**PHISING PREVENTION USING SOURCE CODE ANALYSIS**

**CHAPTER ONE**

**INTRODUCTION**

**1.1 BACKGROUND INFORMATION TO THE STUDY**

Phishing is a crime in which a perpetrator sends the fake e-mail, which appears to come from popular and trusted brand or organization, asking to input personal credential like bank password, username, phone number, address, credit card details, and so forth. The fake e-mails often look amazingly legitimate, and even the website where the Internet user is asked to input personal information also looks similar to legitimate one. Phishing messages propagate over e-mail, SMS, instant messengers, social networking sites, VoIP, and so forth, but e-mail is the popular way to perform this attack and 65%of the total phishing attack is achieved by visiting the hyperlink attached to the e-mail. Moreover, spear phishing attack is becoming popular nowadays. Business e-mail compromise (BEC) is observed as a major Internet threat in 2015. In BEC, the intruder uses spear phishing methods to fool organizations and Internet persons. More sophisticated spear phishing attacks targeted particular individual or groups within the organization. Phishing is metaphorically similar to fishing in the water, but instead of trying to catch a fish, attackers try to steal consumer’s personal information. When a user opens a fake webpage and enters the username and protected password, the credentials of the user are acquired by the attacker which can be used for malicious purposes. Phishing websites look very similar in appearance to their corresponding legitimate websites to attract large number of Internet users. Recent developments in phishing detection have led to the growth of numerous new visual similarity based approaches. Visual similarity based approaches compare the visual appearance of the suspicious website to its corresponding legitimate website by using various parameters. Due to different phases of phishing detection, this paper contains the following:

1. Background, History, and Statistics section presents the history of phishing attacks, worldwide financial losses due to phishing attacks, the lifecycle of phishing attack, and classification of various types of phishing attacks. This section describes the overall picture of phishing attacks from a high level perspective.
2. Next, we describe how attacker fools an Internet user and how they bypass the anti-phishing system.
3. Similarly, we present various types of phishing detection techniques, their advantages, and drawbacks.

Also, we provide a comprehensive literature review of visual similarity based phishing detection approaches, which incorporates document object model (DOM), Cascading Style Sheet (CSS), HTML tag, image processing, and hybrid techniques. Moreover, we present a comparison between various visual similarity based anti-phishing techniques. It provides a better understanding of the problem, current solution space, and future research scope to efficiently deal with phishing attacks using visual similarity based approach.

In addition, we provide several issues and challenges in detection of phishing attacks.

The rest of the paper is structured as follows. Section 2 contains the background, history, and statistics of phishing attack. Section 3 describes the overview of phishing detection using visual similarity based approaches. Section 4 presents the taxonomy of various types of phishing detection and filtering techniques; especially this section focuses on visual similarity approaches in detail. Section 5 presents the performance and evaluation matrices to judge the anti-phishing system. Section 6 presents the open issues and challenges in phishing detection and protection. Finally, Section 7 concludes the paper.

**1.2 STATEMENT OF PROBLEMS**

The problem taken up for study in this research is the need for a detection and prevention of phishing, for the client side techniques and for the server side techniques. The client side phishing consists of three stages of filtering techniques, such as URL verification; parse tree validation and behavioral model approach. The server side phishing consists of one-time password mechanism, watermarking, and preventing phishing through session hijacking and detecting e-mail phishing. The aim of the study is to first to collect all the URLs from the web, and check their formats. Second, the contents, source code, and all the forms are validated and finally the e-mail contents are checked based on the legitimate features.

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 dis-similar (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 being. 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.

**1.3 SIGNIFICANCE / JUSTIFICATION OF THE STUDY**

The majority of phishing attacks appear to originate in china. Last year, a significant rise in phishing attacks on shared virtual scams. In these attacks, a phisher breaks into a web server that hosts large number of domains then places then places the phishing content on every domain which infected thousands of web Sites simultaneously. While the financial industry continues to be primary target for phishers, it’s certainly not the only sector vulnerable to attack. Auction sites, payment services, retail and social networking sites are also frequent targets, as are cell phones providers and manufacturers. No business or brand is inherently safe. It if poses on company’s official website it diminish its online brand and deter customers from using actual website out of fear of becoming fraud victims.

Criminals began to target specific companies and their customers in order to improve their returns. These attacks used advanced technical capabilities such as exploitation of weak website coding or security flaws within desktop operating system. The phishes combined multiple technical elements along with social engineering to target an attack on specific brands. Any organization that does web based e-commerce is a potential target. This time, as expected, targets included social networks, search engines and email services, telecom companies, e-payment services, banks, and other credit and financial institutions. However, there were a few surprises as well, such as tax and customs agencies, the governments of various countries, car companies, insurance companies, medical institutions, oil companies, and transportation companies (including some airlines).

