published as follows:

- According to the reports published [1 – 4], the statistics of phishing incidents around the world has increased dramatically. In Malaysia alone [4], a total of 1033 unique phishing websites are reported on the fourth quarter of 2013. It is a 7.4% increment compared to the third quarter of 2013.
- According to a report conducted by RSA [2], the Anti-Fraud Command Center has estimated that in the first half of 2012, the potential loss in global organization committed solely by phishing is $687 million.

In summary, the study of new anti-phishing solution is necessary to compensate for weaknesses in existing anti-phishing solutions and to detect undiscovered phishing techniques.

1.3. Case Study Scenario

Figure 1.2 shows the flow of phishing attacks. The attacks consist of five steps as follows:

- Step 1: Phisher identifies potential fraud from available resources on the internet. Phisher send fake email to thousands of victims by modifying the content and header of email.

- Step 2: The victims receive the email and they have no doubt (possibly careless or limited knowledge in email header structure) about the content and address of email sender.
- Step 3: The victims respond to the email and open up the link in which it directs the victims to a fake website. Then, the victims submit their confidential data by logging to the fake website.
- Step 4: Phisher reviews the victims' data and takes advantage of the leaked information. Phisher gains access to the victims' respective bank website by using the stolen data.
- Step 5: Phisher successfully transferred the money from the victims account into his own bank account.

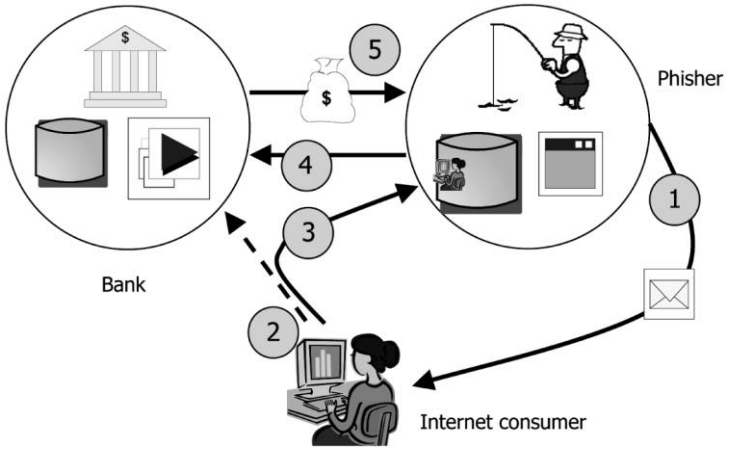


Figure 1.2: Example of a phisher masquerades as a bank agent to deceive consumers.[1]

It is nearly impossible to predict the occurrence of phishing attacks. For example, a tourist who wants to book a hotel and a flight ticket over the internet make payments by using internet banking has a very high possibility to be targeted by phishers. If he has limited knowledge of phishing attributes or inability to distinguish between a rogue website and a legitimate website, then it is likely that he will suffer a significant financial loss.

[1] Reprinted from *Modern Online Banking Cyber Crime – InfoSec Institute*, by P. Paganini, 2013, Retrieved 21 Dec 2013, from http://resources.infosecinstitute.com/modern-online-banking-cyber-crime/. Copyright 2013 by InfoSec Institute. Reprinted with permission.

1.4. Research Problems

- Existing textual-based anti-phishing solutions are depending on the content of a webpage to classify the legitimacy of a website. However, these solutions are incompetent to classify image-based phishing website. Phisher can replace the textual contents with images to evade phishing detectors. Hence, more victims will fall into image-based phishing attacks.
- Some phishers create phishing website that is visually dissimilar (i.e., webpage layout) to the legitimate website to phish potential victims. They preserve iconic images from legitimate websites to convince victims that the current webpage is benign. Existing image-based phishing prevention techniques are insufficient to classify such phishing attacks. As a result, these phishing websites will be falsely classified as legitimate websites.
- Most of the existing anti-phishing solutions are unable to reveal the identity of targeted legitimate websites. Instead, they only notify the matching attributes of phishing. It can become a serious threat to internet users if existing anti-phishing solutions cannot identify the identity of new phishing website.
- Existing anti-phishing solutions have low detection speed to classify websites. The detection speed would only get worse for website with dynamic contents because the phishing detectors need to extract and process even more data.

1.5. Research Objectives

The primary objective of this research is to develop a new anti-phishing solution that can reveal the identity of a website and determine the legitimacy accordingly. To achieve that, the specific objectives of the research are defined as follows:

- To evaluate the effectiveness of using graphical elements to recognize the identity of the website.
- To investigate the use of search engines that could reveal the identity of the targeted website.
- To analyze the factors that delay the detection of phishing in the classification.

1.6. Scope of Research

There are varieties of anti-phishing topics to be explored and studied. However, the scope is defined in this research to ensure appropriate contributions are made within the allotted timeframe. Therefore, this research focuses on the detection of phishing websites. Other types of phishing attacks (e.g. email, voice over internet protocol, etc.) are beyond of the scope.

1.7. Outline of the Thesis

The remaining of the thesis is organized as follows:

- Literature review of existing anti-phishing solutions are presented in chapter two. This chapter describes the phishing vectors and phishing techniques used in the creation of phishing websites. Then, the chapter will review different solutions of existing techniques. The review also identifies the advantages and disadvantages of the existing techniques.
- Chapter three presents the methodology of our proposed technique. It describes the application of favicon in Google search by image engine. The chapter also examines additional approach in offsetting the missing favicon scenario. We also discuss in detail a scheme for detecting phishing websites, with or without the presence of a favicon.
- Chapter four presents the experimental results and analysis of the proposed technique. First, we introduce the tools needed to implement the experiment. Then, we explain the method used to collect the data (legitimate and phishing websites). Next, we perform a variety of experiments to assess the proposed technique. Specifically, we are interested to observe the performance of these features in respond to the availability of favicon in legitimate websites and phishing websites. Lastly, we discuss about the limitation imposed by the proposed technique.
- We conclude our work in chapter five, where we summarize all the work from chapter one to chapter four. Next, we describe the contribution of this research. Then, we discuss the direction of future work.

Chapter 2

Phishing Attacks and Prevention Studies

This chapter describes the classification of phishing attacks. The chapter also includes studies on the prevention of phishing attacks. We start by introducing various types of vectors used by phishers to carry out the attacks. Then, we describe the types of phishing that occur on the website. We also discuss some of the approaches taken by the government and non-profit organizations against phishing attacks. For the phishing prevention studies, we analyze current state-of-the-art phishing prevention methods based on the strengths and limitations. Finally, this chapter summarizes the advantages and disadvantages of each method of preventing phishing.

2.1. Classification of Phishing Attacks

Phishing attacks pose a significant risk to the entrepreneurs to develop their business. The severity of phishing attacks can be seen from the report published by APWG [1]. In short, existing anti-phishing solutions are less effective against phishing attacks. Consumer can lose confidence to perform online transactions if the number of phishing activity increased or remained. To cope with the phishing attacks, we review different number of vectors used by phishers to disseminate the attacks. Then, we also review phishing techniques commonly used on the web.

2.1.1. Phishing Vectors

Phishing techniques have evolved since the advancement of information technology. It offers opportunities for phishers to increase the complexity of attacks using web technology. In addition, the phisher can use this technology to victimize internet users from different countries. Although phishing goal is to harvest personal credentials for illegal activities, but it

requires a vector for phishers to launch an attack. Figure 2.1 shows different vectors used in the phishing attacks. In general, we can classify phishing vectors [6] according to the properties described as follows:

- *Spoofed email*. A falsified message was sent by a phisher requesting immediate action from the victims. Usually, the message is modified so that it looks professional and ordinary to the victims. The email could be asking the victims to change account information, update details and verify account information. Sometimes, the victims are asked to visit the embedded link in which it redirects the victims to the phishing website and harvest the credential information.
- *Instant messaging (IM)*. A type of online chat in which they perform a real-time text transmission through the internet. Typically, the chat is insecure and free of speech. Phishers can embed high-profile malicious code into the message and distribute it to the entire group of users. The malicious code can lead victims to a phishing website to dig up personal information. It also can hijack the computers to steal information from the victims.
- *Phone and SMS*. Phishers falsify the message and deliver it to victim's phone number to harvest personal credentials. The message was designed to create a sense of urgency so that the victim will respond to the request immediately. Typically, the message contains a high reward if the victim reacts to it within the prescribed period. However, the leaked information is used by phishers for illegal activities rather than rewarding the victims as promised in the message.
- *Internet browser vulnerabilities*. An old browser (i.e., outdated version) carries a high risk of being targeted by phishers. This risk increases if users never perform updates to the browser. Victims of hijacked computers may not notice changes in the browser. In fact, infected browsers may leak the sensitive information during the transmission. The vulnerability is mostly occurred in the settings of ActiveX, HTML, images, Java, JavaScript, and other web technologies. For example, some websites require ActiveX enabled to view the content or perform certain tasks. Phishers take advantages of these vulnerabilities to steal information from the victims.

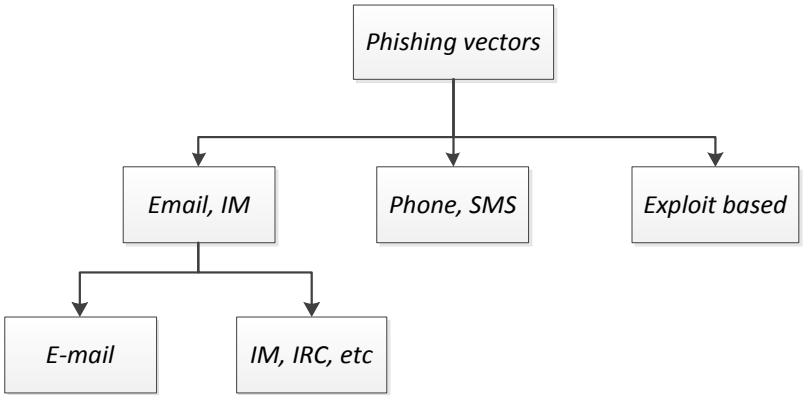


Figure 2.1: Vectors of phishing attacks.

2.1.2. Phishing Techniques

In this section, we will discuss the techniques used in the creation of phishing websites. It is also based on the scope of the research. Nevertheless, phishers like to use different techniques when making bait, especially in phishing websites. This is because most of the casual internet users do not understand or lack of knowledge about security risks on the internet. They tend to follow all the instructions when surfing the web. Therefore, this group of internet users may become victims of phishing if their web browser is outdated or computer does not install anti-phishing software. Figure 2.2 shows the types of phishing techniques used to obfuscate victims on the internet.

- *Malware or Trojan*. Malware or Trojan is a malicious program that is designed to corrupt and steal confidential information directly from victims through computing devices [7]. Typically, the malicious program is delivered to the computing devices via a malicious link embedded in the e-mail. This program will install into the computing devices in an automated fashion without permission and notification once the victim has visited the link. Malicious programs can perform a range of activities such as recording keystrokes on the keyboard, capturing screenshots etc. Then, this piece of information will be sent to the phishers account. It can cause severe financial loss to the victim if the phishers use stolen information for illegal activities.
- *Flash-based.* Flash-based phishing website or Phlashing is a type of phishing technique that uses flash animation to develop the content of a spoof website [8]. It is different from the usual phishing website where the source code of flash animation is hidden from client web browser. Viewers are required to use flash decompiler to read

the source code. Phishers take advantage of this limitation to defeat phishing detectors that scan the textual content of a webpage to find the attributes of phishing. Thus, this phishing technique causes the textual-based phishing detectors to be less effective against flash-based phishing websites. This type of phishing website began to be seen in June 2006 and is becoming more common in late 2006.

- *Popup window.* This type of phishing technique opens another window on the foreground of a browser [9]. This technique will inherit some of the graphical elements of a window opened previously. It convinces internet users that the current window has a relationship with the previous window. However, the information submitted is sent to a different URL destination. Phishers often use this technique in the website to trick internet users. This technique has the potential to circumvent existing phishing detectors. This is because most phishing detectors do not analyze the popup window.
- *Link manipulation.* Link or hyperlink is a stream of text that connects the reader to another website or other parts of the document. This type of phishing technique makes changes to the hyperlink so that it looks similar to the target URL [10]. In addition, phishers are also fond of using the at-sign (@) or dash (-) symbol to mimic the original URL. Many internet users have been given advice by the financial sectors not to visit the suspicious URLs for transaction. This is because the technique is able to deceive victims who are not careful in surfing the internet. Therefore, it is important to verify the URL domain name before visiting.
- *Visual deception.* Visual deception is a type of phishing technique that resembles legitimate websites to create a spoof website [11]. It mimics the layout, components and images used. Usually, the victims of this attack cannot distinguish between the clone website and original website without comparing the original website domain name. In addition, visual deception technique can avoid phishing detector that analyzes the textual content of a website. Phishers take advantage of this limitation to deceive internet users who do not have the ability to detect clone website. This technique can cause serious financial losses to internet users if no countermeasure is proposed to detect this type of phishing websites.
- *URL obfuscation.* URL obfuscation has the same concept of link manipulation except the phishers make changes to the website URL [12]. In addition, phishers can hide the actual phishing URL from the browser address bar. They do so by using a long URL

to force the victim to focus only on the front of the URL. Therefore, the victim who cannot remember the website domain name is gullible by this technique.

- *Cross-site scripting.* Cross-site scripting or XSS is a type of phishing technique that injects malicious code (e.g., JavaScript) to the website to change the content [8]. Information submitted to this website will be sent to the phishers. This type of phishing attack possesses a high threat to both internet users and website owners. This is because the malicious code is not easily detected by phishing detectors. Conversely, phishing detector should do a scan on the website structure to detect changes in the code. In addition, detection of phishing can become complicated if phishers combine this technique with other phishing techniques to increase the complexity of the attacks.

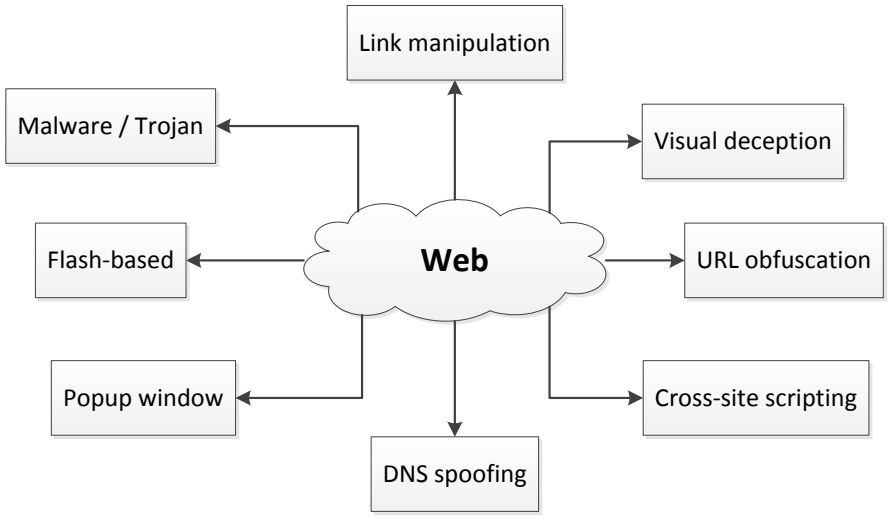


Figure 2.2: Types of phishing techniques through the web.

2.2. Prevention of Phishing Attacks

Phishing is a very complicated deception technique. It manipulates the perception of victims without acknowledging the stolen information. The deception techniques include URL manipulation, appearance impersonation, iconic images etc. In addition, the phisher can transform the layout of a website (e.g., textual content) into image-based to evade the detection. As a result, the high-profile phishing attack that integrates a variety of techniques can defeat the purpose of anti-phishing solution. Nevertheless, we will study a variety of anti-phishing solutions to identify the pros and cons. To do so, the non-technical approaches are reviewed, followed by the technical approaches.

2.2.1. Non-technical Approach

In order to fend off these attacks, government sector (i.e., United States of America) has put much effort [13] to strengthen the security on the web. This includes disseminating precaution guidelines through digital media, conducting discussion forums, educating consumers [14] and etc. However, these efforts did not meet the ideal expectations since there are still many internet users are victimized by phishers. Figure 2.3 shows the category of the anti-phishing organization. Phishing prevention approaches can be divided into non-technical and technical. Non-technical approach delivers information to internet users through various communication channels such as using posters, educational programs, campaigns, games, and etc. The aim is to educate and raise public awareness about phishing attacks. Meanwhile, the technical approach seeks to build greater security for web browsers and provide automated protection for internet users against phishing attacks when surfing the web.

Anti-Phishing Working Group (APWG) [1] and PhishTank [15] are two well-known nonprofit organizations (NPOs) that assist researchers and developers to battle against phishing attacks. They provide blacklist in the form of API for developers and researchers to test against their anti-phishing techniques. However, the provided data is not sufficient to capture the entire phishing activities as it depends on human interaction to report phishing attacks. Since the effectiveness of blacklist is depended on the up-to-date listing, new phishing website can easily escape from the blacklist until someone reports it to the NPO. For websites that are verified and proved deceptive, these NPOs will inform respective hosting to suspend the domain name. It is not uncommon that the reported case may in fact a false alarm and could cause harm to a newly launched legitimate website if anonymous mistakenly reports it to the NPO. This could damage a benign website in terms of reputation and trust.

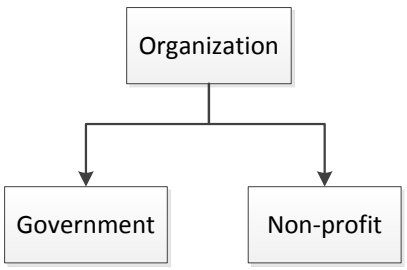


Figure 2.3: Anti-phishing organization category.

Abbasi et al. [16] have conducted several experiments involving 400 participants to evaluate the results displayed by the anti-phishing tools for legitimate websites and phishing websites. Abbasi et al. stated that participants were more likely to agree with the decision made by the anti-phishing tool that claims to have 90 percent accuracy. Instead, participants were less likely to agree with the decision made by the anti-phishing tool that claims to have 60 percent accuracy. This study shows that there is much room for improvement for existing anti-phishing tools so that internet users can have a better understanding about the warnings presented and can make better decisions about the legality of the website. In other words, a new phishing detection mechanism is needed to overcome the limitations imposed by existing anti-phishing solutions.

2.2.2. Technical Approach

Figure 2.4 shows the general framework of phishing detection mechanism. This framework is adopted by most of the existing anti-phishing solutions to detect phishing websites [17 – 19, 22, and 35]. Under normal circumstances, this framework will involve extracting features from the website. Then, the extracted features will be used by one or more components with a threshold function to determine the legitimacy of the website. Database, search engine and classifier are the three components commonly used in phishing detection mechanism. There are several anti-phishing solutions using more than one component to improve the system performance. This is because the phishers prefer to use a variety of techniques during the creation of phishing websites. For instance, a study conducted by Huh and Kim [17] suggests that the detection of phishing can become more efficient when using various components. In the study, they suggest using several different search engines (i.e., Google, Yahoo, and Bing) to obtain the number of the search results and the reputation of the query website. Then, this data may feed to a classifier for classification. Examples of the classifier used in the experiment are Linear Discriminant Analysis, Naive Bayesian, K-Nearest Neighbor and Support Vector Machine.

While combining different components to classify the phishing website can improve the detection accuracy, but it requires more processing time to compute the results. Basnet and Sung [18] also mentioned about the time lag in their study when extracting the necessary

features from different web services. Consequently, the proposed features have caused major performance bottleneck to the system. They plan to integrate additional features in the future to improve the system performance. This includes phishing detection speed. Phishers can still steal important information from the victim if the anti-phishing solutions have poor detection speed. Therefore, many anti-phishing solutions will select the most suitable components with the proposed method. As a result, many anti-phishing solutions are designed to handle certain types of phishing attacks. For example, CANTINA [19] has high detection accuracy against textual-based phishing website but achieve relatively low performance against image-based phishing website.

In the next few sections, we will review the components that are commonly used in phishing detection mechanism. In particular, we want to study the mechanism of these components (as shown in Figure 2.4). We begin by introducing the mechanism of each component, followed by examples of research that uses these components to detect phishing websites. Next, we want to examine the pros and cons of each component.

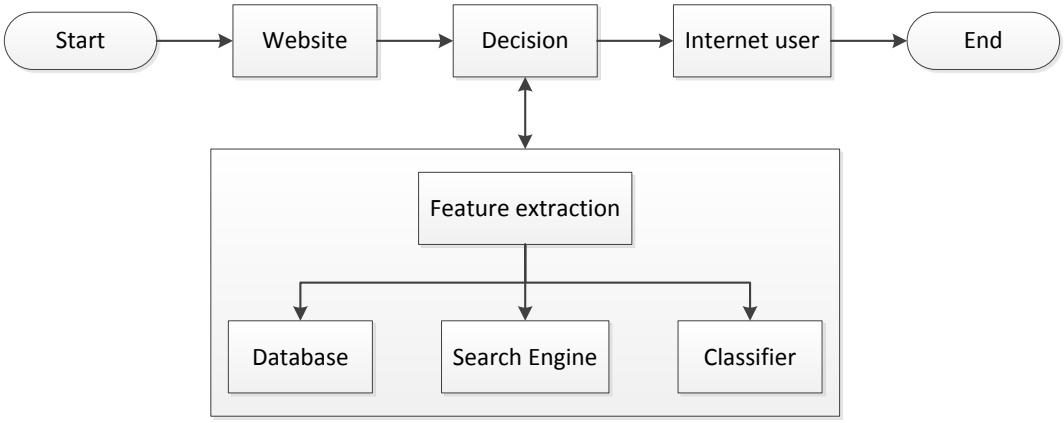


Figure 2.4: The general framework of technical phishing prevention approach.

2.3. Database

Database is one of the components commonly used in phishing detection mechanism. Usually, it is designed to record the properties of legitimate websites. Website URL, domain name registration date, image name, image size, and image file format is part of the properties recorded in the database. Many anti-phishing solutions use the information recorded in the database to compare with the properties taken from other websites. This component helps to

simplify the process of searching for data based on the query string. For example, the legitimacy of the URL can be determined in a short period of time based on the list of blacklisted URLs recorded in the database. This process is fast and does not involve a lot of calculations to obtain the necessary features for classification. Typically, the list-based approach and image-based approach would require a database to check the legitimacy of the website, as shown in Figure 2.5. Details of these two approaches are described in the next subsection.