**1.4 AIMS & OBJECTIVE OF THE STUDY**

The objective of this project is to examine the source code of the application and identify vulnerabilities before the application is deployed.

* **AIMS**

This research aims towards a comprehensive literature review for the analysis of various prevention mechanisms. We will review the trends in these capabilities over the past two years and discuss currently deployed countermeasures. This project focus to identify and consolidate the prevention mechanism and to formulate a methodology to evaluate these models.

* **OBJECTIVES**

The objective of our research is to study and analyze various issues in phishing attack. To provide benefit to the user so that a systematic and more reliable approach can be used to maintain the prevent users from email scams, and it is also:

1. To review relevant literature
2. To design a new approach that can analyze web source code
3. To test and evaluate the design.

**1.5 HISTORICAL BACKGROUND OF THE STUDY**

According to APWG, the term phishing was coined in 1996 due to social engineering attacks against America On-line (AOL) accounts by online scammers.

The term phishing comes from fishing in a sense that fishers (i.e. attackers) use a bait (i.e. socially-engineered messages) to fish (e.g. steal personal information of victims). However, it should be noted that the theft of personal information is mentioned here as an example, and that attackers are not restricted by that as previously defined in Section II. The origins of the ph replacement of the character f in fishing is due to the fact that one of the earliest forms of hacking was against telephone networks, which was named Phone Phreaking. As a result, ph became a common hacking character replacement of According to APWG, stolen accounts via phishing attacks were also used as a currency between hackers by 1997 to trade hacking software in exchange of the stolen accounts. Phishing attacks were historically started by stealing AOL accounts, and over the years moved into attacking more profitable targets, such as on-line banking and e-commerce services. Currently, phishing attacks do not only target system endusers, but also technical employees at service providers, and may deploy sophisticated techniques such as MITB attacks.

According to Weider D. et. al., the primary motives behind phishing attacks, from an attacker’s perspective, are:

* **Financial Gain**: Phishers can use stolen banking credentials to their financial benefits.
* **Identity Hiding**: Instead of using stolen identities directly, phishers might sell the identities to others whom might be criminals seeking ways to hide their identities and activities (e.g. purchase of goods).
* **Fame and Notoriety**: Phishers might attack victims for the sake of peer recognition.

According to APWG, phishing attacks were in a raise till August, 2009 when the all-time high of 40,621 unique phishing reports were submitted to APWG. The total number of submitted unique phishing websites that were associated with the 40,621 submitted reports in August, 2009 was 56,362.

As justified by APWG, the drop in phishing campaign reports in the years 2010 and 2011 compared to that of the year 2009 was due to the disappearance of the Avalanche gang4 which, according to APWG’s 2nd half of 2010 report, was responsible for 66.6% of world-wide phishing attacks in the 2nd half of 2009. In the 1st half of the year 2011, the total number of submitted phishing reports to APWG

**CHAPTER TWO**

**2.1 LITERATURE REVIEW**

Phishing is one of the luring techniques used by phishing artists with the intention of exploiting the personal details of unsuspected users.

Phishing website is a mock website that looks similar in appearance but different in destination. Criminals are duping computer users into revealing sensitive information including bank accounts, social security numbers, credit card numbers, passwords, and more (Tommie 2005). Phishing attacks cost millions of dollars in fatalities to business organizations and end users (Geer 2005). The unsuspected users post their data thinking that these websites come from trusted financial institutions. Only specialists can identify these types of phishing websites immediately. But all the web users are not specialists in computer engineering and hence they become victims by providing their personal details to the phishing artists (Santhana and Vijaya 2011).

Phishing is continuously growing since it is easy to copy an entire website using the HTML source code. By making slight changes in the source code, it is possible to direct the victim to the phishing website. Phishers use lot of techniques to lure the unsuspected web user. They send generic greetings to the customers to check their account immediately. They also send threat messages of account cancellation and directing the users to update their account immediately to avoid cancellation. Data mining techniques can improve the assessment of phishing attacks (Xi Chen et al 2011).

Phishing is a serious threat to both users and enterprises; numerous anti-phishing techniques have been developed. In general, the techniques have been classified as either List-based or Heuristics-based technologies. List based techniques maintain a blacklist or white list or both. A lot of anti-phishing methods use a blacklist to prevent users from accessing phishing sites. These techniques follow a quantitative approach for evaluating the phishing possibility of a given website using the refined security risk elements for domain and web page. Design and implementation of the website risk assessment system for anti-phishing are also included (Young-Gab 2012).