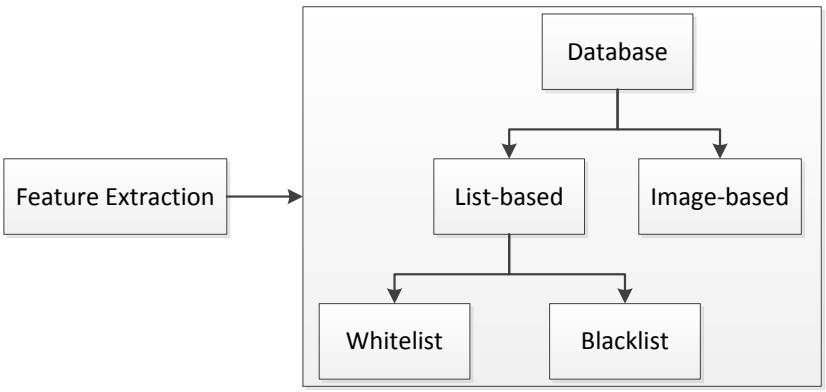


Figure 2.5: The structure of the database used to store and retrieve data.

2.3.1. List-based Approach

List-based approach is the basic for phishing detection. It classifies the legitimacy of a website based on a list of URLs that are recorded in the database. This approach is lightweight and does not require high computing power to perform. List-based approach can be divided into whitelist and blacklist. Whitelist is a list of known legitimate websites, while the blacklist is a list of phishing websites that pose a threat to internet users. Many anti-phishing techniques rely on a combination of whitelists and blacklists to classify websites. One of them is Prevost et al. [20] in which they combine the whitelist and blacklist along with other features for classification. Systems that use list-based approach include Phish Tank Site Checker, Google Safe Browsing, FirePhish, and CallingID Link Advisor [21]. These systems are deployed as a toolbar or an extension of web browser. They will raise an alert to the internet users if the website is not safe to visit. Typically, this approach saves the data locally in a web browser or hosted on a central server. While this approach is simple, it is not effective against websites that are not listed in the database. This is because this approach requires human interaction to report suspicious websites and to update the list.

2.3.1.1. Whitelisting Approach

Whitelist is a list that is used to store information about the legitimate websites. Internet users are free to visit the websites listed. Generally, it only stores the IP address and the website URL. Whitelist can be deployed as a web browser extension or toolbar. It protects internet users from visiting the phishing websites. In other words, it serves as a protective layer that sits between the internet user and the website. This approach is very light. It does not require complex calculations to verify the legitimacy of the website. Instead, it only compares the URL based on the list. Internet users can visit the website if the website's URL is listed in the whitelist. Otherwise, they have to add the URL into the list manually.

It is undeniable that the whitelisting approach can offer maximum security for web browsers, but there are not many studies conducted to assess and improve the usage of whitelist. One of such studies is from Cao et al [22]. They proposed an approach called "Automated Individual White-List (AIWL)" that utilizes whitelist to evaluate the website. In the study, they applied Naive Bayesian classifier to identify the website with a login page. Then, this login page is analyzed by the Login User Interface (LUI) to examine the browser's cache data, browsing history and webpage properties. This technique will warn the internet user if the website requires credential information but the URL is not listed in the whitelist. Nevertheless, this website will be whitelisted if the internet user continues to submit credential information several times despite the warning is presented.

There are many anti-phishing solutions implement the mechanism of whitelisting approach to the system. For example, Cao et al. [22] and Dong et al. [23] applied the mechanism of whitelisting approach in their anti-phishing system. The system will monitor and record activities that occur in the web browser. Then, the system will update the user profile if the activities are different from the profile. Apart from that, there are many advantages of using a whitelist in anti-phishing solutions, such as:

- It is very rare for a whitelisting approach to generate false alarms to legitimate websites. Unless the website address has changed prior to updating the whitelist.
- It can be used to reduce the number of legitimate websites that are classified incorrectly. For example, Xiang and Hong [24] planned to set up a whitelist containing some legitimate websites normally targeted by phishers.

While using the whitelist can improve the performance of anti-phishing solution, but we also need to take into account the limitations incurred by the whitelist, such as:

- The initial list used by this approach is not automated. Whether it has a zero entry or it has to rely on some other mechanism to retrieve the initial list.
- Internet users with zero knowledge of phishing can hardly determine the legitimacy of a website before adding to the list. Studies have shown that internet users are not good at identifying phishing websites [25 – 28]. Therefore, they can be easily fooled by a high quality phishing website.

2.3.1.2. Blacklisting Approach

Blacklist is a list that keeps a record of phishing websites. It is widely adopted in phishing detection techniques because it can be used as a layer of security to protect internet users from being victimized by phishers. Typically, it lists the IP address or URL of the website if the website is identified as a threat to the public. Blacklisting approach is effective in detecting malicious websites provided the website URL is listed in the blacklist [29]. This approach is favored over other detection techniques due to low false positive and simplicity in design and implementation. In addition, this approach is lightweight. It can be deployed as a browser extension or toolbar. Most of the web browsers are equipped with a blacklist. For example, popular web browsers such as Google Chrome, Mozilla Firefox, and Internet Explorer [30 – 31] apply blacklisting approach as a layer of protection to their users. They will make regular updates to maintain the effectiveness of the blacklist. Google Chrome browser has a built-in feature that will automatically update the blacklist [32]. This approach will alert internet users if the website URL exists in the blacklist. However, it does not restrict internet users to visit the blacklisted URLs despite warnings have been presented. Figure 2.6 shows an example of the Google Chrome browser that warns the internet users when the website is verified as phishing.

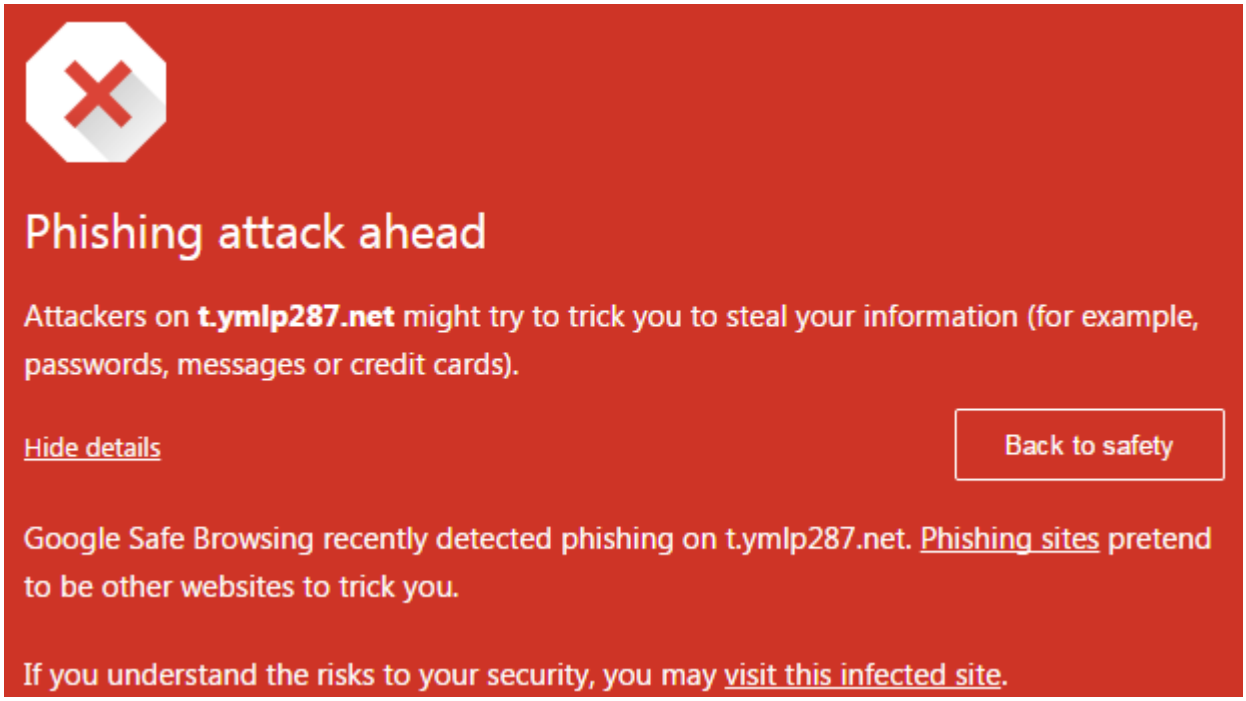


Figure 2.6: A warning from Google Chrome browser is displayed to internet users attempting to access a confirmed phishing website.

Although there are many studies [33 – 36] conducted to assess the performance of the blacklist, but very few focus on enhancing the usability of the blacklisting approach. For example, Prakash et al. [37] proposed a mechanism called "PhishNet" to generate a blacklist which is more like a predictive blacklist from the original phishing websites. They evaluate the performance of this blacklist against phishing websites reported recently by members of anti-phishing organizations such as PhishTank. The results showed that PhishNet can detect new and old phishing websites effectively. This experiment also revealed that the URL of phishing website is syntactically close to each other or semantically similar (i.e., IP address). PhishNet utilizes Domain Name System (DNS) and content matching techniques in an automated fashion to verify the malicious websites are indeed malicious. To be more specific, it uses regular expressions to parse the URL of the website into different parameters (e.g., www, IP address, URL protocol, etc.). Then, these parameters are matched with the predictive blacklist for classification.

It is undeniable that the blacklisting approach can improve the performance of phishing detection. For example, Ma et al. [38] have conducted a study using URL based features to classify websites. They achieved encouraging results with minimal error rate. Furthermore,

they managed to further reduce the error rate when combined with the blacklist. In addition to this, there are several advantages of using a blacklist in phishing detection, such as:

- Anti-phishing organizations (e.g., PhishTank and Anti-Phishing Working Group) are actively involved in the capture of new phishing websites. Then, this data is shared with other organizations to fight phishing attacks.
- Most of the web browsers are equipped with a comprehensive blacklist. The blacklist database is updated regularly to add new phishing websites. Thus, internet users can enjoy surfing the web without having to worry about becoming a victim of phishing.

Blacklisting approach seems to be able to provide adequate protection to internet users. But there are still many cases in which internet users falling prey to phishing websites. For example, a study conducted by Sheng et al. [30] revealed that the anti-phishing solutions can have poor performance in detecting phishing websites when relying only on the blacklist. They claimed that the combination of blacklists and other heuristic techniques can achieve better phishing detection rate. Nevertheless, there are some limitations imposed by this approach are described as follows:

- Blacklisting approach is less effective against newly launched phishing websites, especially during the first hour. This approach requires confirmation and evidence from variety of sources to determine the legitimacy of a website. Therefore, it takes time before a new phishing website is listed in the blacklist. An empirical study conducted by Sheng et al. [29] revealed that the damage caused by phishing websites during the first hour was the highest before they disappear from the internet.
- A large number of new phishing websites were detected each day. However, many non-profit anti-phishing organizations still rely on manual verification for the classification. For example, PhishTank depends on human interaction to report and confirm phishing website. Hence, it becomes less effective to protect internet users during the first hour due to the delay in updating the blacklist.
- This approach may slowly turn internet users away and ignore the delivered warning when it often produces false alarm [39].

2.3.2. Image based Approach

Phishers are particularly fond of using visual deception techniques to create phishing websites. They mimic the look and design by re-using images and other components directly from the targeted legitimate websites. Examples of the copied image are the banner, logo, background, flash objects, advertising, and etc. While examples of the component are the form, text descriptions for internal and external links, title, copyright, and etc. There are several reports of phishing [40 – 41] showed that the use of visual deception technique in phishing websites is increasing. These reports also indicated that many internet users have fallen prey to this phishing technique. In other words, the existing anti-phishing solutions are less effective against this type of phishing websites. Usually, high-profile phishing websites replicate the entire contents of targeted legitimate websites with images. They use very little text to construct the HTML and user input form [42]. Content-based phishing detection approach is less effective for detecting image-based phishing websites because it contains insufficient text to extract. Therefore, the content-based phishing detection approach will fail when it is used to detect image-based phishing websites. Internet users with zero or little knowledge on phishing can hardly distinguish between legitimate websites and impersonated websites without comparing the URLs [25].

2.3.2.1. Visual Layout

There are many aspects to investigate for detecting image-based phishing websites. For example, Medvet et al. [43] proposed a technique that examines the visual layout of a website. This is because most phishing websites are similar to targeted legitimate website in terms of appearance. In other words, the phisher replicates content and images to mimic the look of targeted legitimate websites. The proposed technique can distinguish legitimate websites from phishing websites. The technique extracts the signature of a query website to compare with the signatures of legitimate websites that are stored in the database. This signature includes text and image elements. Examples of text element are textual content, hex color for the foreground and background, font size, font family name and position of the text on the page. Examples of image elements are the properties of HTML image *src* tag, width and height, color histogram, the value of 2D Haar wavelet transformation, and the position of each image on the page. Medvet et al. declare the query website as a phishing website if the

signature is similar to the signature stored in the database. Medvet et al. stated that there are three different types of image-based phishing websites which are level 0, level 1 and level 2. Level 0 is considered to be insignificant visual difference. Level 1 is considered to be slightly different visual and level 2 is deemed as noticeable visual difference. The technique proposed by Medvet et al. yields excellent results for level 0 and level 1 phishing websites. However, this technique is insufficient to detect the entire set of level 2 phishing websites because some phishing websites rebuild the page layout that is completely different from the original website.

2.3.2.2. Image Processing

In addition, to measure the visual layout of a website, many anti-phishing solutions also depend on the image processing tools to analyze the images on the website. For example, the techniques proposed by Hara et al. [44] and Dunlop et al. [45] are using image processing tools to classify the legitimacy of a website. Both researches have excelled in the experiment. However, the proposed techniques failed to analyze random advertisements that are displayed on the page. They proposed to integrate the whitelist and blacklist as a resolution to reduce false alarms caused by random advertisements.

While anti-phishing framework used in [44] is similar to [45], but there are some differences in the mechanism. Hara et al. [44] suggested the use of ImgSeek [46 – 47] to find similar images from a database based on the captured images from the website for classification. ImgSeek uses wavelet transformation to identify images that are similar to the query image. From here, it will generate a total of 100 images that resemble the query image. Hara et al. discovered that the phishers like to imitate the appearance of phishing websites that have the same target. They rarely make major changes to the appearance of the website. Moreover, they make use of phishing kits [5] that are available on the black market to launch massively visual similar phishing websites. Hara et al. declare a website as phishing if ImgSeek can find the image by 35 percent similarity of the database, but the domain name for the two images does not match. Nevertheless, the proposed technique encountered relatively high false positive, but the use of whitelist has substantially reduced false positives.

On the other hand, Dunlop et al. [45] have proposed similar phishing detection approach called “GoldPhish” that also utilizes image processing tools to detect phishing websites. The proposed technique employs Optical Character Recognition (OCR) technology to extract hidden messages from the screenshot of a website. According to Dunlop et al. analysis, visual appearance of a website is designed in such a way to convey important messages to internet users. Therefore, the resolution used in OCR is restricted to 1200×400 pixels where 1200 is the width and 400 is the height. GoldPhish has three stages of processing namely, the Image Capturing, Optical Character Recognition, and Google Search. In the first stage of GoldPhish, screenshot of incoming website is captured. Then, the screenshot is converted to TIFF image and saved in a temporary folder. Next, the image file is analyzed by Microsoft Office Document Imaging (MODI) to extract a list of textual information. MODI is a free program provided as part of Microsoft Office for image analyzing. In the final stage, a list of extracted messages is fed to Google search engine in order to retrieve the websites associated with the messages. The proposed technique declares the website as phishing if the domain name does not match with any domain names returned by Google. However, the authors in [45] noticed that the GoldPhish becomes ineffective if the phishing website contains dynamic contents such as advertisement in which the content may change for each visit. The resilience of OCR technology is limited by the amount and style of text, logos, and images caught in the OCR. In addition, GoldPhish may become ineffective when the OCR software produces inadequate data to be fed to the Google search engine. For example, the Google search engine may not be able to return the right results if the data cannot describe the website properly.

Afroz et al. [48] propose a technique called “PhishZoo” that integrates image processing techniques for detecting image-based phishing websites. The authors employed Scale Invariant Feature Transform (SIFT) algorithm [49] to extract local features of images to detect the differences and not the layout of the website to calculate the differences. The technique also uses a whitelist as a resolution to increase the detection accuracy and reduce the time to detect. PhishZoo has two stages of processing: *Profile Making* and *Profile Matching*. In the first stage, PhishZoo will create a whitelist. This list is compiled by the user and it consists of information about the legitimate websites. This information includes SSL certificates, website URL, content associated with the appearance of the website, and features extracted from the logo. In the second stage, incoming websites are examined with the whitelist to filter out genuine websites. If the website is not on the whitelist, then PhishZoo

will use Term Frequency-Inverse Document Frequency (TF-IDF) technique to extract important keywords from the content of the website. Then, the keywords are used in the whitelist to find a profile that matches these keywords. Next, PhishZoo will extract the local features of the website logo and use these features to find the degree of similarity based on the local features of the matching profile. PhishZoo declares the website as a phishing website if the similarity score obtained is higher than the preset threshold. This technique possesses few limitations as follows:

- Afroz et al. [48] stated that the SIFT technique may fail to match the query image with the images in the database when the query image has been rotated more than thirty degrees. Phishers could exploit this vulnerability to obfuscate the SIFT technique.
- PhishZoo current implementation requires the user to select an image that can represent the website manually. The authors noticed that PhishZoo computation time can increase dramatically if using SIFT technique on all the images on the website.
- Phishers can evade the detection of PhishZoo by transforming the textual content into image-based content. Therefore, this will cause TF-IDF failed to get important keywords. As a result, PhishZoo will use the wrong profile for comparison.

2.4. Search Engine

Search engine is a program designed to search for information from the database (i.e., the World Wide Web) based on the search query made by internet users. Figure 2.7 shows the type of information that can be obtained from the search engine. There are many search engines available currently. However, three of the most popular search engines are Google, Yahoo, and Bing [50]. Search engine ranking is based on the number of unique visitors who use the service. These three search engines have a high ranking is because they contain large database compared to other search engines. In other words, many legitimate websites are included in the search results when internet users search via these search engines. Needless to say, search engines have become the choice of many anti-phishing solutions [17, 18, 19, 38, and 52] to obtain information about a particular website. This information will be used to determine the legitimacy of the website. A simple search uses very little time. However, if the search had to crawl through a different parties or websites to get the data, then it will be very

time consuming. This can degrade the speed of anti-phishing detection to determine the legitimacy of the website.

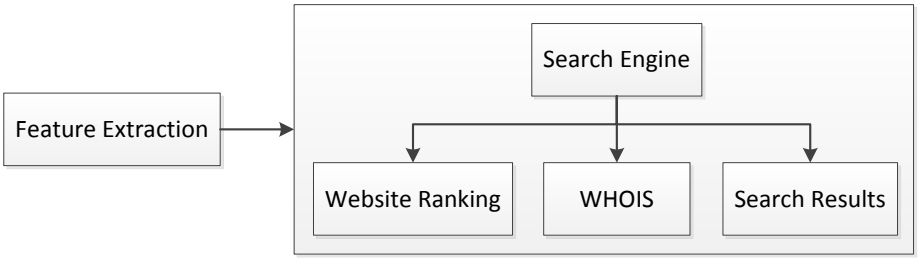


Figure 2.7: Three types of information that can be retrieved by using a search engine.

2.4.1. Website Ranking

Most of the legitimate websites have a ranking on the World Wide Web (W3). This ranking was determined by several factors. For example, domain age, domain history, keyword in title tag, page loading speed, image optimization, outbound and inbound link are among the factors [51] used by search engines to evaluate a website. Website developers should follow the guidelines set by each search engine when building a new website if they want the website to be indexed and appear in search results.

Website ranking can be used in anti-phishing solutions for determining the legitimacy of a website. This is because most of the phishing websites have a low ranking while most of the legitimate websites have a high ranking on the W3. A study conducted by Huh and Kim [17] showed that the website ranking can be used to demonstrate the validity of the website based on the number of results returned by search engines. They carried out this experiment using different search engines (e.g., Google, Yahoo, and Bing) to assess the rate of growth of website ranking for new legitimate websites and new phishing websites. Huh and Kim observed that the number of search results returned to new legitimate websites have increased gradually within the next few days as compared with the new phishing websites. They also found that the new legitimate websites received more inbound links than new phishing websites. In addition, the experiment has shown that time spent on querying the search engine is reasonable low. Although the technique performed well in classification and detection speed, but it does not favour the unpopular websites where they have less inbound links or fewer number of search results.

On the other hand, Sunil et al. [52] proposed a technique that combines phishing detection with Google PageRank [53] to examine the legitimacy of a website. PageRank is a toolbar that can show the ranking of a website. This toolbar is compatible with most of the web browsers and it is free. PageRank uses a scale of 0 - 10 to rate the importance of a website based on the analysis carried out by Google. A scale of 0 indicates the website is relatively new or has less inbound and outbound links. A scale of 10 indicates the website is safe. Google is constantly reviewing and updating the PageRank value based on the characteristics of the website. In addition, Sunil et al. also proposed five heuristics inherited from CANTINA [19]. The five heuristics are suspicious URL, IP address, dots in URL, forms, and age of domain. The authors also claimed that the use of these heuristics and Google PageRank, it has the potential to reveal authenticity and validity of the website. The proposed technique is effective but the Google PageRank is not favour to new legitimate websites because the PageRank update is not done instantly. Therefore, this feature is likely to cause false alarm for new websites as the PageRank value is low.

2.4.2. WHOIS

WHOIS is an internet service that allows access to the information of the registered domain. Most of the web developers are required to register their website with a web hosting provider. This includes the majority of phishing websites. Web hosting provider allows a website to be found in W3. During website registration, the registrars are given the option to protect their privacy that will be accessible through WHOIS. However, not all information can be protected. For examples, registrar name, primary name server, secondary name server, IP address, IP location are some of the information accessible via WHOIS.

Ma et al. [38] proposed a solution that combines the information obtained from WHOIS with the textual properties of the URL for classification. Ma et al. noticed that most of the phishing websites has not been blacklisted. This can be attributed to either they are too new (e.g., less than a week old), never evaluated or assessed incorrectly. In addition, the authors also found that the majority of phishing websites are hosted from the United States with the highest number of phishing incidents. Therefore, the authors investigate the information received from the WHOIS to verify the authenticity of hostname such as ISP, IP address, domain

name and geographical location. To achieve that, the URL is divided into hostname and path. For example, the URL: *www.geocities.com/usr/log.html*, the hostname is *www.geocities.com* and the path is *usr/login.html*. Next, the authors examined length of the hostname and URL. They also check the number of dots in the URL. Ma et al. have conducted several experiments using different features to evaluate the performance of the technique. Nevertheless, they observed that the combination of WHOIS, lexical, and blacklist can achieve the highest detection accuracy. Although the proposed technique is effective, but Ma et al. proposed to give a lower rating for websites hosting from the ISP that has a bad reputation. However, there are also some legitimate websites erroneously classified as phishing websites for using hosting services provided by the ISP who has a bad reputation.

2.4.3. Search Results

Search results are the results returned by the search engine associated with the search query specified by internet users. Typically, the top entry in the search results is the most relevant to the search query. In other words, websites that best match the search query will be presented in the search results. While phishing websites also contain keywords and metadata in accordance with the search query, but the likelihood for search engines (e.g., 0.0025 percent for Google) to return a phishing website at the forefront of the search entries is very low [54]. This is because majority of the phishing websites are short-lived and have fewer inbound links (usually none) to compete with other legitimate websites [17]. Inbound link is a hyperlink that used by a third-party website to refer to the original webpage. Therefore, the results returned by the search engine can be used in phishing detection approach.

A study conducted by Zhang et al. [19] has demonstrated the ability of search results in phishing detection. The authors utilized the search results returned by Google to determine the validity of the website. Google was selected for this study because it has indexed the majority of legitimate websites. Furthermore, all of the legitimate websites have higher ranking in W3 compared with the phishing websites. Zhang et al. suggested the use of TF-IDF to generate lexical signatures for search queries. TF-IDF is chosen because it can generate a list of important keywords related to the website content. Therefore, the authors use the TF-IDF to extract the five most important keywords from the query website. Then, these keywords are fed to the Google search engine. The proposed technique declares the

query website as a phishing website if the search results do not contain the query website domain name. In addition, Zhang et al. observed that the accuracy of detection is in line with the number of search entries used in the classification. In other words, the accuracy of detection increases as the number of entries increases. This approach is very effective against phishing websites, but this approach also leads to high false positive for legitimate websites. This indicates that the Google search engine is very sensitive to the search query. If TF-IDF can not get keywords that represent the website, it is likely that the proposed technique can erroneously classify legitimate websites as phishing websites.

Likewise, Basnet and Sung [18] also conducted a similar study to demonstrate the ability of search results in phishing detection. However, they used different mechanisms to classify websites. They propose to perform a search directly using the website URL and website domain instead of feeding the keywords for search engines. In other words, they check whether the website URL and website domain exist in the search engines. For this reason, Basnet and Sung use the top three search engines (e.g., Google, Yahoo, and Bing) [50] for this study. They claimed that at least one of the search engines must have legitimate websites indexed if not all. In addition, Basnet and Sung also use the archive data from PhishTank [15] to detect phishing websites. For example, they use the top 10 domains, top 10 IPs, and top 10 popular targets to check the legitimacy of the website. They claimed that this data can increase the confidence level for the proposed technique when classifying the website. Furthermore, three different web services are used in the proposed technique. There are StopBadware [55], hpHosts [56], and APWG [1]. These three web services are used to check the validity of the website. Basnet and Sung will flag the website as phishing if the website URL and website domain exist in either one of the web services. Nevertheless, the proposed technique achieved extraordinary results in the experiment. However, there are some problems with this technique:

- The proposed technique faces major performance bottleneck due to the time lag involved in querying two or more search engines. Internet users may try to submit confidential information to a fraudulent website before the technique can warn them.
- The proposed technique uses search engines to check the legitimacy of the website. However, this technique does not favor the new legitimate websites if the websites have not been crawled and indexed by search engines.

- The proposed technique can fail to detect phishing website if phishers use more than a hosting to host the website.

Since phishing websites are duplicating the look and design, so they look identical to the legitimate website. Knowing that, Chang et al. [57] proposed a technique that uses Google image search engine to return the list of search entries relating to the query website. To achieve that, they chose to extract the logo of the website because it usually represents the identity of a legitimate website. To do so, first, they capture screenshot of the query website. Then, the screenshot is segmented into three dimensions (1 × 3, 2 × 3, and 3 × 3). The segmentation will reduce areas that are not related progressively to get a tighter fit for the logo. In addition, Chang et al. also cropped the logo manually so that this logo contains minimum unrelated areas. Thus, four different data sets that contain different dimensions are produced namely dataset 1 (1 × 3), dataset 2 (2 × 3), dataset 3 (3 × 3) and dataset 4 (logo manually cropped). Next, they feed Google image search engine with these data sets and observe the performance of detection. The proposed technique will declare the query website as a phishing website if the domain name of the query website does not match with any of the domain name returned from the top 30 of search entries. Nevertheless, the proposed technique achieves good performance in classifying legitimate websites and phishing websites. But this technique has some limitations as follows:

- If the logo is found in a different dimension of the screenshot, then this technique can provide a wrong image for the Google image search engine.
- The authors [57] mentioned that there are some circumstances in which the website will contain multiple logos (i.e., the logo of the website and the logo of the social networking). Without effective segmentation, this technique cannot extract the correct logo.
- The proposed technique may segment the image that closely resembles to the logo. Therefore, it can cause Google image search to return results unrelated to the logo.

2.5. Classifier

In the field of anti-phishing, researchers adopt classifier as a decision maker to distinguish legitimate websites from phishing websites based on the characteristics of the website. There

are many types of classifiers can be used to solve the classification problem, as shown in Figure 2.8. Support Vector Machine, Logistic Regression, k-Nearest Neighbor, and Naive Bayes are some of the examples of classifier commonly used in classification problems. Each classifier is different in terms of performance. For example, low variance and high bias classifier (e.g., Naïve Bayes) is better for training if the sample size is fixed. However, high variance and low bias classifier (e.g., k-Nearest Neighbor) is better for training if the sample size is increased gradually. This is because high variance and low bias classifier can provide accurate model for the data with lower asymptotic error.

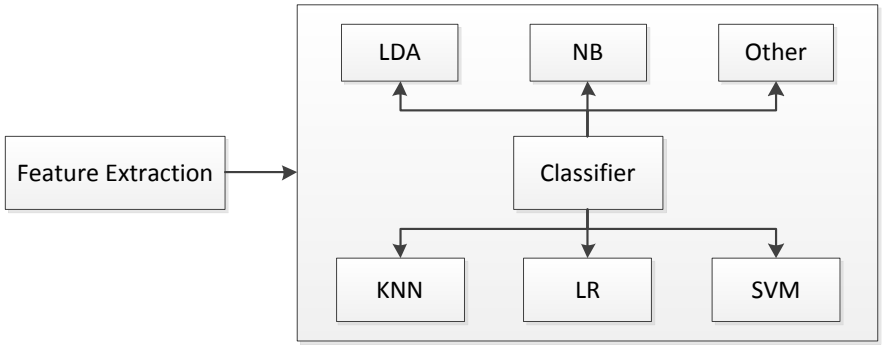


Figure 2.8: Classifier commonly used in website classification.

Basnet and Sung [18] adopted Logistic Regression (LR) to determine the legitimacy of a website. LR is a statistical model that can predict the outcome (i.e., legitimate or phishing) based on the data fitted into the logit function of logistic curve. LR is used because it uses white box model which often has decision rules that are easier to interpret in terms of relevant and irrelevant features. They achieved encouraging experimental results using data obtained from PhishTank [15] and Yahoo random page service[2].

On the contrary, a study conducted by Ma et al. [38] has shown that each classifier has different performance in terms of detection rate and detection speed when using the same data. They observed that Support Vector Machine (SVM) and Naive Bayes (NB) outperformed LR in terms of detection speed during the training process. However, NB is experiencing high error rate compared to SVM and LR. During the testing phase, Ma et al. observed that a significant improvement for LR in terms of detection speed while having the same detection rate with SVM. They decided to use LR for the remaining experiments. This is because of the difference in error between SVM and LR is so small. They claimed that LR

[2] http://random.yahoo.com/bin/ryl

has better interpretability where it is useful to understand how the model performs and how it can be improved.

Similarly, Huh and Kim [17] also conducted a study that demonstrated the ability of the classifier to determine the legitimacy of a website. Huh and Kim used Linear Discriminant Analysis (LDA), NB, *K*-Nearest Neighbor (KNN), and SVM in their experiment to analyze the rate of growth of website ranking returned by search engines (e.g., Google, Bing, and Yahoo). They found that all of the classifier can achieve good detection speed (i.e., less than 0.5 seconds). In particular, Yahoo search engine achieves the lowest time in the classification (i.e., 0.005 seconds). However, Huh and Kim noticed that LDA and SVM did not perform well (i.e., less than 75 percent accuracy) for Yahoo search engine when internet protocols (e.g., http) are included in the URL for the search query. Instead, they can achieve good accuracy (i.e., higher than 75 percent accuracy) for Yahoo search engine without using the protocol in the search query. Likewise, NB did not show good performance when using Bing search engine with or without the use of protocol for URL in the search query. Huh and Kim explained that NB relies on each attribute (i.e., with or without protocol in the URL) being independent, but both attributes were not truly independent in the Bing search engine. Nevertheless, KNN (K = 3) produced the best results in the three search engines. This indicates that KNN is able to find two or three data corresponding to the query website when K is set to 3.

2.6. Summary

In this section, we will summarize and discuss the findings from section 2.1 to section 2.5 that will lead to our research direction. In short, phishing is a type of cybercrimes that aims to steal important information such as personal information, credit cards, driving licenses, and others for activities that violate the law without the consent of the owner. Phishers exploit vulnerabilities found on the internet to build a phishing website. They send this bait (i.e., phishing websites) to victims through various communication channels such as e-mail, instant messaging, phone, SMS, and web browser. Furthermore, they can develop sophisticated phishing techniques due to advances in information technology.

Many organizations, whether for-profit or nonprofit have joined forces to fight against phishing attacks. They collect and study the characteristics of phishing websites through different channels (e.g., PhishTank 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 methods used to convey information include using posters, educational programs, campaigns, games, and others. However, these efforts did not meet the expectations due to phishing incidents still occur, as reported in [1 – 4]. Therefore, it is necessary to understand the pros and cons of anti-phishing solutions in order to develop a new solution that can overcome the limits imposed.

List-based approach is a type of anti-phishing techniques that assess the legitimacy of a website based on the list. Normally, the list will be used as an extension of the web browser or toolbar. This approach will trigger a warning to internet users if the website is malicious or does not exist in the database. The list-based approach is simple and lightweight. It only compares the URL to the list in the database. This list can be divided into whitelist and blacklist. 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. Whitelist can provide good security because it only allows internet users to access websites that have been whitelisted. It is very rare for a whitelist containing phishing URL. However, internet users can inadvertently whitelist a phishing website if they cannot distinguish between legitimate websites and phishing websites. Conversely, the blacklist is a list consisting 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, as reported in [29]. In addition, the study [30] has shown that the blacklist is less effective against zero hour phishing (aged less than one hour).

Image-based approach is another type of anti-phishing techniques used to analyze images on the website. 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 approach is proposed as a result of an increase in image-based

phishing websites, as reported in [40 – 41]. Furthermore, it can be used to overcome the limitations imposed by the anti-phishing solutions that extract the contents of the website (i.e., the textual content) for analysis. Nevertheless, there are many solutions proposed to address this type of phishing websites. These solutions include utilizing the visual layout of website [43], image processing tools (e.g., ImgSeek [44] and OCR technology [45]), and image processing techniques (e.g., SIFT [48]). This approach is very effective against phishing websites that target legitimate websites whose information is already in the database. However, this approach requires a comprehensive database to become effective. In other words, it may cause false alarms to legitimate websites that have not been registered with the database. The effectiveness of this approach also depends on the quality of the image. For example, OCR will have difficulty to scan the contents of the image if the image is blurry. In addition, the method proposed by Medvet et al. [43] who studied the visual layout of the website may fail if the website contains advertisements that display dynamic content.

Another type of anti-phishing solution is to utilize the search engine. In other words, this type of anti-phishing solution will leverage the power of search engine and perform further analysis based on the returned search results to determine the legitimacy of a website. Usually, this solution uses the top three search engines to obtain information. The top three search engines are Google, Yahoo, and Bing. The results returned by these search engines are very sensitive to the search query. Each entry in the search results are usually listed in order of importance (i.e., website ranking and Meta keywords) related to the search query. There are many studies that use search engine to check the legitimacy of a website. For example, Sunil et al. [52] proposed a method that uses website ranking derived from the PageRank [53] to examine the legitimacy of the website. Huh and Kim [17] 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 number of search entries returned by Google for new legitimate websites and new phishing websites. Similar methods can be found in [18, 19, and 56]. 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, this solution will become less effective for legitimate websites that do not have a high ranking in search engine index.

Classifier is a mathematical function that involves optimization algorithm. It can be used to determine the legitimacy of a website based on the characteristics. There are many types of classifier and each will give different performance (i.e., detection rate and detection speed). Therefore, it is necessary to evaluate the results of each classifier used to solve classification problems. There are many studies conducted to show the performance of the classifier in anti-phishing. For example, Basnet and Sung [18] stated that the Logistic Regression (LR) is sufficient to achieve good performance in classifying websites. Ma et al. [38] confirmed the statement made by the authors in [18] after running an experiment that demonstrates the ability of LR, Naive Bayes (NB), and Support Vector Machine (SVM) to classify websites.

Chapter 3

Methodology and Detection Schemes

This chapter describes in detail the methodology and algorithm of Phishdentity. First, we explain the motivations that led to the development of our proposed framework. The proposed framework is illustrated graphically to demonstrate the procedure of each module for the websites. Then, we introduce two main components of the proposed technique, namely the website favicon and Google search by image. We also explain the proposed features used in analyzing the search results returned by Google. Next, we propose a solution to overcome the limitation when the favicon is missing. This chapter includes detailed explanations of the detection scheme to classify legitimate websites and phishing websites.

3.1. Proposed Framework

Our proposed technique is called Phishdentity. It is driven by Chang et al. [57] which is to find the identity of a website using Google image search engine. However, there are some differences with the technique proposed in [57]. Instead of using an entire webpage for analysis, we decided to use the website favicon for analysis. Moreover, our proposed technique does not require high computing power to perform. Favicon is chosen because it represents the brand of the website. In addition, the favicon will not be affected by dynamic contents (e.g., advertisement) displayed on webpages. The technique proposed by the authors in [44 – 45] can misclassify a website if the website contains advertisement that will change the content for each visit. Nevertheless, we decided to use Google image search engine for our proposed technique because it allows the image to be used as a search query to find information. Google is selected for our proposed technique because it has indexed majority of legitimate websites [19]. Furthermore, using Google image search engine can eliminate the need for maintaining a database that may affect the effectiveness to detect phishing website.

Figure 3.1 shows the proposed framework of Phishdentity for classifying incoming webpages. Phishdentity consists of two modules which are module A and module B. Module A utilizes the URL of a webpage as the input for analysis. The URL is transferred to the first process of the module for favicon extraction in which the path of the URL is changed to the favicon image path. Then, the favicon image path is used in the next process which is Google search by image. In this process, the favicon image path is fed to Google search by image engine to return a list of entries relevant to the query favicon. Next, we extract the required features from the search results for analysis. In this process, we determine the identity of the website using the entries of search results. This process also contributes to the final decision to determine the legitimacy of the website. On the other hand, module B utilizes the URL of a webpage for analysis. Instead of using search engine (e.g., Google) to retrieve the results for feature extraction, we analyze the URL to extract the required features. In the first process, the URL is analyzed based on lexical analysis. Example of lexical analysis is whether the URL contains the suspicious symbols or excessive use of the number of dots in a domain. Next, we analyze the URL based on host-based analysis. For example, we examine the domain age and also check whether the URL domain is formed by IP address. After that, the next process will perform domain analysis. In this process, we apply Web of Trust service to obtain the ranking of URL. We calculate scores in module B based on the results obtained from each process (i.e., lexical, host-based, and domain analysis). Finally, we combine the scores obtained from both modules to get the final score for the website. If the final score exceeds certain threshold, then it is classified as a phishing website. Otherwise, it is classified as a legitimate website.

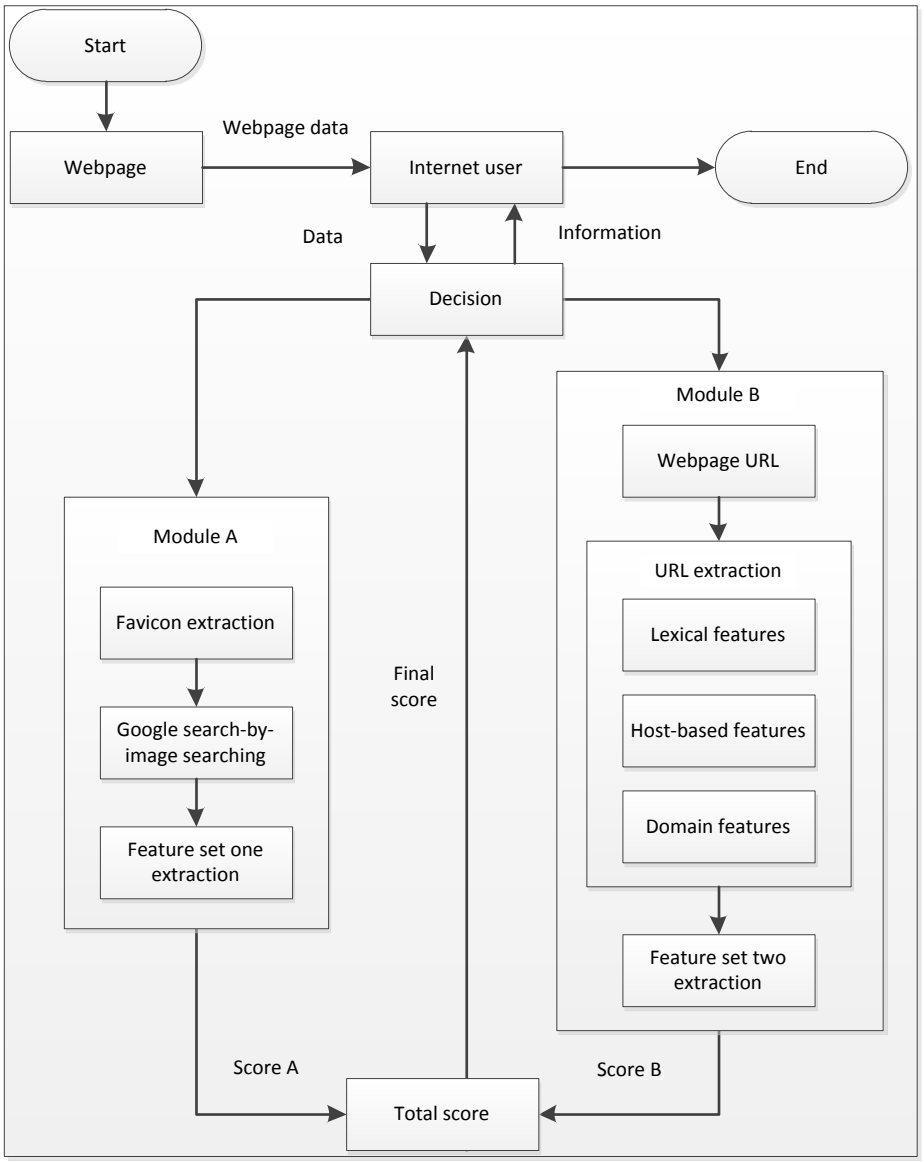


Figure 3.1: Proposed framework of Phishdentity.

3.2. Website Favicon

Favicon is a shortcut icon attached to the URL that is displayed in the desktop browser's address bar, browser tab, or next to the website name in a browser's bookmark list. Figure 3.2 shows an example of the Internet Explorer browser showing the Google favicon. Favicon represents the identity of a website in a 16×16 pixels image files. It is also available in several different image sizes, such as 32×32, 48×48, or 64×64 pixels in size. As of today, most of the desktop browsers can support favicon in different file format (e.g., ICO, PNG, GIF, animated

GIFs, JPEG, APNG, and SVG). Table 3.1 shows seven ways in HTML to access the favicon of a webpage.

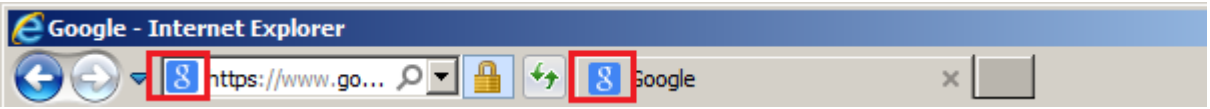


Figure 3.2: Example of Google's favicon displayed on the browser's address bar and tab.

Table 3.1: Ways in HTML used to access the favicon.

Method	HTML Codes
I	http://www.domain.com/favicon.ico or https://www.domain.com/favicon.ico
II	<link rel="shortcut icon" href= "/favicon.ico" />
III	<link rel="icon" href= "/favicon.ico" />
IV	<link rel="icon" type="image/vnd.microsoft.icon" href="/favicon.ico" />
V	<link rel="icon" type="image/png" href="/favicon.png" />
VI	<link rel="icon" type="image/gif" href= "/favicon.gif" />
VII	<link rel="apple-touch-icon" href= "images/favicon.ico" />

Method I is the most direct way to access the favicon. This method displays the favicon of a website in an empty webpage. Method II to VII is the common HTML scripts to display the favicon. Method VII is used exclusively by Apple devices to display the favicon with the dimension of 57×57 pixels and above. Based on the Table 3.1, *rel*="shortcut icon", *rel*="icon", and *rel*="apple-touch-icon" are the three common attributes used to display the favicon on the browser's address bar and on the browser bookmark, while *href* indicates the path in which the favicon is located. The *type* tag is used to specify the file format when the favicon is not in ICO format. Method II to VII is located in the head section of HTML structure in order to load and display the favicon on the browser. We observed that most of the legitimate websites of our data collection has a unique favicon and only 23 websites that do not have the presence of favicon. Contrary, most of the phishing website has a favicon, where the appearance is identical with only 13 phishing websites do not have the presence of favicon. We argued that the phishers prefer to download the favicon directly from legitimate websites. In other words, the phisher copy the favicon image path of legitimate websites and placed it in the *href* attribute.

3.3. Google search by image

By default, the Google search engine allows text to be used as a search query to lookup all sorts of images. Nevertheless, Google also allows users to search for information based on the content of an image. This mechanism of search-by-image content is essential for our proposed technique to retrieve the right information about an image. Figure 3.3 shows an example of Google search by image interface. Basically, there are two options to use Google search by image as described below:

- *Paste image URL*. This option allows the user to input a query image which is located in Internet. The complete path to the image is required. For example, using the URL of *http://www.paypal.com/favicon.ico*, Google will return a list of information related to the PayPal.
- *Upload an image*. This option allows the user to upload an image from the computer local drive. In addition, the user can drag and drop the image directly into Google search by image interface.

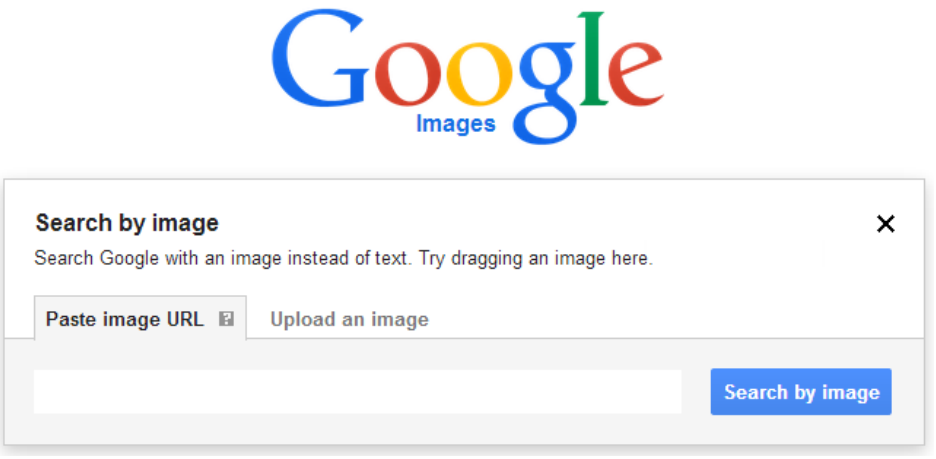


Figure 3.3: Example of Google search by image interface.

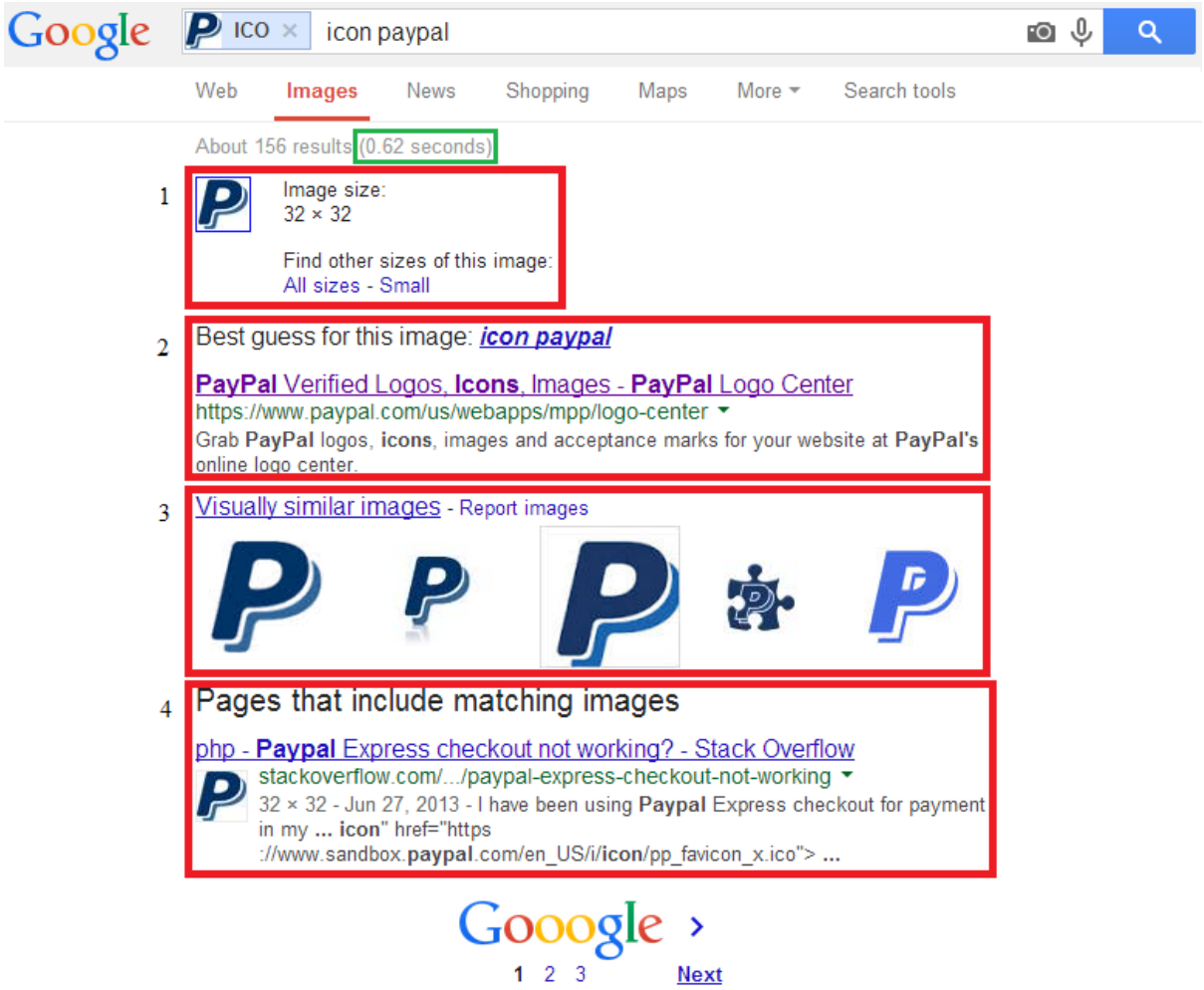


Figure 3.4: Example of Google search by image results when PayPal favicon is queried.

Google search by image is an image query function with 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., colors, 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 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 3.4 shows an example of the search result returned by Google search by image engine when PayPal favicon is queried. Based on Figure 3.4, there are a total of four elements displayed on the search result page (marked with red line boxes) as mentioned below:

- *Preview image*. This element contains several versions of the query image in a variety of sizes if the image data from search engine database match the query image.

- *Best guess*. This element returns a text description of the query image with a link for further information if the query image contains matching image data from the search engine database.
- *Visually similar images*. This element returns a list of images that are visually identical to the query image.
- *Pages that include matching images*. This element returns a list of entries in the form of text description or image which are related to the matching query image.

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

Figure 3.5: Code snippet of Google search by image API.

In this research, we employ the *Paste image URL* option into our proposed technique to feed the favicon into Google search by image engine. First, we feed the favicon image path of a webpage to *Paste image URL* option. To achieve this, we use a custom API developed by Artur Schaback [58], as shown in Figure 3.5. The API utilizes Google search by image engine to return search results relevant to the query image. *image_url=<Image URL>* is the parameter in which the image path is inserted. Then, we can apply proposed feature set one to the search results returned by the API. To be more specific, we use second and fourth elements that occur in the search results (Figure 3.4). Second and fourth elements contain valuable information that provides clues to our findings. Examples of information are the text description and highlighted text as shown in Figure 3.4. From here, we can uncover the identity of a webpage based on analytical results obtained from the proposed features. Furthermore, Google search by image engine was able to return search results in a significant time (i.e., less than a second as indicated by the green line box in Figure 3.4). The first and third elements are not selected as the two elements do not have enough information to be extracted. The next step of the proposed technique is to perform feature extraction on the selected elements from the search results. Detail of the feature extraction will be discussed in the next section.

3.4. Proposed Feature Set One

In this section, the proposed features are explained in detail to illustrate the impact of each feature towards the classification of websites. These features contribute towards revealing the identity of websites. Based on the observation and experiment done on the search results, we propose a total of five features that utilize the search results for classification. Our proposed features are divided into two layers, as shown in Figure 3.6. In the first layer, we extract four features from the search results namely the second-level domain, path in URL, title and snippet, and highlighted text. The first layer will produce a list of unique second-level domains with different frequencies on each feature. In the second layer, we extract the domain name of a query website to find matching from a list of unique second-level domains. Details of each feature and identification mechanisms are revealed in the next sections.

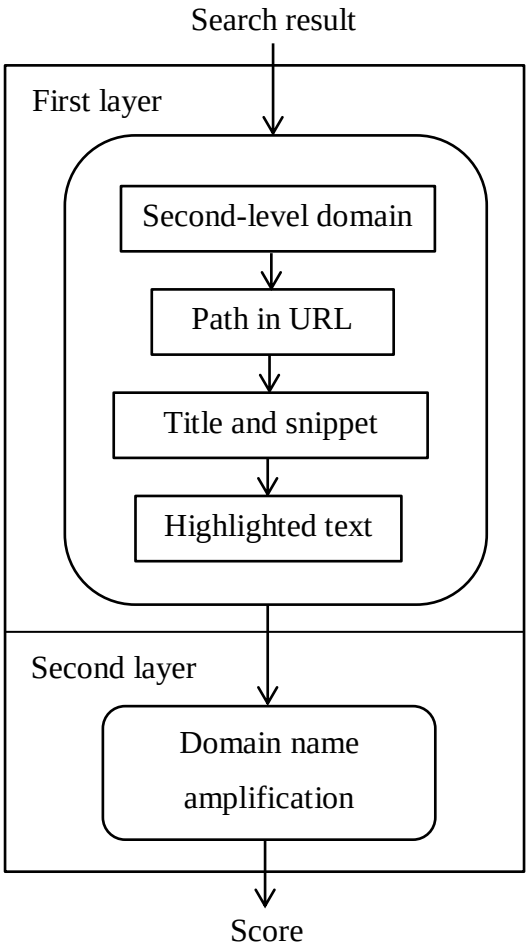


Figure 3.6: Architecture of the proposed feature set one.

3.4.1. Second-level Domain

Second-level domain (SLD) is a name by which a website is known. It is located directly below the top-level domain (TLD). For example, in *http://www.mydomain.com/*, *mydomain* is the second-level domain of *.com* TLD. We propose to use the SLD as part of the features to identify the website because it represents the brand name of a legitimate website. It has high possibility to uncover the identity of a website based on the search results. To this end, we have decided to use each entry of the search results for analysis. Hence, the SLD is extracted from a list of entries returned by the Google search by image engine (as shown in Figure 3.7 where the SLDs are marked with red line boxes). In order to avoid confusion towards the counting, we use the term "unique term" to represent each unique SLD extracted from the entries. It is also used for the following sections. Therefore, the frequency of each unique term is computed from the number of occurrence of SLDs found from the search result (as shown in Table 3.2).

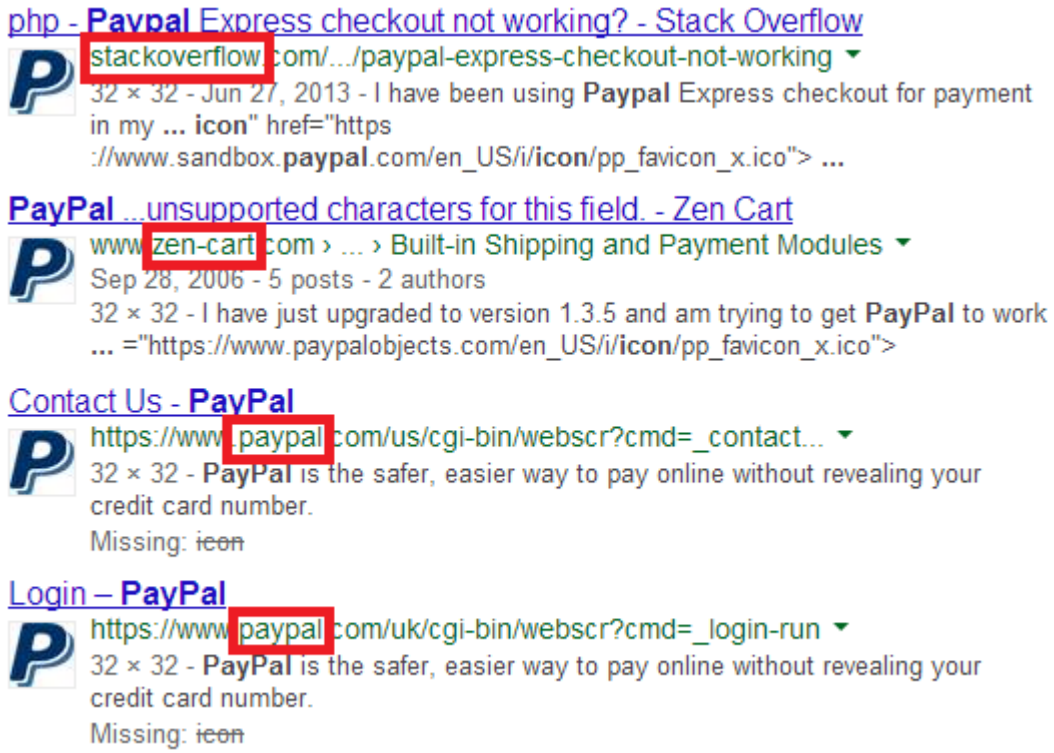


Figure 3.7: Example of second-level domains.

Table 3.2: Calculating the frequency of unique terms in the second-level domain.

Unique term	Second-level domain
stackoverflow	1
zen-cart	1
paypal	2

3.4.2. Path in URL

The path is usually located after the top-level domain for the URL, as show in Figure 3.8 in which the path is marked by a red line box. It indicates the location of a resource. For example, in *http://www.domain.com/image/index.php*, */image/index.php* refers to a unique location for a file named *index.php*. Path is used as part of the proposed features because the search results often contain terms in the path that are associated to the identity of the targeted legitimate website. Furthermore, phishers are known to use terms that are related to the identity of legitimate website to construct the path. While phishers can change the path of URL in a different way, but they are still interested to maintain the identity 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 based on the search results returned by Google search by image engine. Then, we use the unique terms extracted in the proceeding section to find matching identities from all paths. Hence, in order to capture this property, the number of occurrence for every unique term found in the path is recorded. Table 3.3 demonstrates the results of the counting for each unique term using the path in URL feature based on the entries in Figure 3.7.

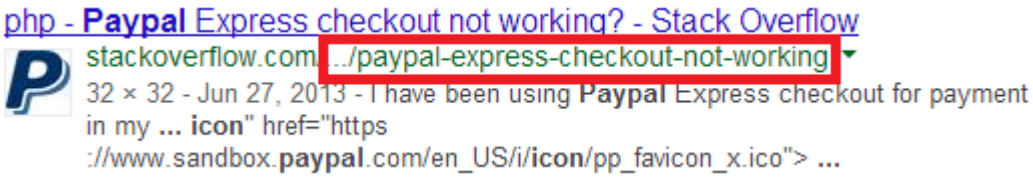


Figure 3.8: Example of a path.

Table 3.3: Calculating the frequency of unique terms in the path.

Unique Term	Path in URL
stackoverflow	0
zen-cart	0
paypal	1

3.4.3. Title and Snippet

One of the elements from the Google search by image is shown in Figure 3.9. The element contains the title and text description. From Figure 3.9, 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 query website does not always appear in the URL of the search entries. In contrast, the identity of the website can be found in the title or snippet of the search entries. Therefore, each unique term used in the previous section will be used to search for matching within all the titles and snippets of search results. Hence, the number of occurrence for each unique term found in the titles and snippets are recorded, as shown in Table 3.4.

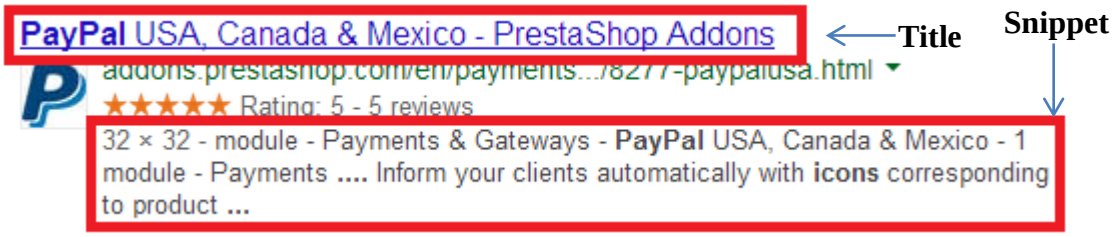


Figure 3.9: Example of an entry with title and snippet in the search results.

Table 3.4: Calculating the frequency of unique terms in the title and snippet.

Unique Term	Title and Snippet
stackoverflow	0
zen-cart	0
paypal	10

3.4.4. Highlighted Text

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 to the search favicon. Figure 3.10 shows the corresponding bold texts of PayPal when PayPal favicon is fed to the Google search by image engine. This feature has very high potential to reveal the true identity of the favicon as the content of a favicon must comply with the image content defined by Google search by image to have bold text appeared in the search results. Therefore, the proposed technique extracts all the bold text from each entry in the search results to find a match based on the unique terms extracted from Section 3.4.1. The numbers of occurrences for each unique term found in all the highlighted text are recorded (as shown in Table 3.5).

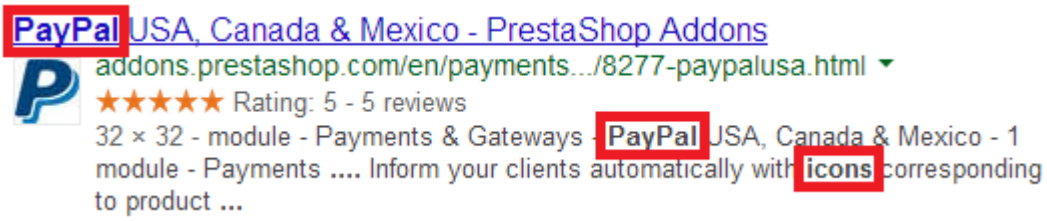


Figure 3.10: Example of highlighted text on one of the entries in the search results.

Table 3.5: Calculating the frequency of unique terms of the highlighted text.

Unique Term	Highlighted Text
stackoverflow	0
zen-cart	0
paypal	9

3.4.5. Domain Name Amplification

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. Usually, the domain name is formed by a second-level domain (SLD) and a top-level domain (TLD). For example, *http://mydomain.com/* is the domain name where *mydomain* is the SLD and *.com* is the TLD. It is also very common for a domain name to have a subdomain, for example *http://subd.mydomain.com/*. Typically, a legitimate website has a unique name in which it differs from other websites. Contrary, phishers are more likely to incorporate the name of legitimate website into phishing URLs to confuse internet users. For this reason, we use the SLD of query website to find matching based on a list of unique terms extracted previously from the results returned by search engine. Then, we increase five percent to the total frequency aggregated from each feature extracted from the first layer (i.e., Section 3.4.1 – 3.4.4) for the corresponding unique term (as shown in Table 3.6, where the number of total new frequency for term "*paypal*" is increased from 22 to 23).

Based on our early observation on the experiments, an increase of five percent to the total frequency of first layer (i.e., SLD, path in URL, title and snippet, and highlighted text) reduces significant number of incorrect classified legitimate websites. This feature does not reduce the accuracy of our proposed technique in detecting phishing websites. We proposed an increase of five percent to the total frequency is due to insignificant results (i.e., search

entries) returned by the Google search by image engine for some legitimate websites. It occurs when the content of the favicon is less distinctive (image features such as region of interest).

Table 3.6: Changes in the total frequency of the first layer.

Unique Term	Second-level Domain	Path in URL	Title and Snippet	Highlighted Text	Total Frequency (old)	Total Frequency (new)
stackoverflow	1	0	0	0	1	1
zen-cart	1	0	0	0	1	1
paypal	2	1	10	9	22	23

3.5. Proposed Feature Set Two

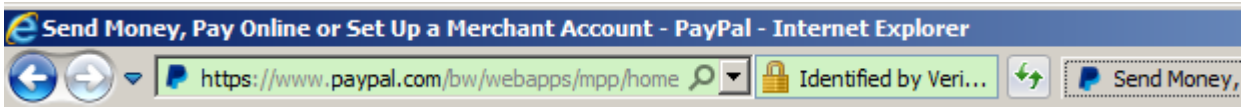
We are aware that there may be some websites that do not have the favicon. Phishdentity will be ineffective if the favicon is missing from the website. Therefore, we propose additional approach to compensate when favicon is missing. In order to minimize additional overhead, this approach should use little computational power to analyze the website. Throughout the study, we observed that analyzing the properties of the URL requires the least computational power compared with other approaches. For this reason, we propose several features that examine the properties of query website such as lexical features, host-based features, and domain features. These features are suspicious URL, dots in domain, age of domain, IP address and web of trust. We discuss in detail for each feature in the following sections.

3.5.1. Lexical Analysis

Lexical analysis is the process of extracting a sequence of characters based on the revelations given to form a sequence of tokens for analysis. For example, given, *http://www.paypal.com/* we may extract the "*www*", "*paypal*", and "*com*" for analysis. For this reason, we proposed two features that are based on the lexical URL of query website for analysis. Details of the proposed features are discussed in the following subsections.

3.5.1.1. Suspicious URL

This feature examines the URL for at-sign symbol (@) or dash symbol (-). If the at-sign symbol is present in the domain name, it forces the string to the left to be removed while the string to the right is considered to be the actual URL. We noticed that there are some latest browsers (i.e., Google Chrome) still experiencing 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 to surf the internet where they are exposed to this type of phishing attack. Phishers are manipulating this technique to redirect victims' browser to phishing website. This technique is very famous among phishers to trick potential victims that rarely check the website URL when browsing the internet. Moreover, this trick is made possible as a result of the very long URL in which most of victims' browser is unable to display the entire length of URL. In addition, the Internet Explorer browser can support a maximum of 2048 characters for the length of the URL while other browsers support more characters than Internet Explorer browser. Therefore, a legitimate URL often appears ahead of phishing URL in the address bar. For example, Figure 3.11 shows this scenario. It only displays half of the full URL in the browser address bar where the other half is covered by browser's buttons, menus, and tabs. Likewise, the dash is also often used in phishing domains. Phishers imitate legitimate domain names by inserting dashes into the URL to make unsuspecting user to believe it is the legitimate domain name. For example, *http://www.pay-pal.com/* is imitating PayPal domain name, *http://www.paypal.com/*. However, the use of dashes in a domain name is rarely seen from a legitimate website. This technique can easily deceive users who do not understand the syntax of URL and cannot tell the difference in domain names. Thus, to capture this property, we use at-sign and dash symbols to search for matching symbol from the query website. First, we tokenize the URL by breaking it into a stream of text. To achieve this, the token is separated from the URL when it meets a dot or slash. The tokenization process continues until end of URL. Next, we find the corresponding symbol by going through each token. If there is a matching symbol in the token, then we assign 1 to this feature. Otherwise, we assign zero to this feature.



(a) Visible part of PayPal homepage URL.

(b) Hidden part of the URL.

Figure 3.11: Part of the phishing URL is hidden behind the browser's address bar.

3.5.1.2. Dots in Domain

Based on our observation during the data collection phase, we find that there is no legitimate websites that have more than five dots in the URL domain. In contrast, we observed a total of 134 (26.8%) out of 500 phishing websites use more than five dots in creating the URL domain. We reasoned that phishers use such trick to obfuscate internet users from perceiving the actual phishing URL. Table 3.7 illustrates some examples of this feature. Example 1 in the table shows the legitimate website URL and the rest are phishing examples. This feature was mentioned in [19] [52]. Hence, we adopt their ideas in our proposed technique to classify websites. To do so, we extract the domain from the query website. Then, we perform counting on number of dots that are present in the domain. We assign 1 to this feature if the domain of the query website contains five or more dots. Otherwise, we assign zero to this feature.

Table 3.7: Examples of dots in URL feature.

Example	URL	Numbers of Dot
1	https://www.paypal.com/index.php	2
2	http://www.pay.pal.site.real.com/index.php	5
3	http://site.paypal.com.my.origin.com/somepath/index.php	5
4	http://www.paypal.com.my.www.domain.com/index.php	6

3.5.2. Host-based Analysis

Host-based analysis is the process of analyzing the website domain by examining the history of the domain and hosting services used. For example, given, *http://www.mydomain.com/* we

may examine the age or name of the domain for legitimacy. If the website is old (e.g., more than 30 days) or hosting provider is authorized, then we can assume it is safe to visit. For this reason, we proposed two features that are based on the host of the query website for analysis. Details of the proposed features are discussed in the following subsections.

3.5.2.1. Age of Domain

This feature examines the age of domain with WHOIS service. Based on the experiments conducted, we observed that many phishing websites have a very short lifespan. Typically, they last from few hours to few days before disappearing from the internet. CANTINA [20] proposed a similar feature to check the age of 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 [59]. In addition, the report also showed that most of the phishing websites only have an average lifespan of a week. We assign 1 to this feature if the age of the query website is equal or less than 30 days. Otherwise, we assign zero to this feature.

3.5.2.2. IP Address

An IP address is numerical numbers 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 websites that use IP address in the URL domain to create phishing websites. However, we did not find any legitimate websites that use IP address in the URL domain. We believe that legitimate websites will not use the IP address as the address of the website for public access. This is because the IP address has no meaning apart from being an internet resource identifier. In addition, IP address is difficult to recognize because it does not represent the brand of a website. While phishers prefer to use symbols to mislead internet users, they also like to use IP address in the URL domain to trick internet users. Using IP addresses to create a website is the cheapest way because it can be achieved

by using a computer to serve 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 domain from the query website and look for the existence of IP address. If the URL domain is formed by IP address, then we assign 1 to this feature. Contrary, if the URL domain is not IP address, then this feature will be assigned zero.

3.5.3. Domain Analysis

Domain analysis is the process of analyzing the legitimacy of a website based on the rating given by third parties. For example, a website with a high rating is considered as legitimate, while a website with a poor rating is considered as phishing. For this reason, we use the rating obtained from third parties (e.g., Web of Trust service) to assess the query website. A detail of this feature is described in the next subsection.

3.5.3.1. Web of Trust

Web of Trust (WOT) [60] 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. WOT offers its services in the form of a browser plugin for free. This plugin is compatible with many browsers. However, Internet Explorer browser is not supported at this time. The plugin shows the safety rating of a website in the search results, as shown in Figure 3.12. It will show a warning message to internet users if the website's reputation is not good. In addition, WOT has released an API for developers to perform testing against malicious websites. It uses the website URL to find matching URL from the WOT database. If the URL exists in the database, then the API will return a value with scale of 0 to 100. A value of 100 indicates that the website has excellent reputation and it is safe to visit while a zero value indicates that the website has very poor reputation and it is unsafe to visit. It returns a null value if the API cannot find any information about the URL. A null value indicates that either the website does not exist or is not available in the WOT database. We use a zero value to represent the null value. The rationale is that if a website is phishing and has a rating of moderate or higher, then internet users can fall prey to this phishing website. Conversely, if the website is legitimate and not available in the WOT database, then other features of our

additional approach can overcome this limitation. The value of a website's reputation is computed based on the two parameters described as follows:

- *Reputation*. This parameter measures a website's reputation based on the ratings given by internet users. Reputation rating scale from very poor to excellent. Very poor rating means that this website is not safe to visit and excellent rating means that the website is well received by the public.
- *Category*. This category consists of several terms used to describe a website such as malware, phishing, fraud, suspicious, spam, etc. This parameter uses a number of votes received from internet users and third parties to decide the category of a website.

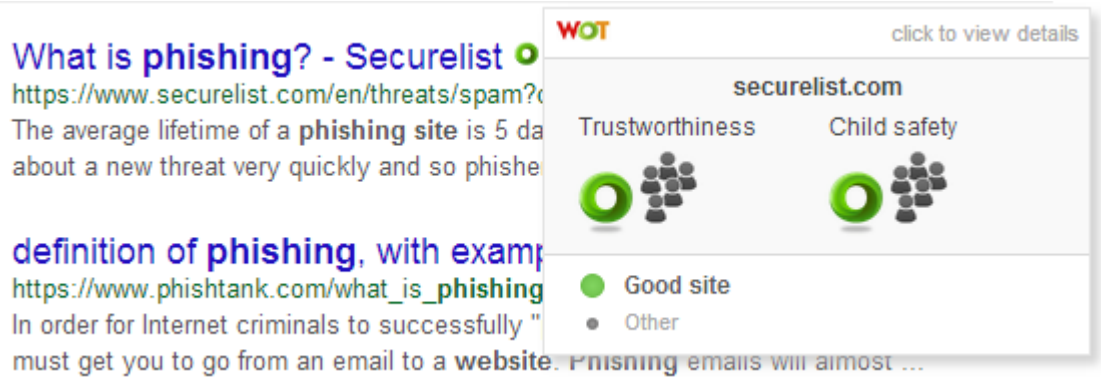


Figure 3.12: Example of WOT safety rating in the search results of Google.

In addition, WOT also includes a confidence value to assess the reputation of a website voted by the public. For example, a website with a rating of 60 is considered a good website. However, the confidence value is only 40, which means that the accuracy of the rating is a little unreliable. By default, the confidence value is provided by WOT, but it allows developers to adjust the confidence threshold so that the reputation value generated by the API for a particular website is much stringent. In other words, a high confidence threshold will result in less false positives, but it has a lower accuracy to catch poorly rated websites. The confidence value is also used to prevent voters from abusing the reputation rating system. Furthermore, the WOT employs third-party listings like malware, phishing, scam, and spam listings to investigate the website. WOT is different from website ranking system because it involves users feedback and third party (i.e., PhishTank) to verify the rating. Therefore, we conclude that the credibility of reputation value generated by the WOT is high. For this reason, we incorporate the API into our proposed technique to inspect the websites. To do so, we feed the website URL into the API.

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

Figure 3.13: Code snippet of WOT API.

Figure 3.13 shows the WOT API code snippet used to retrieve the reputation and category of a query website. The *value* 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 generates a reputation value for the website URL, we compare the value based on the scale used by WOT to evaluate this website. The reputation value used by WOT is shown in Table 3.8 where a value of 80 and higher indicates that the website receives very good feedback from the voters while a value of 19 and below indicates the website could endanger internet users. To this end, this feature is assigned 1 if the reputation is rated within the range of 0 to 19. On the other hand, if the reputation of the query website is rated above 20, then this feature is assigned zero. Moreover, this feature will be assigned 1 if it has malicious conduct described by the category regardless of reputation. The reason we assign 1 for website with a rating of 19 and below is because of WOT will display a warning message via the plugin for website with a very poor rating. Conversely, websites that receive a rating of 20 and above will not receive a warning. It only changes the color displayed in the plugin based on the ratings received. For example, the plugin will show green color for a website with good reputation (as shown in Figure 3.12).

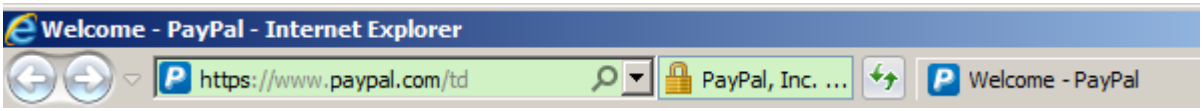
Table 3.8: WOT rating of reputation.

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

3.6. Phishing Detection and Discovery Scheme

Phishers are particularly interested in replicating the legitimate websites with little change in the content. Internet users who have no knowledge of these attacks cannot distinguish between a legitimate website and a phishing website. Usually, the appearance of different phishing websites which are targeting the same legitimate websites will look similar to each other [44]. The appearance includes textual and graphic elements such as favicon. For example, appearance of a phishing website including the favicon is identical to other phishing websites targeting the PayPal website as shown in Figure 3.14. Phishers seldom make major changes to the content of phishing websites other than the input form used to gain personal

credentials from potential victims. The main reason is to reduce the workload required as usually a phishing website has a short lifespan. Besides, major changes may arouse the user suspicion.



(a) Legitimate PayPal website.

(b) Two different phishing websites which are targeting the PayPal website.

Figure 3.14: Phishing websites targeting PayPal website has a favicon.

The preceding section (i.e., Section 3.4 and 3.5) is the theoretical discussion on each feature. The next sections will discuss the integration of these features with Google search by image API to classify websites. First, we describe the procedure on how to leverage the favicon with Google search by image API to find the identity of a website. Then, we describe the procedure on how to overcome the scenario when the favicon is absent. We include a flowchart for module A and B (Figure 3.1) in order to formulate how we integrate the proposed features for classifying a website.

3.6.1. Identification of Phishing Websites

Figure 3.15 shows the flowchart of classifying websites when the favicon is present. In order to access the favicon, we start by appending the domain of a website to a string, *favicon.ico*. For example, given, *http://www.paypal.com/index.php* we extract the domain, *http://www.paypal.com/*. Then, this domain is appended with *favicon.ico* so that it looks as such *http://www.paypal.com/favicon.ico*. Table 3.9 shows another three different examples to access the favicon. The newly formed URL will be fed to Google search by image engine using custom API to obtain the search results. We use a maximum of thirty indices of the search results returned by Google for analysis. The main reason to use this amount of search results is that we do not want to overload the Phishdentity's computation. Based on the experiments conducted, it is sufficient for Phishdentity to classify the websites correctly. We

have dedicated Section 4.5 in the next chapter to evaluate the website when different numbers of search results are used.

Table 3.9: Three examples of URLs after appended with *favicon.ico*.

Given URL	Path of the favicon
http://www.bing.com/index.php	http://www.bing.com/favicon.ico
http://www.rhb.com.my/img/file/logo.png	http://www.rhb.com.my/favicon.ico
http://www.ebay.com/signup/index.php	http://www.ebay.com/favicon.ico

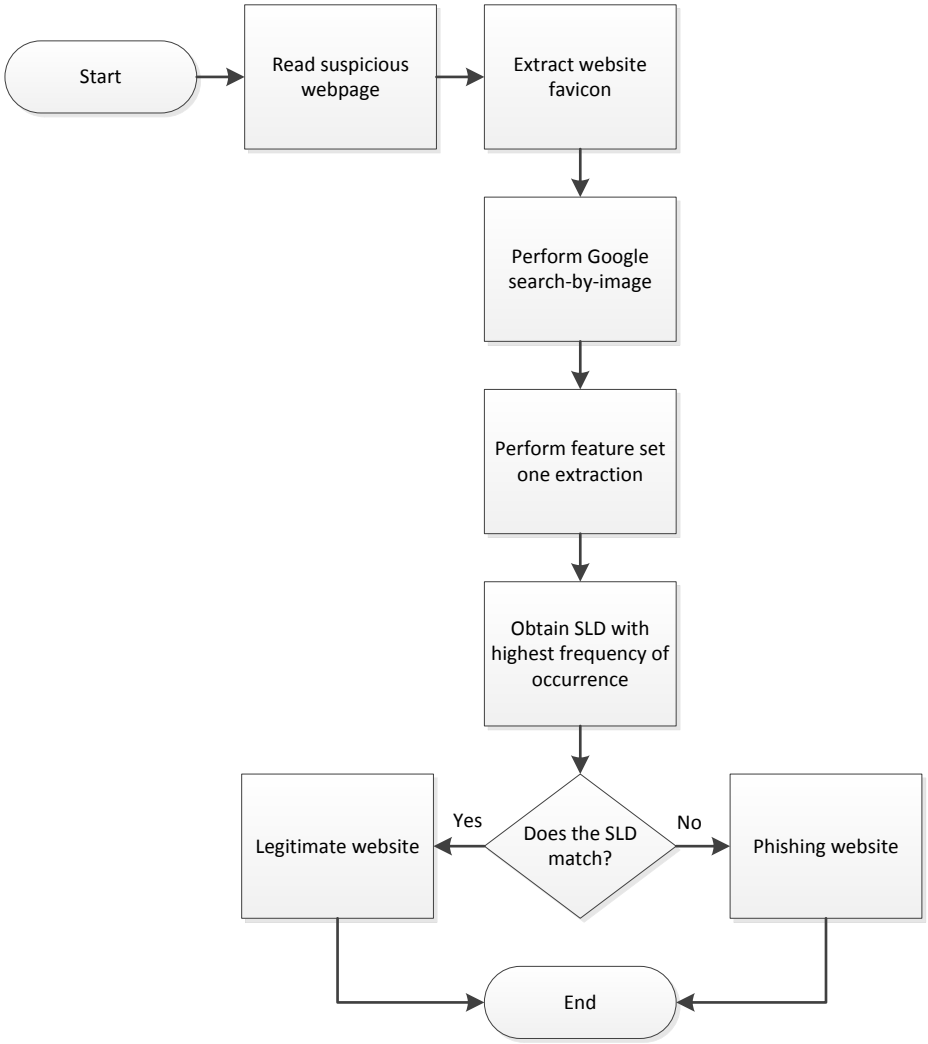


Figure 3.15: Flowchart of query website classification with the presence of favicon.

Once we obtain the search results, we apply the proposed feature set one (as discussed in Section 3.4) into the search results. First, we extract all the URLs from the entries of search results. Then, we perform URL separation where all the SLDs are recorded under the SLD table while all the paths are recorded under the path in URL table. We also extract all the

terms from the title and snippet of each entry in the search results. We record these terms under the title and snippet table. During this extraction process, terms that are bolded and occurred in the title and snippet are recorded under the highlighted text table. Next, we form the unique term from the SLD table. The unique term is the unique SLD extracted from a list of entries recorded in the SLD table. From here, a list of unique terms is produced. We use this list to find matching terms across each table (the detail has been discussed in Section 3.4). For each matching term in each table, the frequency matches is counted. Hence, we have created a list of unique terms with different frequency values. To formulate the matching process, we use the following equations to calculate weighted frequency value for the unique terms across each table.

$$h_i^{sld} = \frac{f_i^{sld} * w_{sld}}{F_{sld}}, \quad (1)$$

$$h_i^{path} = \frac{f_i^{path} * w_{path}}{F_{path}}, \quad (2)$$

$$h_i^{tns} = \frac{f_i^{tns} * w_{tns}}{F_{tns}}, \quad (3)$$

$$h_i^{hlt} = \frac{f_i^{hlt} * w_{hlt}}{F_{hlt}}, \quad (4)$$

Where $h_i^{sld,path,tns,hlt}$ refers to the weighted frequency value of *i*-th unique term for the four features. While $f_i^{sld,path,tns,hlt}$ is the frequency value of *i*-th unique term in which *i* is from the list of unique terms. $F_{sld,path,tns,hlt}$ is the total frequency in one feature. $w_{sld,path,tns,hlt}$ is the weight assigned to each feature (as shown in Table 3.10) and the value is determined empirically. We assign fourth feature to have the highest weight because the highlighted text are usually has high tendency to represent the identity of the favicon. Based on the experiments conducted, we observed that this feature has very low occurrence frequency in the search results. It only occurs when the query favicon is truly matched with the image content stored in Google image database. First and third features are assigned with modest weight mainly because of the frequency of identity depends on Google search engine index. These features are important to our proposed method, but it is slightly lower in the level of importance compared with the fourth feature. Second feature is assigned with the lowest weight because phishers can always make changes to the path without affecting the whole

website. Thus, the frequency of its occurrence in the search results is not consistent compared to other features in the first layer.

Table 3.10: Weight assigned to each feature in the first layer.

Feature, *h*	**Notation**	**Weight, *w***
Second-level domain	w_{sld}	20
Path in URL	w_{path}	10
Title and snippet	w_{tns}	30
Highlighted text	w_{hlt}	40

After getting the weighted frequency of each unique term from each table, we need to calculate the final frequency value of each unique term from all the tables. To do so, we use the following equation to obtain the final frequency value.

$$H_i = \frac{h_i^{sld}+h_i^{path}+h_i^{tns}+h_i^{hlt}}{w_{sld}+w_{path}+w_{tns}+w_{hlt}}, \quad (5)$$

Where H_i refers to the final frequency value of *i*-th unique term.

Once we have computed the final frequency of all the unique terms in the first layer, we need to amplify the final frequency value of a unique term that corresponds with the SLD of query website in second layer. To do so, we extract the SLD of query website to find matching from the list of unique terms. If the list contains the SLD of query website, then we increase the final frequency of this unique term by five percent. For example, if the final frequency for term "*paypal*" in the first layer is 20 and the term is also corresponds with the SLD of query website, then the new final frequency for that term will be 21 in the second layer. If the list does not contain the SLD of query website, then we append the SLD into the list of unique terms to count the frequency based on the features of first layer. Next, we need to obtain a unique term with the highest final frequency value from the list. To do so, we use the following formula.

$$UT_i = arg\max_i \{H_i\}, where\ i = 1, 2, 3, \dots, n. \quad (6)$$

UT_i denotes the *i*-th unique term and n is the last index of *i*-th unique term. Therefore, the unique term with the highest final frequency value 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 score of 100. Otherwise, we will assign a score of zero as below:

$$S_1 = \begin{cases} 100, & if\ UT_{max} \neq SLD_{query} \\ 0, & otherwise \end{cases}, \quad (7)$$

UT_{max} denotes the unique term with highest final frequency value (also refers to the identity of a query website) and SLD_{query} is the SLD of a query website. The reason we use a score of 100 for a website that does not match the identity is to facilitate the calculation of final integration with Module B. We have dedicated Section 3.6.3 to discuss how we use the score to classify websites.

3.6.2. Identification of Phishing Websites with the Absence of Favicon

Our initial concept of Phishdentity is to leverage the website favicon with Google search by image to uncover the identity of a website. However, we discovered that Phishdentity will fail if the favicon is missing. Therefore, we proposed an additional approach which is based on the URL of a website for classification. URL analysis is lightweight and easy to access for assessment compared to other approaches. While phishers can conceal malicious content on the website, they cannot hide the URL or the IP address from the public. There are many studies conducted to detect phishing websites based on URL [17, 18, and 38]. These studies produce fairly good results in the classification. For this reason, we proposed a total of five features based on the URL to detect phishing websites. These features are suspicious URL, dots in domain, age of domain, IP address and WOT. The theoretical explanation has been discussed in Section 3.5.

Figure 3.16 shows the flowchart of classifying a query website when the favicon is missing. We start by extracting the full URL of a query website. The reason we choose to extract the full URL of a query website is because many phishing websites only have a single webpage for public access while other webpages are being blocked or leading to different websites. That being said, we want to extract URL address that leads us to the right destination. For example, the URL, *http://www.pay.pal.site.com/loginnotices/index.php?q=user* will lead the victims to visit PayPal phishing website.

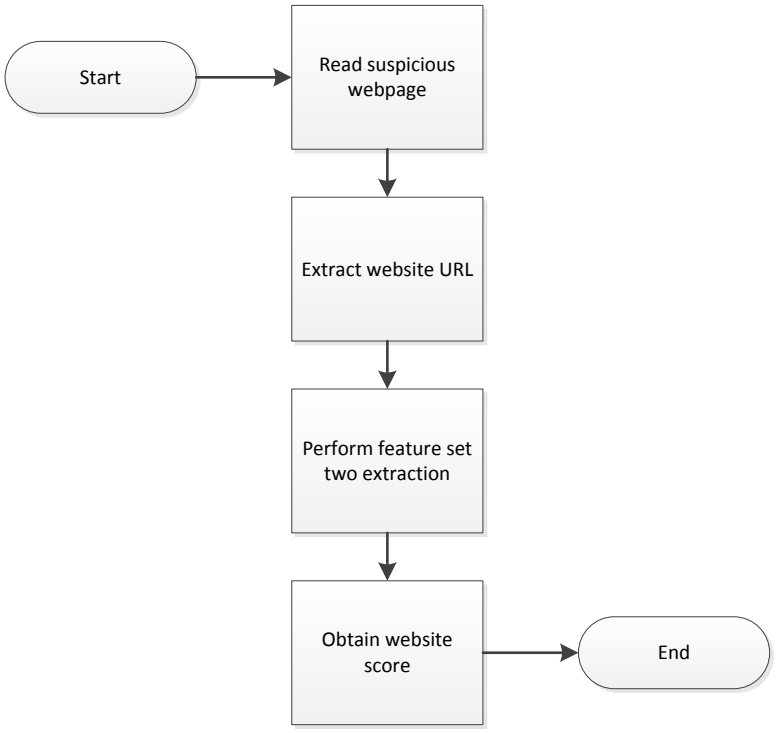


Figure 3.16: Flowchart of query website classification with the absence of favicon.

Once we have extracted the URL from a query website, we perform feature extraction two from the URL. The extracted feature set two will be used for analysis. First, we use the WOT API to retrieve the reputation and category of the extracted URL. To do so, we insert the extracted URL into the respective parameter in order to use the API (refer to Figure 3.13). If the API returns a reputation value higher than 20 and it is not listed as a threat based on the category information, then we will assign zero to this feature. However, if the reputation value is equal or less than 20 and it is not listed as a threat category, then we will assign 1 to this feature. If the extracted URL is listed as a threat category, then this feature will be assigned 1. Information received from the category is used as an indicator to check against the legitimacy of a website regardless of its reputation is rated as either very poor or excellent.

Table 3.11: Weight assigned to each feature of module B.

Feature, *h*	**Weight, *w***
Web of Trust	40
Age of domain	30
Suspicious URLs	10
Dots in URL	10
IP address	10

Next, we inspect the extracted URL for suspicious symbol. To this end, if the extracted URL contains an at-sign (@) or a dash (-) symbol, then we assign 1 to this feature. Otherwise, it is assigned zero. After that, we examine number of dots based on the extracted URL domain. To do so, we extract the domain from the URL and count the numbers of dot that are present. If the extracted URL domain contains five or more dots, then we assign 1 to this feature. Otherwise, it is assigned zero. Then, we apply age of domain feature to check the age of the extracted URL domain. If the extracted URL domain is older than 30 days, then we assign zero to this feature. If the age of the extracted URL domain is equal or less than 30 days, this feature will be assigned 1. Lastly, we review the query website to extract the IP address feature. This feature determines whether the extracted URL domain is formed by IP address. If the extracted URL domain is an IP address, then we assign 1 to this feature. Otherwise, it is assigned zero. We use the following equation to formulate the calculation.

$$S_2 = \sum w_i * h_i, \tag{8}$$

h_i is the value obtained from i-th feature in which i denotes as a list of features in the additional approach. w_i refers to the weight assigned for each feature and it is determined empirically as shown in Table 3.11. S_2 is the score for additional approach. WOT feature is assigned with the highest weight mainly because it can display a website ranking and the ranking order can change based on votes received from the public and also information obtained from third parties. Age of domain feature is assigned as the second highest weight because every website owner must register their website with a hosting provider to obtain a meaningful domain name. Thus, they cannot easily fake the age of the website. We give a higher weight to WOT than age of domain because it uses active information like user's feedback and third party listings to validate the legitimacy of the website. Suspicious URL, dots in domain and IP address are assigned with the same weight. Mainly, they are the local features of the URL and the data is not verified by any third parties. Therefore, we are of the opinion that the interests of suspicious URLs, dots in the domain and IP address is a little lower than the WOT and domain age.

3.6.3. Final Integrated Phishing Detection System

In order to classify the website, we combine the scores obtained from Eqs. (7) and (8). However, it is crucial to rationalize the scale of the final score so that both equations can

contribute to classify a website. To achieve this, we use a scale of 100 to represent the final score. In other words, the scores obtained from Eqs. (7) and (8) will be amended so that the total score is equal to 100. We use a threshold of 50 as the baseline to determine the optimum score for Eqs. (7) and (8). The reason why we have to find the optimal score is due to additional approach can be used to offset Phishdentity inability to classify websites without favicon. We have dedicated Section 4.7 to experiment with different scores given to Eqs. (7) and (8). We use the following equation to calculate the final score of a query website.

$$FinalScore = S_1 + S_2, \quad (9)$$

S_1 is the score obtained from Eq. (7) and S_2 is the score obtained from Eq. (8). $FinalScore$ is used to determine the legitimacy of the query website. If the $FinalScore$ exceeds τ threshold, then the query website is classified as phishing. Otherwise, the query website is classified as legitimate. We have dedicated Section 4.7 to experiment with different variants of the threshold. This experiment will provide the optimum threshold. There are three observations that lead us to believe that the website favicon is useful in identifying the identity carried by a website:

- Legitimate website domain name will appear in the search results when the favicon is fed to the Google search by image while the phishing website domain name with the same favicon will not show up in the search results. Usually, this is due to legitimate website will receive positive feedback from internet users. In addition, legitimate website will follow the prospects defined by the Google search engine when building a website.
- Phishers often mimic other phishers work when targeting the same legitimate website (possibly this is the result of using phishing toolkit). They make many copies of phishing websites with little modification on input forms to distribute to the victims. Therefore, most phishers will replicate the entire content including the favicon directly from the legitimate website instead of creating a new website content. Creating a new one requires bigger effort and is time consuming.
- While there are many websites that allow owners of other websites to park their favicon with inbound or outbound links attached, but the favicon that is indexed by the Google crawler would display the domain name of the actual favicon owner at least once in the search results regardless of how many websites have shared the favicon. In addition, the query image from a legitimate website is unique and

according to the standards defined by the World Wide Web Consortium (W3C). Therefore, the actual owner of the favicon will be indexed higher than other websites that share the favicon on their webpages.

Likewise, there are three observations that lead us to believe that a solution based on the website URL can be effective in detecting phishing websites especially when the favicon is missing:

- WOT is a website reputation rating system that receives votes from the community to evaluate a particular website. WOT leverages the information received from third parties such as PhishTank to examine the reputation of a website. In addition, WOT has a large collection of website database accumulated from many years and it also collaborates with different security companies to enhance efficiency in evaluating websites. Therefore, the WOT can be very useful in preventing internet users from victimized by phishers.
- Age of domain feature is important to our proposed technique because it provides an insight about the age of a website. A legitimate website usually has a long lifespan while a phishing website has a very short lifespan. It is also accordance to the report released by APWG [59] in which the average lifespan for phishing websites are one week. Moreover, this feature helps improve our proposed technique in classifying the suspicious website.
- Regardless of how much effort phishers put to conceal the phishing websites from anti-phishing detection, they cannot hide the URL or IP address from the public as it is the main gateway for the internet browser to reach the phishing website.

Chapter 4

Experimental Results and Analysis

This chapter demonstrates the assessment of Phishdentity through different experiments to evaluate the performance and the constraints imposed. We start by introducing the tools and the programming language needed to implement the prototype. Then, we explain the procedures used to collect the data for the experiments. We also describe the design and the purpose of each experiment. In addition, we also determine the optimum setting for parameters used for the prototype. Likewise, we also design a set of experiments to evaluate the prototype for websites without the presence of favicon. Finally, we discuss the limitations imposed on the prototype.

4.1. Prototype Implementation

We have implemented a prototype of Phishdentity and it is written in C# language using the Microsoft Visual Studio 2010 Professional Edition. The Microsoft Visual Studio is an integrated development environment (IDE) designed for easy to build, debug and deploy all kinds of applications. The prototype consists of two parts and of 1800 lines of codes. First part of the prototype employed the Google search by image API to return a list of websites related to the search favicon. From here, we perform feature extraction to obtain the necessary data. Second part of the prototype integrates the WOT API to return the reputation score and the category for a given URL. We also use a third party service such as the WHOIS website [61] to obtain the age of a given URL. Next, we investigate the URL to obtain additional features such as suspicious symbol, numbers of dot, and IP address. All of these features are necessary for the enhancement of classification. Table 4.1 shows the specification of hardware and software used to implement the prototype for Phishdentity.

Table 4.1: Hardware and software specification.

Hardware	
Component	**Specification**
Processor	AMD quad-core processor
RAM	4 GB of memory
Video card	Radeon HD 7480D
Hard disk	500 GB of storage space
Internet connection	1 Mbps downlink / 512 Kbps uplink
Software	
Component	**Specification**
Operating system	Windows 7 Home Premium
Development platform	Microsoft Visual Studio 2010 Professional Edition

4.2. Data Collection

We have collected a total of 500 phishing websites and 500 legitimate websites within six days, from January 26, 2014 to January 31, 2014. The phishing websites were selected directly from PhishTank within eight hours of being reported by the community members while the legitimate websites were selected manually from Alexa top 500 global websites [62] ranging from e-commerce website to personal blog. During the collection of data from Alexa top 500 global websites, we observed that there are several websites that do not have the presence of favicons. Similarly, we noticed that there are some phishing websites that do not have the presence of favicons.

Table 4.2: Collection of data from Alexa top 500 global websites and PhishTank.

Alexa top 500 global websites		PhishTank	
Presence	**Absence**	**Presence**	**Absence**
477 (95.4%)	23 (4.6%)	487 (97.4%)	13 (2.6%)

Table 4.2 shows the number of websites (500 legitimate and 500 phishing) that has the presence of favicon. However, all of these favicons are yet to be verified. We will use the prototype that was built earlier to reveal the favicon identity. Nevertheless, Figure 4.1 shows the categorization of websites in Alexa top 500 global websites. From this figure, technology, Internet and search engine related websites accounted for 16% each of the 500 legitimate websites. While news and media related websites accounted for 13% of the collection. Auction and shopping related websites are the fourth largest in the collection in which it

accounted for 8% of the collection. In addition, business, file sharing, and social network related websites accounted a total of 18% of the collection. Financial related websites only accounted for 5% of the collection. Websites that accounted less than 5% of the collection are grouped under the category of "Others".

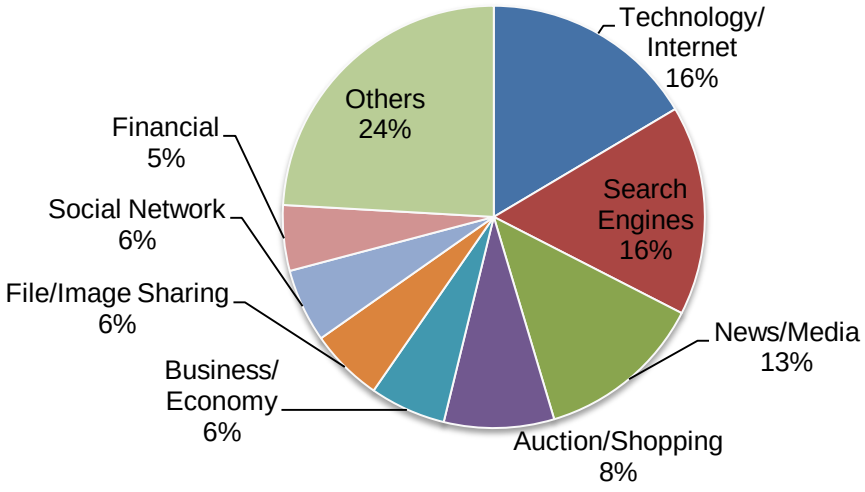


Figure 4.1: Categorization of Alexa top 500 global website.

On the other hand, Figure 4.2 shows the categorization of 500 phishing websites. From this figure, financial related websites accounted 27% of the 500 phishing websites. While auction and shopping related websites accounted for 22% of the collection. Social network related websites are also one of the main phishing targets in which it accounts 15% of the collection. File-sharing related websites are also favored by phishers in which it accounted for 10% of the collection. Game related websites that involve with money transactions also become one of the targets of phishing in which it accounted for 8% of the collection. Technology and Internet related websites only accounted 6% of the collection. Pornography and travelling related websites each accounted for 5% and 2% of the collection. Websites that accounted less than 2% of the collection are grouped under the category of "Others".

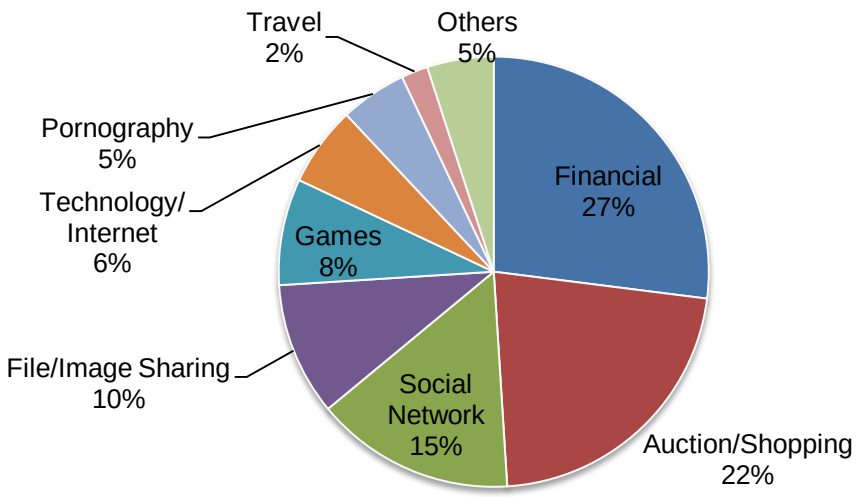


Figure 4.2: Categorization of 500 phishing websites.

4.3. Experimental Outline

We have designed four sets of experiments to evaluate the performance of Phishdentity. These experiments are also used to find out the optimum settings of Phishdentity to detect phishing websites. Specifically, we want to examine the settings of Phishdentity from different sets of experiments to achieve high detection rate in classifying websites. For instance, a phishing website is indeed classified as phishing while a benign website is indeed classified as legitimate. Moreover, we also look into the average duration needed to classify a website. This aspect of the research can indicate the effectiveness of our proposed technique compared to other anti-phishing solutions. In other words, an anti-phishing approach with high detection rate is deemed as effective solution if the average duration needed to classify a website is low. It is also mentioned in the literature review that time is an important criteria to evaluate the effectiveness of an approach [38]. Therefore, an efficient feature extraction can accelerate the classification of websites. Nevertheless, the experiments are outlined as follows:

- Experiment 1: Experiment 1 covers the basic features of Phishdentity. This experiment uses the default parameter of Google search by image API to return the search results. We use a total of 500 legitimate websites and 500 phishing websites to evaluate the performance. We start by testing the performance of first layer. First layer consists of SLD, path in URL, title and snippet, and highlighted text. Then, we

integrate the first layer (i.e., a list of unique terms with different frequencies) into second layer for performance testing. Second layer utilizes the query website domain name for classification. In particular, we are interested to observe how the second layer affects the detection rate of Phishdentity when it is combined with the first layer. We use the terms "Basic Phishdentity" and "Basic Phishdentity and domain name" to refer to the setup used in the experiment. Basic Phishdentity uses the first layer while Basic Phishdentity and domain name use the first and second layers to classify websites. From here, we pick the setup that offers the highest performance for classification and apply it into second experiment for further testing.

- Experiment 2: Experiment 2 is designed to assess the performance of Phishdentity when different number of search entries is used. More precisely, we change the parameter of the API to return different number of search entries (i.e., top 1, 10, 30 and 50 entries of search results) for the experiment. We use a total of 1,000 websites in this experiment which comprises of 500 legitimate and 500 phishing. This experiment helps finding out the optimum number of search entries required to reveal the identity of a website. In addition, this experiment helps Phishdentity to save unwanted computational power in order to classify a website. Zhang et al. [19] also use the same experiment to evaluate the performance of CANTINA when different number of search entries is used. We will choose the number of search entries (top 1, 10, 30 or 50 entries), which offers the highest detection rate and the highest detection speed based on the results of this experiment. Then, we use this setup for Experiment 3 for further testing.
- Experiment 3: This experiment is designed to examine the performance of Phishdentity about the absence of favicon. This experiment is important to our proposed technique because it reveals the potential of Phishdentity in classifying websites. For this reason, we design two test beds for this experiment. First test bed will be used to assess the performance of Phishdentity without additional approach. Second test bed will be used to evaluate the performance of Phishdentity when it is combined with an additional approach using the threshold function. Based on our preliminary observations, additional approach can improve Phishdentity's weakness when favicon is absent. This experiment will produce the final version of Phishdentity. We use it to perform benchmarking with other anti-phishing solutions.
- Experiment 4: In order to show the effectiveness of Phishdentity in classifying websites, we have decided to perform benchmarking with other anti-phishing

solutions. The benchmarking is evaluated based on two criteria, namely the detection rate and detection speed. The detection rate is the amount of correctly classified websites while the detection speed is the duration required to classify a website. In this experiment, we choose CANTINA [19] and GoldPhish [45] for benchmarking with Final Phishdentity. CANTINA is a content-based anti-phishing approach that utilizes the term frequency-inverse document frequency (TF-IDF) technique. It extracts five important keywords from the website and feeds it to the Google search engine. Then, it searches for a matching domain name from the search results to determine the legitimacy of the website. On the other hand, GoldPhish is an image-based anti-phishing approach that utilizes the optical character recognition (OCR) technique. First, it uses a predefined resolution to capture a screenshot of the website. Then, the OCR extracts the textual information from the captured screen to feed with the Google search engine. Next, it searches for a matching domain name from the search results to determine the legitimacy of the website. We use 500 legitimate websites and 500 phishing websites for the benchmarking.

All of these experiments are described in the next section where it covers the problems of the experimental study, hypothesis, procedures, materials, results, analysis and conclusion. The experimental results are displayed using the following measurement metric:

- True positive (TP). A phishing webpage is correctly classified as a phishing webpage.
- True negative (TN). A legitimate webpage is correctly classified as a legitimate webpage.
- False positive (FP). A legitimate webpage is misclassified as a phishing webpage.
- False negative (FN). A phishing webpage is misclassified as a legitimate webpage.
- F-score (F_1). It shows the classification accuracy of the model and the equation was composed by $F_1 = \frac{2TP}{2TP+FP+FN}$.

4.4. Experiment 1 - Evaluation of Phishdentity on Google search by image API

Research problem: Phishdentity relies on Google search by image API to return information about the website favicon. This information is used with the proposed features to uncover the favicon hidden identity. However, the effectiveness of this API in phishing detection is not clear. Based on our early observation, the API can return different search results when feeding with the same favicon. Although the search results still contain the clues about the favicon, but it can diminish our proposed technique to obtain the correct identity. Therefore, we design this experiment to evaluate the performance of proposed features with the Google search by image API.

Hypothesis: Phishdentity can reveal the identity of favicon using Google search by image API.

Procedures: We designed two test beds for this experiment to demonstrate the effectiveness of Google search by image API to return information about the website favicon. Before starting the experiment, we configure the API to return all the search entries when it is fed with the favicon. Procedure on how we prepare the test bed is described as follows:

- Test bed 1 (Basic Phishdentity): We design this test bed to adopt the features of first layer. The first layer consists of SLD, path in URL, title and snippet, and highlighted text. First, we feed the API with the favicon. Then, we extract all the SLDs from the entries in search results. Next, we form a list where it represents the unique SLD extracted from the entries. This list will be used to count and record the frequency for each matching term found in the features of first layer. After that, we need to find a unique term with the highest frequency from the list. To do so, we add up the frequency of each unique term for each feature in the first layer. From here, we can find a unique term with the highest frequency and it will be deemed as the identity of the query website. This identity will be used to compare with the SLD of query website. Our proposed technique will classify the query website as legitimate if the identity corresponds with the SLD of query website. Otherwise, it is classified as phishing.

- Test bed 2 (Basic Phishdentity and domain name): This test bed inherits the setup of first test bed except that we integrate the second layer into the test. When we obtain the frequency of first layer, the SLD of query website is used to find a matching term from a list of unique terms. If the SLD is found on the list, the total frequency for that unique term is increased by five percent. Otherwise, we retain the old frequency and append the SLD to the list. Newly added SLD is acted as a unique term and it will be used to find matching terms from the search results. Thus, a unique term with the highest number of frequency from the list is deemed as the identity of the favicon. We use this identity to compare with the SLD of query website. If they are matched, then the Phishdentity will classify it as a legitimate website. Otherwise, it is classified as a phishing website.

Material: We use a total of 1,000 websites (500 legitimate and 500 phishing) for each test bed.

Results: We compile the results of the two test beds into a table as shown in Table 4.3.

Table 4.3: Evaluation results of Phishdentity under different test beds.

Test bed	TP (%)	TN (%)	FP (%)	FN (%)	F_1
Test bed 1	95.80	86.00	14.00	4.20	0.9133
Test bed 2	97.40	94.60	5.40	2.60	0.9606

Table 4.3 shows the results of each test bed using 500 legitimate websites and 500 phishing websites. Based on Table 4.3, the first test bed has the highest error rates (i.e., FP and FN) compared to the second test bed. However, the second test bed has shown some improvement in the classification after adding domain name amplification feature into the test.

Analysis: We perform analysis on each test bed to understand the strengths and weaknesses of the proposed features.

- Test bed 1: This test bed has produced a fair amount of false positives when we apply the features of first layer into the search results. We consider these features are effective against phishing websites despite there are some phishing websites managed to bypass the detection. We believe that the detection rate for phishing websites can

be further improved by adding additional approach into the test. Nevertheless, the phishing websites which managed to bypass the detection exist in the last few entries of the search results. Thus, the test bed will falsely identify the identity if the frequency is higher than other identities. The Google search by image API includes the identity in the search results is due to the favicon is identical to the legitimate favicons in terms of image content. Similarly, some of the phishing websites that copy the favicon directly from the legitimate websites are able to avoid the detection. There are also some phishing websites that are hosted under legitimate domain. This act can mislead the API to return information related to the phishing website. On the other hand, the legitimate websites that are misclassified are due to the missing of query website domain name in the entries of search results. We noticed that the identity is hiding in other elements of the search results. For example, the PayPal identity can be found in title and snippet feature but it is missing from SLD feature. We also noticed that the API did not return sufficient information to some of the favicons. It sometime return search results with half of the information in favor of another identity. Hence, the test bed will obtain unrelated identity with higher frequency. The API also can return incorrect information when feeding with a wrong version of the favicon. This happens when the query website has a newer version of the favicon than the Google image database. Although this issue has contributed to the number of false positives, but we believe that the new favicon will be available in the Google image database soon. It is also in accordance with the frequent update of Google web crawler to the database. In addition, we noticed that there are few legitimate websites with the absence of favicon, which contribute to the high numbers of false positives. We will further analyze this issue in the following experiment (i.e., Experiment 3) where an approach that is based on the URL is applied to overcome the absence of favicon. High numbers of false positives are also contributed by some legitimate websites that restrict access to the favicon through the API. However, this issue does not occur on phishing websites during the test.

- Test bed 2: This test bed has shown some improvements in website classification when we integrate second layer into the test. The improvement is noticeable in classifying legitimate websites in which it reduces a total of 8.6% in false positive numbers. The decline in false positive numbers is due to the test bed being able to locate the identity in the entries of search results after adding with a domain name. Similarly, it also enables the test bed to locate the identity even when the search

results contain little information about the favicon. However, the number of false positives is still high. There are few factors that contribute to the high number of false positives. First, there are a total of 23 websites without the presence of favicon. Therefore, it contributes a total of 4.6% to the false positive numbers. The API cannot return any information relating to the query website if the favicon is absent. Second, the Google image database is yet to update all the favicons that are different from the original websites. This will cause the API to return unrelated information. Third, there are some websites that restrict access to the favicon when we perform the test under this test bed. All of these factors contribute to the high number of false positives. On the other hand, this test bed also has shown some improvement in detecting phishing websites. It reduces by 1.6% from 4.2% to 2.6% false negative numbers. The decline in false negative numbers is due to the search results contain a lot of information about the legitimate favicons rather than the imitated favicon. We realized that the API has the ability to return more accurate information about the favicon from time to time, this includes the imitated favicon. Nevertheless, we believe that the features of first layer and Google search by image API are sufficient to detect phishing websites. Furthermore, we have proposed a solution that is based on the URL for classification. It will be used to improve the classification and overcome websites without the presence of favicon.

Conclusion: Phishdentity achieves the highest detection rate when it uses both layers. It works best when the query website is indexed by Google web crawler. This enables the Google search by image API to return sufficient information relating to the favicon. In addition, the Google search by image API is unlikely to include phishing entries in the search results unless phisher can obfuscate Google web crawler when indexing. We apply the setup of second test bed (Basic Phishdentity and domain name) to the next experiment.

4.5. Experiment 2 - Evaluation of Phishdentity on Search Results

Research problem: Previous experiment used all the search entries to reveal the favicon identity. However, the process is computationally intensive and time consuming. We believe it is bad for the user if the anti-phishing tools have high detection rate but poor in detection speed. Victims can fall into the trap of phishing if the anti-phishing tools could not give an

immediate answer to the victim. Therefore, we would like to examine the performance of Phishdentity using different number of search entries. The results of this experiment will help us to determine the number of search entries required to classify a website.

Hypothesis: Phishdentity performance improved as the number of search entries is increased.

Procedure: We design four different test beds to examine the efficiency of Phishdentity to classify a website using different number of search entries. Procedure on how we prepare the test bed is described as follows:

- Test bed 1 (Top 1): In this test bed, we configure the Google search by image API to return only the top 1 entry of search results. We apply the proposed feature set one to extract first and second layer. Next, we need to identify a unique term that has the highest number of frequency from a list of unique terms. This unique term is considered to be the favicon identity and will be used to compare with the SLD of a query website. If the unique term corresponds with the SLD, Phishdentity will flag it as a legitimate website. Otherwise, it is flagged as a phishing website. In addition, we measure the duration needed to classify a website. We start recording time when the API feeds the favicon until the query website is classified. We repeat this process for the whole dataset. Then, we sum the time elapsed of each website. We calculate the average by dividing total time elapsed with total dataset and record the average.
- Test bed 2 (Top 10): This test bed has the same setup as the first test bed except that we configure the API to return only the top 10 entries of search results.
- Test bed 3 (Top 30): This test bed also has the same setup as the first test bed. However, we configure the API to return only the top 30 entries of search results.
- Test bed 4 (Top 50): Similarly, this test bed has the same setup as the first test bed except that the API returns only the top 50 entries of search results for assessment.

Material: We use a total of 1,000 websites (500 legitimate and 500 phishing) for each test bed.

Results: Table 4.4 shows the efficiency of Phishdentity to classify a website. Table 4.4 (a) shows the detection rate of each test bed using different number of search entries. Table 4.4

(b) shows the average duration needed to classify a website based on different number of search entries.

Table 4.4: Evaluation of Phishdentity based on different number of search entries.

(a) Detection rate.

Test bed	TP (%)	TN (%)	FP (%)	FN (%)	F_1
Test bed 1	99.60	37.40	62.60	0.40	0.7594
Test bed 2	98.80	79.80	20.20	1.20	0.9019
Test best 3	97.40	94.60	5.40	2.60	0.9606
Test bed 4	97.40	94.60	5.40	2.60	0.9606

(b) Detection speed measured in seconds.

Test bed	Legitimate	Phishing
Test bed 1	2.13	2.03
Test bed 2	7.49	7.88
Test best 3	10.35	11.12
Test bed 4	13.80	13.40

Table 4.4 shows the results of four different test beds using different number of search entries. The first test bed has a pretty high number of false positives with a relatively low number of false negatives. The number of false positives has decreased after setting the API to return only the top 10 entries of search results, but it decreases slightly the accuracy of detecting phishing websites (as shown in test bed 2). The third test bed has shown some improvement in classifying legitimate websites when we use the top 30 entries of search results. However, this test bed decreases the accuracy of detecting phishing websites. Nevertheless, we did not see any improvement in the fourth test bed when we set the API to return the top 50 entries of search results. Instead, it takes a longer period of time to compute the result. We noticed that there is not much difference in detection rate for legitimate websites and phishing websites. This is due to the API being able to find sufficient information about the favicon from Google image database. Therefore, the detection speed is increased when we increase the number of search entries.

Analysis: We perform analysis on each test bed to observe how it reacts with the dataset based on different number of search entries used.

- Test bed 1: This test bed performs poorly in classifying legitimate websites when we set the API to return only the first entry of search results. It has high chance to cause false alarm on legitimate websites. However, it achieves the highest detection speed compared to three other test beds. It also achieves the lowest number of false negatives but this setup is not the ideal solution to classify a website. Phishing websites that are managed to bypass the detection are those who hosted under the legitimate domain. Hence, the API can include false identities in the search results. Nevertheless, misclassification of legitimate websites is very costly. Not only it damages the company reputation, but it also breaks the trust between the company and consumers. We observed that the API did not always return the query website in the first entry of search result. We believe that it is due to the internal indexing mechanism used by the Google search by image engine to display the search results. The identity of a query website might be hidden in other entries of search results. Therefore, using only one search entry to reveal the favicon identity is insufficient to our proposed features. Moreover, one search entry contains very little useful information for feature extraction.
- Test bed 2: This test bed has shown great improvement in classifying legitimate websites when we set the API to return only the top 10 entries of search results. But it reduces the accuracy in detecting phishing websites. Contrary, it requires a longer period of time to compute the result compared to the detection speed of first test bed. This is also due to the number of search entries has increased. In other words, the search time of 10 entries has increased the time required for computation. We believe that the performance of this test bed can be further enhanced in order to achieve high detection rate with moderate detection speed. The reason is that a browser does not always display a webpage immediately. It depends on the computer and networking specification to render a webpage before appearing on the monitor screen. Hence, we think it is reasonable that our proposed technique has a slight delay in the computation. Nevertheless, we noticed that the top 10 entries of search results have high chance to return the query website domain name in the search results. In addition, the top 10 entries of search results have much useful information for feature extraction. This setup allows the test bed to perform a search in a wider dimension. Thus, it reduces the amount of misclassified legitimate websites. However, this setup also causes the API to include more false identities in the search results. We argue that the number of

false negatives under this setup is tolerable for end users because Google and other search engines would be able to detect it soon.

- Test bed 3: This test bed has shown much improvement in classifying legitimate websites when we set the API to return the top 30 entries of search results. However, it has caused a decrease in the accuracy of detecting phishing websites. In addition, this test bed takes a longer period of time to compute the result as compared to the detection speed attained by first and second test beds. The increase in period of time is due to the number of search entries increased to 30. This also shows that the search time increases as the number of entries increases. We feel that the loss of 0.8% accuracy in detecting phishing websites is acceptable, given the number of false positives was reduced by 14.8% from 20.2% to 5.4%. Furthermore, the increase of period of time for this test bed is smaller than the increase of period of time for second test bed when we add 20 more entries to the search. Based on our observation, the top 30 entries of search results have higher chance to return the query website domain name compared with the top 10 entries. It also has a bigger dimension to extract useful information for classification. Based on the result shown in Table 4.4, we believe that the top 30 entries of search results are sufficient for our proposed features. It is also due to the Google search by image engine that displays the entries based on the degree of image similarity that corresponds to the image stored in Google image database. Hence, the first entry of search results usually has the highest level of image similarity. Nevertheless, this test bed failed to detect some of the phishing websites. We observed that the misclassified phishing websites host their domain under legitimate domain to bypass the detection. We will integrate additional approach in the next experiment to improve the results.
- Test bed 4: This test bed does not show any improvement in the classification when we set the API to return the top 50 entries of search results. Instead, it achieves the same detection rate as the third test bed. Moreover, this test bed takes the longest period of time to classify a website as compared to the detection speed achieved by the previous three test beds. This shows that the detection rate is not only dependent on the size of the search results. Furthermore, the API can include more false identities in the search results if the number of search entries is set too high. We need to look for another aspect if we want to improve the detection rate. For example, a website URL can be used as a feature to determine the legitimacy. We noticed that the API does not always return the number of search entries in full. Sometimes the search

results contain less than 30 entries. We believe that it is due to the Google search by image engine that does not extract sufficient image data from the favicon. Hence, the API will return less than 50 entries if the favicon contains very little image data. Most of the time, this incident happens on phishing websites where the phishers are known to imitate the favicon of legitimate websites.

Conclusion: Phishdentity achieves the highest detection rate when we set the API to return the top 30 and 50 entries of search results. However, the top 50 entries require a longer time to classify a query website compared with the top 30 entries. First and second test beds have high detection speed, but they perform poorly against legitimate websites. Hence, we select the top 30 entries of search results for use with Phishdentity. This setup is applied to the next experiment.

4.6. Experiment 3 – Evaluation of Phishdentity with the Absence of Favicons

Research problem: Previously, we were unable to classify a legitimate website when the favicon is absent. It will cause the Google search by image API to return irrelevant information about the favicon. We realized that there are 23 websites from Alexa top 500 global websites does not have a favicon in it. In other words, it contributes a total of 4.6% to the false positive number. Figure 4.3 shows an example of a website using the default browser favicon instead of the website logo. We argued that this number is quite high because it is very costly for a company to regain trust from the consumers and business partners if the reputation is damaged. Hence, this experiment is designed to investigate the additional approach to overcome websites without the presence of favicon.

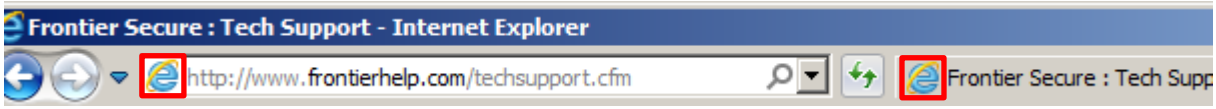


Figure 4.3: Example of a website using the default browser favicon.

Hypothesis: An approach based on the URL can increase the detection rate for Phishdentity in classifying websites, with or without the presence of favicon.

Procedure: Two test beds are designed in this experiment. The first test bed is designed to demonstrate the effectiveness of Phishdentity without using the additional approach. The second test bed will demonstrate the effectiveness of Phishdentity with the additional approach for classifying websites. Procedure on how we prepare the two test beds are described as follow:

- Test bed 1 (Phishdentity): We adopt the selected setup from Experiment 2 for this test bed. In other words, we compare the results of the first test bed with the second test bed for this experiment.
- Test bed 2 (Phishdentity + additional approach): This test bed has the same setup as the first test bed except we integrate the additional approach (the theoretical explanation has been discussed in Section 3.5) into the test. In other words, we examine the URL of the query website with the additional approach when we have obtained the results of first test bed. The additional approach comprises of suspicious URL, dots in domain, age of domain, IP address and WOT. First, we inspect the URL of the query website for suspicious symbol. Then, we count the number of dots that appear in the URL domain. Next, we check the age of the query website using WHOIS service. After that, we examine whether the URL of the query website is formed by IP address. Finally, we use the WOT API to return the reputation and category of the query website. We combine all the scores produced by additional approach with the results of first test bed. In order to integrate Phishdentity with the additional approach, we use threshold 50 as a baseline to find the optimum weight for S_1 and S_2 . In other words, a website is classified as phishing if the $FinalScore$ obtained from Eq. (9) exceeds threshold 50 and vice versa. To achieve this, we distribute a weight of zero to S_1 and a weight of 100 to S_2. From here, we observed the number of correctly classified websites each time S_1 weight increased by one and S_2 weight is reduced by one. We stop the process when it produces the highest number of correctly classified websites. Table 4.5 (a) shows that Phishdentity has achieved the lowest error rate when we assign a weight of 40 to S_1 and a weight of 60 to S_2. Next, we need to find the optimal threshold in the classification of website using optimal weight obtained from Table 4.5 (a). This is because the previous threshold (i.e., threshold 50) is used as a baseline to find the optimal weight for $S_1$and S_2 in which they will be used in Eq. (9). To do so, we set the threshold value to zero initially. Then, we observed the number of correctly classified websites each

time the threshold value increased one. We halt this process when it produces the highest number of correctly classified website. Table 4.5 (b) shows the results of an optimal threshold value for Phishdentity and additional approach.

Material: We use a total of 1,000 websites (500 legitimate and 500 phishing) for each test bed.

Results: Table 4.5 shows the evaluation results for Phishdentity and additional approach. More precisely, Table 4.5 (a) shows the optimal weight for S_1 and S_2 when we set the threshold to 50. Table 4.5 (b) shows the optimal threshold value for Phishdentity and additional approach when we have obtained the optimum weight for S_1 and S_2. Table 4.5 (c) is the results of each test bed based on the favicon availability.

Table 4.5: Assessment for Phishdentity and additional approach for the availability of favicon.

(a) Optimal weight for S_1 and S_2 when τ is set to 50 as the baseline.

S_1	S_2	**TP (%)**	**TN (%)**	**FP (%)**	**FN (%)**	F_1
0	100	92.80	96.20	3.80	7.20	0.9440
20	80	95.20	96.80	3.20	4.80	0.9597
40	60	97.40	97.20	2.80	2.60	0.9730
60	40	97.40	96.60	3.40	2.60	0.9701
80	20	97.40	95.60	4.40	2.60	0.9653
100	0	97.40	94.60	5.40	2.60	0.9606

(b) Optimal threshold, τ for Phishdentity and additional approach.

Optimal threshold, τ	**TP (%)**	**TN (%)**	**FP (%)**	**FN (%)**	F_1
0	97.40	94.60	5.40	2.60	0.8749
20	97.00	97.80	2.20	3.00	0.9182
40	97.80	91.20	8.80	2.20	0.9468
50	97.40	97.20	2.80	2.60	0.9730
60	97.00	97.80	2.20	3.00	0.9739
80	55.80	99.80	0.20	44.20	0.7154
100	41.40	100	0.00	58.60	0.5856

(c) Performance assessment on Phishdentity and additional approach.

Test bed	**TP (%)**	**TN (%)**	**FP (%)**	**FN (%)**	F_1
Test bed 1	97.40	94.60	5.40	2.60	0.9606
Test bed 2	97.00	97.80	2.20	3.00	0.9739

Table 4.5 (a) has shown that Phishdentity has the fewest errors in classifying the website when we assign a weight of 40 to S_1 and a weight of 60 to S_2. In contrast, Table 4.5 (b) has shown that additional approach works best with Phishdentity when the τ is set to 60. Table 4.5 (c) shows the results of website classification from two different test beds based on the favicon availability. First test bed reveals that Phishdentity is able to classify legitimate and phishing websites. It has an acceptable amount of false positive with a relatively small number of false negative. The second test beds have shown an increase in classifying a legitimate website when we integrate additional approach to the test. But it reduces slightly the accuracy of detecting phishing websites. It reduces false positives by 3.2% from 5.4% to 2.2%, while false negatives increased by 0.4% from 2.6% to 3.0%.

Analysis: Second test bed has shown an improvement in classifying the legitimate websites when we integrate additional approach into the experiment. It reduces the false positive by 3.2% from 5.4% to 2.2%. However, this approach is subjected to higher false negative, where it increases the number of false negative from 2.6% to 3.0%. We argued that the increment of 0.4% in false negative is acceptable, given the number of false positives was reduced by 3.2%. This showed that a solution based on the URL can be used to repair the detection results. We observed that the additional approach works well with Phishdentity when we have allocated a weight of 40 to S_1 and a weight of 60 to S_2. Nonetheless, the threshold has not been optimized. This is because we only use it (i.e., threshold 50) as a baseline to find the optimal weight for S_1 and S_2. Nevertheless, we found that a value of 60 for τ can result in the fewest number of errors in the classification. According to the results obtained from WOT, we noticed that most of the legitimate websites without the presence of favicon is a legitimate website. These websites have a lifespan of more than one year. Furthermore, they do not put any symbol or overuse of dots in the domain. We also noticed that the WOT did not perform well for some of the legitimate websites. First, the WOT categorizes pornographic website listed in the Alexa ranking as a dangerous website. While this type of website has a lot of malicious advertising that will infect a computer with virus, but it is still classified as legitimate in the final calculation. The rationale is that we cannot deny every legitimate websites are harmless (i.e., pornographic websites), but our objective here is to determine the legitimacy and not the danger of the website. Second, WOT has poor rating on some of the e-commerce websites that are also listed in the Alexa ranking. We observed that the website does not use a secure connection when the user wants to log in, or make digital payments. This weakness is

unlikely to cause false alarms to legitimate websites. It is because the legitimacy is not solely determined by WOT. Third, some of the phishing websites are regarded as a good website by the additional approach, as shown in Table 4.5 (c) where false negative has increased by 0.4%. We observed that these phishing websites are hosted under legitimate website. Thus, the identity can be obtained from the search results. In addition, the website also has the attributes of a legitimate website. For example, the age of which exceeds the threshold that we have defined for age of domain.

Conclusion: The incorporation of additional approach has improved the performance of Phishdentity in classifying websites, especially for websites that do not have a presence of favicon. This shows that an approach based on the URL can be used to provide the necessary information required to analyze the properties of phishing. Therefore, we name this setup as Final Phishdentity and it will be used for benchmarking test in the Experiment 4.

4.7. Experiment 4 - Evaluation of Final Phishdentity

Research problem: We have conducted three different experiments to assess the performance of Phishdentity. However, these experiments do not contain the benchmarking test. It only covers the validation of the proposed features and the optimization of the Google search by image API. We believe that it is important to conduct an experiment that compares the performance of different techniques of anti-phishing. For this reason, we chose CANTINA [19] and GoldPhish [45] as the benchmark to assess the effectiveness of Final Phishdentity. The experiment will show the detection rate and detection speed of each technique. Then, we analyze the advantages and disadvantages of each technique.

Hypothesis: Final Phishdentity achieved higher efficiency in classifying websites compared with the techniques of other anti-phishing.

Procedure: We design three test beds in this experiment to benchmark the effectiveness of different techniques of anti-phishing. First test bed is designed for Final Phishdentity while second and third test beds are designed for CANTINA and GoldPhish, respectively. The procedure on how we prepare the test bed is described as follow:

- Test bed 1 (Final Phishdentity): We adopt the selected setup from Experiment 3 to be compared with the results of the second and third test beds.
- Test bed 2 (CANTINA): This test bed is designed to simulate the proposed mechanism as described in [19]. It uses a total of seven features to classify a website. First, we feed the Google search engine with the five most important keywords extracted from the query website. Then, we search for a domain name that corresponds with the query website domain name from the top 30 entries of search results. Second, we check the age of the query website to see whether it is aged more than a year. Third, we examine the query website whether it contains a logo that originated from other websites with different image path. Fourth, we inspect the URL of the query website for suspicious symbol. Fifth, we verify the query website domain name is not formed by an IP address. Sixth, the total number of dots that appear in the query website domain name should not exceed five. Finally, we look for webpages that ask for personal information. All the features will return a value of 1 if the query website has properties that match the description of the feature. It returns a value of -1 if it does not match. Next, we assign the weight to each feature before summing all the scores. If the final score is bigger than zero, then it is classified as legitimate. Otherwise, it is classified as phishing.
- Test bed 3 (GoldPhish): This test bed is designed to simulate the proposed mechanism as described in [45]. First, we install the latest version of Microsoft Office Document Imaging (MODI) from Microsoft website. Then, we set the resolution of the screen dump to 1200×400 pixels. Next, we use MODI to perform OCR to extract the textual information of the screen dump of each website. Typically, these messages are returned in the form of verse. From here, we feed the Google search engine with each verse to return only the top 5 entries of search results. Finally, we compare the query website domain name with the domain name of each search entry. If the query website domain name corresponds with any search entry, then the query website is deemed as legitimate. Otherwise, it is deemed as phishing.

Material: We use a total of 1,000 websites (500 legitimate and 500 phishing) for each test bed.

Results: Table 4.6 shows the results of performance comparison between each test bed. Table 4.6 (a) shows the detection rate of each test bed and Table 4.6 (b) shows the average duration needed to classify a website for each test bed.

Table 4.6: Performance comparison between Final Phishdentity, CANTINA and GoldPhish.

(a) Comparison of detection rate.

Test bed	TP (%)	TN (%)	FP (%)	FN (%)	F_1
Test bed 1	97.00	97.80	2.20	3.00	0.9739
Test bed 2	58.60	96.80	3.20	41.40	0.7244
Test best 3	98.40	61.20	38.80	1.60	0.8297

(b) Comparison of detection speed measured in seconds.

Test bed	Legitimate	Phishing
Test bed 1	10.98	11.88
Test bed 2	36.53	18.97
Test best 3	17.65	38.12

Table 4.6 shows the performance comparison in terms of detection rate and detection speed for each test bed. First test bed shows that Final Phishdentity can classify the website with minimal errors. It uses the shortest time compared with second and third test bed. Conversely, second test bed shows that the CANTINA has achieved a detection rate comparable to the Final Phishdentity in classifying legitimate websites. But CANTINA does not perform well in detecting phishing websites. It also takes more time to compute the results of the classification. Nevertheless, third test bed shows that GoldPhish has the highest detection rate in detecting phishing websites when compared to the first and second test beds. However, it performs badly in classifying legitimate websites. In addition, we find that GoldPhish has a slow detection speed similar to CANTINA.

Analysis: Final Phishdentity performs very well in classifying websites with low false positives and low false negatives. Therefore, we use this result to compare the differences with CANTINA and GoldPhish in order to find out the strengths and weaknesses.

- Test bed 2: This test bed shows that CANTINA can classify legitimate website with minimal errors. We noticed that the TF-IDF performs very well in extracting the five most important keywords that describe the query website. These keywords enable

Google search engine to return related information about the query website. We believe that the TF-IDF is effective in retrieving information if the content of a website is formed by textual content. Apart from that, CANTINA also introduced additional features to inspect the attributes of phishing based on the content of a website. These features improve the detection rate in classifying legitimate websites. We also noticed that CANTINA does not perform well for some legitimate websites. It falsely classified 16 legitimate websites as phishing websites. The TF-IDF technique can produce incorrect lexical signatures if the website uses images to convey information. This is due to very little textual information that describe about the website. Likewise, CANTINA can falsely classify a legitimate website if the Document Object Model (DOM) parser cannot parse sufficient information for TD-IDF to extract the correct keywords. This issue is also discussed in [20] where they planned to investigate for alternative approach. Nevertheless, we observed that the detection speed is affected by DOM performance. CANTINA can be very slow in computing the result if the websites have a large amount of information. On the other hand, CANTINA does not work well for phishing websites. It falsely classified 41.4% of phishing websites as legitimate websites. 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 despite the phishing websites have been detected by the TF-IDF. Our proposed technique does not suffer the weaknesses imposed on CANTINA. This is because the Final Phishdentity utilizes the favicon as main input with Google search by image API to uncover the hidden identity of the query website. In addition, we also propose an additional approach that is lightweight and is based on the URL to examine the legitimacy.

- Test bed 3: This test bed shows that GoldPhish performs badly in classifying legitimate websites. The number of falsely classified legitimate websites is as high as 38.8%. We argued that GoldPhish faced several limitations when using the OCR tool to extract the textual information from the screen dump. First, GoldPhish is using a fixed size window to crop the screen dump. This will potentially exclude some important content that is located outside of the cropped region. Second, we noticed that most of the websites from the Alexa top 500 global websites have advertisements on the home page. There are some advertisements which are so large that it covers half of the screen dump. 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 different messages about the website. Fourth, the OCR does not work well with some of the uncommon type and size of font. This causes the OCR unable to recognize the characters correctly. In short, the OCR will make more mistakes if the quality of screen dump is poor. Nevertheless, this test bed shows that GoldPhish performs very well in detecting phishing websites. The number of falsely classified phishing websites is only as high as 1.6%. The Google search engine is unlikely to return any information about the phishing website if phishers tried to confuse the OCR by using a modified content. In addition, it can be very time consuming for phishers to create new content for the website and the visual for the new website may vary from the original. GoldPhish requires longer period of time to compute the results compared to the Final Phishdentity. This is because the OCR feed the Google search engine with each line of verse. Hence, the duration in between the search is increased. On the other hand, GoldPhish spent longer duration to compute the results for phishing websites. Based on our observations, OCR extracts a lot of information from the screen dump of phishing.

Conclusion: Final Phishdentity proved to be very good in classifying websites. It has better detection rate and detection speed compared with CANTINA and GoldPhish. We believe that our proposed technique can be further enhanced by adding additional features in the future. In addition, we also want to increase the size of the data to prove the effectiveness of Final Phishdentity.

4.8. Limitations

We discovered that our proposed technique has four limitations along the experiments. The first limitation is that the phishers could alter slightly the content of a favicon to avoid Phishdentity detection. They could also replace the favicon with other website favicons. However, this limitation is not vital. This is because Google search by image engine would extract different contents from the altered favicon and return information not related to the targeted legitimate website. Moreover, the domain name of a phishing website with altered favicon would not appear in the search results. Thus, phishing websites with altered favicon can still trigger Phishdentity detection.

The second limitation is that WOT does not contain information for each legitimate website. This limitation is likely to happen on new legitimate website and unpopular website. However, while time progresses, eventually the new website will be listed in the WOT database. As for the unpopular website, most probably it will not be targeted by phishers. We believe that the missing data will be available in the WOT database soon. This is because the WOT has very large community actively updating the information about the old and newly discovered websites. Hence, the WOT system can be used to examine the legitimacy of a website.

The third limitation is the overhead time occurred during information retrieval from Google search. More precisely, the retrieval needs time to analyze, access, content extraction of a favicon and time to return search results. While the overhead may affect the effectiveness of Phishdentity in classifying websites, but we believe that this limitation will not interfere with the user experience in web browsing. We argued that Phishdentity can inform internet users before they submit confidential information if the website is deemed as phishing.

The fourth limitation is that the Google search by image engine is not in favor of the new legitimate websites. This can happen if Google has yet to index the new legitimate websites. As a result, Google search by image engine may not be able to return adequate information necessary for Phishdentity to identify the identity of new website. Nevertheless, this limitation is not likely to affect the performance of Phishdentity in the classification. This is because we have proposed an additional approach which is based on the website URL for classification. Additional approach is lightweight and can be used to offset Phishdentity inability to classify websites that do not have the presence of favicon.

Chapter 5

Conclusions and Future Work

In this final chapter, we conclude the work from the first chapter to the fourth chapter. Then, we present the achievements of this thesis based on the objectives of our research to address the problem statements. Finally, we outline the direction for future research.

5.1. Conclusions

In chapter one, we have described the motivation of this thesis to develop a new phishing detection mechanism. This is because phishing has caused a serious defect to economic growth especially to the reputation of online business. As a result, it can make internet users to lose confidence in online business. More importantly, phishing activity trends did not show a significant decline, though many anti-phishing organizations have conducted various activities to protect internet users. Moreover, we also have included a case study that simulates a scenario where internet users do not have any protection when surfing the web. Nevertheless, most of the existing anti-phishing solutions cannot reveal the identity of a website. These solutions only notify the matching attributes of phishing. In addition, these solutions do not have a high detection speed for websites classification. Eventually, all these drawbacks will cause internet users to ignore the presented warning. Therefore, the objectives of this research are proposed to address the limitations imposed in existing anti-phishing solutions.

In chapter two, we have discussed the methods used by phishers to distribute phishing websites. More specifically, the phishers take advantage of the loophole in technology to create phishing websites. They incorporate various phishing techniques during the creation of phishing websites. We also have discussed a wide range of activities undertaken by government and non-profit organizations in an effort to combat phishing. But, they still cannot achieve the ideal results. Based on the review, there are still many internet users who

become the victims of phishing each year. Nevertheless, we have conducted several studies to understand the mechanism of phishing detection methods. These studies will guide us to identify the pros and cons of each phishing detection method. The list-based approach is the most fundamental approach in the field of anti-phishing. It has been widely used by developers of web browsers to protect internet users from visiting phishing websites. This approach is effective against phishing websites provided there is a comprehensive list in the database. On the other hand, image-based approach was introduced to detect image-based phishing websites. This is because the phishers like to use images to construct the contents of phishing websites. There are several factors that can affect the accuracy of this approach. For example, the level of similarity between phishing website and legitimate website is one of the factors that may affect the accuracy of image-based approach. Search engine based approach is also another phishing detection approaches that can be used to detect phishing websites. This approach decides the legitimacy of a website based on the information obtained from search engines. However, this approach is not conducive to new website and unpopular website. Next, we also studied several commonly used classifiers in anti-phishing solutions. They include KNN, SVM, LR, LDA, and NB classifiers. Nonetheless, KNN outperformed other classification when K is set to 3.

Based on the analysis conducted on most related existing anti-phishing techniques, we have proposed a novel technique named Phishdentity in Chapter three. We begin by introducing the components, namely the website favicon and Google search by image engine. We proposed the use of the favicon because it represents the brand of the website. Favicon extraction is very fast and easy. It can be located by appending a string, *favicon.ico* into the domain of the website. Unlike other image extraction techniques in which the system needs to search the contents of the website to locate the image. Meanwhile, we suggested the use of Google because it is the only search engine that allows images to be used as search queries to find information on the web. Furthermore, Google has also indexed a large number of legitimate websites. In other words, Google can return more accurate results related to the search query. Therefore, it is reasonable to use the search results returned by Google to search for the identity of the website. More precisely, we consider the SLD from the search results that has the highest frequency of occurrence to be the identity of the website. From here, we can determine the legitimacy of a website based on a match between the SLD determined from the search results with the SLD of the query website. However, we observed

that the initial version of the prototype does not perform well if the favicon is missing from the website. Therefore, we have introduced additional approach to overcome the weakness imposed on this technique.

In chapter four, we have conducted several experiments using data obtained from Alexa top 500 global websites and PhishTank to examine the performance of Phishdentity. In the first experiment, we observed that Phishdentity can achieve good detection rate although it has suffered a fair number of false positives. The experiment has shown that Phishdentity may misclassify a legitimate website as a phishing website if the favicon is missing from the website. However, this issue does not affect Phishdentity in classifying phishing websites. In the second experiment, we observed that the performance of Phishdentity increases when we increase the number of entries returned by Google. In other words, the probability of the identity of the website appears in search results are through the growth in the number of entries. We have decided to use the top 30 entries for Phishdentity. This is because it is the optimum number of entries without having to sacrifice more processing time. In the third experiment, we observed that Phishdentity has improved in the classification of legitimate websites when we integrate additional approach into the test. Conversely, it reduces slightly the accuracy of Phishdentity in classifying phishing websites. This experiment has shown that the additional approach can be used to offset the weakness in Phishdentity in classifying legitimate websites without the presence of favicon. Therefore, we name the final version of Phishdentity as Final Phishdentity. In the fourth experiment, we observed that the Final Phishdentity has surpassed the performance of CANTINA and GoldPhish in terms of the detection rate and detection speed. This experiment has shown that the Final Phishdentity is excellent in classifying websites with little errors.

5.2. Research Contribution

The contributions of this work are fourfold:

- We explore a new niche in detecting phishing websites. We exploit favicon extracted from the website with the help of Google search by image engine to discover potential phishing attempt.

- We have used some simple mathematical equations to assist in retrieving the correct identity from search results returned by the Google search by image engine.
- Unlike current anti-phishing techniques, our proposed technique eliminates the need to perform intensive analysis on either text-based or image-based content that can result in reduced detection speed.
- In addition, our proposed technique does not require a database of images or any other pre-saved information from legitimate websites. Thus, it reduces the risk of having high false positive or negative due to outdated database.

5.3. Future Work

In this thesis, we have proposed an anti-phishing solution named Phishdentity. We plan to expand the scale of our corpus so that it covers wider type of phishing attacks. To achieve that, we will collect phishing data from different sources. For example, we will re-use archive phishing data acquired from the major search engines (i.e., Google, Yahoo, and Bing) and cyber security companies to improve the precision of classification.

In addition, we also plan to address the limitations imposed on the current implementation. First, we plan to develop a database to store a list of favicon. This list will filter out legitimate websites that match the information stored in the database. To do so, we will extract the favicon of query website to find matching from the list. If the favicon corresponds to the list but the URL domain is different, then we will give a positive value (i.e., value of 1) for this feature. Otherwise, we set the value to zero. This feature can prevent phishers from using the altered favicon to obfuscate Google search by image engine and Phishdentity. More importantly, the use of a database can avoid the need to reclassify the same website again. This will further improve Phishdentity speed of detection.

Second, we plan to use the Google PageRank value as one of the proposed features. The PageRank will show the ranking of the given website. From here, we can determine the legitimacy of a website based on this value. To achieve that, we will integrate Google Toolbar Rank (GTR) [63] to Phishdentity. GTR is a browser toolbar developed by Google. It has many features including the PageRank. This feature is also used by Sunil et al. [52] to

detect phishing websites. PageRank is useful to our proposed technique especially when the website is not available in the WOT database.

BIBLIOGRAPHY

[1] APWG. Phishing activity trends report. 2nd Quarter 2014. *Unifying the Global Response to Cybercrime*. Retrieved from http://docs.apwg.org/reports/apwg_trends_report_q2_2014.pdf (Accessed 13 October 2014).

[2] RSA, the security division of EMC. http://www.rsa.com/solutions/consumer_authentication/intelreport/11752_Online_Fraud_report_0712.pdf (Accessed 16 Dec 2013).

[3] Anti-Phishing Alliance of China (APAC). *Briefing o handling of phishing websites in August 2012*. http://en.apac.cn/regulation_policy/201210/P020121015431015160285.pdf (Accessed 14 Dec 2013).

[4] Cyber999. *MyCERT 4th Quarter 2013 Summary Report*. http://www.mycert.org.my/en/services/advisories/mycert/2014/main/detail/955/index.html (Accessed 13 June 2014).

[5] M. Cova, C. Kruegel, and G. Vigna. There is no free phish: An analysis of "Free" and live phishing kits. *Proceedings of the 2nd USENIX Workshop on Offensive Technologies*, 2008.

[6] G. Ollmann (2007). *The Phishing Guide (Part 1).* Retrieved from http://www.technicalinfo.net/papers/Phishing.html (accessed 22 Dec 2013).

[7] J. Shi and S. Saleem (2012). *Phishing*. Retrieved from http://www.cs.arizona.edu/~collberg/Teaching/466-566/2012/Resources/presentations/2012/topic5-final/report.pdf (accessed 21 Dec 2013).

[8] R. Miller (2007. *Phishing attacks continue to grow in sophistication. Netcraft.* Retrieved from http://news.netcraft.com/archives/2007/01/15/phishing_attacks_continue_to_grow_in_sophistication.html (accessed 25 Dec 2013).

[9] Trinity College. *Avoid Phishing Scams*. Retrieved from http://www.trinity.utoronto.ca/about/human-resources/avoid-phishing-scams.html (accessed 26 Dec 2013).

[10] Online Security. *Phishing Techniques*. http://www.streetdirectory.com/etoday/phishing-techniques-wfaofl.html (accessed 28 Dec 2013).

[11] B. Friedman, D. Hurley, D. C. Howe, H. Nissenbaum, and E. Felten. Users' conceptions of risks and harms on the web: a comparative study. *Proceedings of the Extended Abstracts on Human Factors in Computing Systems*, pages 614 – 615, 2002.

[12] J. Rathod and D. Nandy. URL obfuscation phishing and anti-phishing: a review. *Proceedings of the Journal of Engineering Research and Applications*, pages 338 – 342, 2014.

[13] M. McDowell (2009, Oct 22). *Avoiding Social Engineering and Phishing Attacks.* Retrieved from https://www.us-cert.gov/ncas/tips/ST04-014 (accessed 24 January 2014).

[14] US-CERT. *APWG Fax Back Phishing Education Program*. https://www.us-cert.gov/ncas/current-activity/2010/08/25/APWG-Fax-Back-Phishing-Education-Program (accessed 25 January 2014).

[15] PhishTank. *Join the fight against phishing*. http://www.phishtank.com/ (accessed 20 Dec 2013).

[16] A. Abbasi, F. Zahedi, and Y. Chen. Impact of anti-phishing tool performance on attack success rates. *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, pages 12-17, 2012.

[17] J. H. Huh and H. Kim. Phishing detection with popular search engine: simple and effective. *Proceedings of the 4th Canada-France MITACS conference on Foundations and Practice of Security*, pages 194 – 207, 2011.

[18] R. B. Basnet and A. Sung. Mining web to detect phishing URLs. *Proceedings of the 11th international conference on Machine Learning and Applications*, pages 568 – 573, 2012.

[19] Y. Zhang, J. I. Hong, and L. F. Cranor. CANTINA: A content-based approach to detecting phishing web sites. *Proceedings of the 16th International Conference on World Wide Web*, pages 639-648, 2007.

[20] S. G. Prevost, G. G. Granadillo, and M. Laurent. Decisive heuristics to differentiate legitimate from phishing sites. *Proceedings of the Conference on Network and Information System Security*, pages 1 – 9, 2011.

[21] M. Usha, and P. Deepika. Phishing – A challenge in the Internet. *Proceedings of the International Journal of Computer Science and Information Technologies*, pages 260 – 263, 2014.

[22] Y. Cao, W. Han, and Y. Le. Anti-phishing based on automated individual white-list. *Proceedings of the 4th ACM workshop on Digital identity management*, pages 51 – 60, 2008.

[23] X. Dong, J. A. Clark, and J. L. Jacob. User behavior based phishing websites detection. *Proceedings of the international multiconference on Computer Science and Information Technology*, pages 783 – 790, 2008.

[24] G. Xiang and J. I. Hong. A hybrid phish detection approach by identity discovery and keywords retrieval. *Proceedings of the 18th international conference on World Wide Web*, pages 571 – 580, 2009.

[25] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 581 – 590, 2006.

[26] A. Karakasiliotis, S. Furnell, and M. Papadaki. An assessment of end-user vulnerability of phishing attacks. *Proceedings of the Journal of Information Warfare*, pages 17 – 28, 2007.

[27] A. Herzberg and A. Jbara. Security and identification indicators for browsers against spoofing and phishing attacks. *Proceedings of the Journal of Transactions on Internet Technology*, 2008.

[28] U. S. Odaro and B. G. Sanders. Social engineering: phishing for a solution. *Proceeding of IT Security for the Next Generation*, 2011.

[29] S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge. Anti-Phishing Phil: the design and evaluation of a game that teaches people not to fall for phish. *Proceedings of the 3rd symposium on usable privacy and security*, pages 88 – 99, 2007.

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

[31] Accuvant Labs. Browser Security Comparison: A quantitative approach. http://files.accuvant.com/web/files/AccuvantBrowserSecCompar_FINAL.pdf (accessed 21 February 2014).

[32] Chrome Browser. *Explore the Chrome Browser*. https://www.google.com/intl/en/chrome/browser/features.html#security (accessed 5 February 2014).

[33] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel. On the effectiveness of techniques to detect phishing sites. *Proceedings of the 4th International Conference on Detection of Intrusions and Malware Vulnerability Assessment*, pages 20 – 39, 2007.

[34] N. Singh and N. R. Roy. A hybrid approach to detect zero day phishing websites. *Proceedings of the International Journal of Information & Computer Technology*, pages 1761 – 1770, 2014.

[35] B. Wardman, T. Stallings, G. Warner, and A. Skjellum. High performance content-based phishing attack detection. *Proceedings of eCrime Researchers Summit*, pages 1 – 9, 2011.

[36] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying suspicious URLs: an application of large-scale online learning. *Proceedings of the 26th International Conference on Machine Learning*, pages 681 – 688, 2009.

[37] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta. PhishNet: predictive blacklist to detect phishing attacks. *Proceedings of INFOCOM on Computer Communication*, pages 1 – 5, 2010.

[38] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1245 – 1254, 2009.

[39] C. Whittaker, B. Ryner, and M. Nazif. Large-scale automatic classification of phishing pages. *NDSS, The Internet Society*, 2010.

[40] A. Herzberg and A. Jbara. Security and identification indicators for browsers against spoofing and phishing attacks. *ACM Transactions on Internet Technology, 8 (4)*, 2008.

[41] Binational Working Group. (2006). *Report on Phishing*. Retrieved from https://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/archive-rprt-phshng/archive-rprt-phshng-eng.pdf (accessed on 15 December 2013).

[42] I. Lam, W. Xiao, S. Wang, and K. Chen. Counteracting phishing page polymorphism: An image layout analysis approach. *Proceedings of the 3rd Information Conference and Workshops on Advances in Information Security and Assurance*, pages 270 – 279, 2009.

[43] E. Medvet, E. Kirda, and C. Kruegel. Visual-similarity-based phishing detection. *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, 2008.

[44] M. Hara, A. Yamada, and Y. Miyake. Visual similarity-based phishing detection without victim site information. *Proceedings of IEEE symposium on Computer Intelligence in Cyber Security*, pages 30 – 36, 2009.

[45] M. Dunlop, S. Groat, and D. Shelly. GoldPhish: using images for content-based phishing analysis. *Proceedings of the 5th International Conference on Internet Monitoring and Protection*, pages 123-128, 2010.

[46] Ricardo Niederberger Cabral, "ImgSeek," http://www.imgseek.net/.

[47] C. Jacobs, A. Finkelstein, and D. Salesin. Fast multiresolution image querying. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 277 – 286, 1995.

[48] S. Afroz and R. Greenstadt. PhishZoo: detecting phishing websites by looking at them. *Proceedings of 5th IEEE international conference on Semantic Computing*, pages 368 – 375, 2011.

[49] D. G. Lowe. Object recognition from local scale-invariant features in ICCV ’99. *Proceedings of the 7th IEEE international conference on Computer Vision*, pages 1150 – 1157, 1999.

[50] List of Search Engines. *Search Engines – Top 10 Search Engines List 2014.* http://www.listofsearchengines.org/ (accessed 1 Dec 2014).

[51] B. Dean. Google’s 200 ranking factors: The complete list. http://backlinko.com/google-ranking-factors (accessed 6 December 2014).

[52] A. N. V. Sunil and A. Sardana. A PageRank based detection technique for phishing web sites. *Proceedings of the IEEE Symposium on Computers & Informatics*, pages 58-63, 2012.

[53] Open SEO Stats (PageRank Status). Retrieved from http://pagerank.chromefans.org/ (accessed 17 September 2014).

[54] S. Markus. A new AV-TEST study: search engines as malware providers. Retrieved from http://www.av-test.org/fileadmin/pdf/avtest_2013-03_search_engines_malware_english.pdf (accessed 18 May 2015).

[55] StopBadware – IP Address Report – Top 50 by Number of Reported URLs, http://stopbadwwawre.org/reports/ip.

[56] hpHosts Online – Simple, Searchable, & FREE!, http://hostsfile.net/

[57] E. H. Chang, K. L. Chiew, S. N. Sze, and W. K. Tiong. Phishing detection via identification of website identity. *Proceedings of the International Conference on IT Convergence and Security*, pages 1-4, 2013.

[58] A. Schaback. Google reverse image search scraping without API in PHP. Retrieved from http://skyzerblogger.blogspot.tw/2013/01/google-reverse-image-search-scraping.html (accessed 23 May 2013).

[59] APWG. Phishing attacks trends report. http://docs.apwg.org/reports/APWG_Phishing_Attack_Report-Jul2004.pdf (Accessed 23 March 2014).

[60] Web of Trust. *API*. https://www.mywot.com/wiki/API (accessed 31 December 2013).

[61] SEOCHAT. Domain age checker. Retrieved from http://tools.seochat.com/tools/domain-age/ (accessed 25 November 2013).

[62] Alexa. *The top 500 sites on the web*. http://www.alexa.com/topsites (accessed 20 Dec 2013).

[63] Google, “Google Toolbar Rank”, http://toolbar.google.com/.