Heuristic-based mechanisms employ several criteria to determine whether a website is a phishing site or not (Chun-Ying et al 2011). The CAPTCHA authentication application designed in an economical mode protects the security unconscious user by enabling safe online banking authentication, thereby addressing the online banking threats. Solving the general ignorance of security warning as well as ensuring safe online banking authentication even on a compromised hosts are the prime challenges of a secure online banking system. The proposed hardware solutions are not feasible for the home users due to its exorbitant cost (Leung 2013). Several commercial digital forensics software suites are available for examining digital media related to computer crimes. Although these tools provide examiners with extensive capabilities for forensic examinations, they can have significant drawbacks in terms of training, initial costs of the tool, and yearly maintenance upgrades. Alternatively, there are Free and Open Source Software (FOSS) tools with equivalent functionality that examiners can use to perform most of the same

tasks possible by commercial applications (Philili et al 2007).

**2.2 DIRECTIONS OF THE WORK**

Detecting and preventing phishing websites are always an important area of research. Various phishing techniques provide abundant and essential ways for effectively detecting and, protecting the confidential information of the individuals and organizations. URL plays an important role in phishing. URL is a major area, through which the websites have been initiated and through the hyperlink the pages are redirected to the next page.

Redirecting the pages is the vulnerable concept in phishing (i.e.) through the hyperlink; the pages are redirected to the legitimate site or the phishing site.

Numerous phishing sites are getting added each day. This motivated many researchers to switch their focus on finding the phishing sites. Many of the phishing techniques that were applied on websites, were used by the research community (either with necessary modification or with new proposals) to protect the individual and organization from their great loss.

Our research work also aims to explore to avoid phishing in E-mail.

Based on the research articles, it is also understood that secret information are collected through e-mail and tempt the users through attractive advertisements. This has motivated to do a literature study involving the phishing detection and prevention in client side techniques and server side techniques. The rest of the sections discuss the major works carried out relating to URL verification, parse tree validation, behavioral response, onetime password mechanism, watermarking mechanism, preventing phishing through session hijacking and e-mail phishing.

**2.2.1 Blacklist and White list**

Blacklist and White list are probably the most straightforward solution for anti-phishing. A white list contains URL of known legitimate sites while a blacklist contains those of known phishing sites. Many current anti-phishing technologies rely on the combination of white list and blacklist. The representative blacklist or white list based systems include Phish Tank, Site Checker, Google Safe Browsing ([www.google.com/safebowsing](http://www.google.com/safebowsing)), FirePhish and CallingID Link Advisor (www.callingid.com). These anti-phishing solutions are usually deployed as toolbars or extensions of web browsers to remind the users whether they are browsing a safe website.

Blacklist suffers from a window of vulnerability between the time a phishing site is launched and the site’s addition to the blacklist. A blacklist of phishing sites also requires frequent updating but still cannot include new phishing sites timely. Similarly, a white list also needs to update its

content in a large scale. Unfortunately, it cannot include all legitimate sites (Liu et al 2010).

IP addresses or domain names are used to block the websites but not possible to block the whole domain. Blacklists get updated in different speeds, and are varied in coverage, as 47% to 83% of phish appears on blacklists 12 hours from the initial test (Lorrie et al 2006). They check whether a webpage is a phishing page or legitimate one, based on its content, HTTP transaction and search engine results. The URL is checked with the black list whether it is available or not (Mingxing et al 2011) and classifieds based on the decision.

Automatic phishing classifiers are built and are currently used to evaluate phishing websites and maintain the blacklist. This blacklist contains Page’s URL and hosting information. To evaluate each page, the classifier considers features regarding the page’s URL, content and hosting information (Colin 2010). The request and response pairs to the blacklist server and often includes lookup strings and responses for domains such as microsoft.com.

The “Blacklist” feature place includes of binary variables for relationship in six blacklists (and one white list) run by Spamhaus. These lists combined provide at best a 20% error rate. When combined the blacklists with the Spam Assassin Botnet plugin features in the “Blacklist+Botnet”

feature set, the error rate improves to 12%. These small features sets are reasonably effective, but including features that result in very large feature sets significantly improve classification accuracy (Justin et al 2009). The main trouble with crawling and blacklist offers is that the anti-phishing organizations are caught up themselves in a race against the attackers.

Unfortunately, there is forever a window of susceptibility during which users are susceptible to attacks. Furthermore, the methods are only as effective as

the excellence of the lists that are maintained (Angelo et al 2007). The Net Craft anti-phishing toolbar (www.toolbar.netcraft.com) prevents phishing attacks by utilizing a centralized blacklist of current phishing URLs. Other examples include web sense, McAfee’s anti-phishing filter, Net Craft anti-phishing system, Cloudmark SafetyBar, and Microsoft Phishing Filter (Pan and Ding 2006). To combine the blacklist and heuristics approach developed a hierarchical blacklist-enhanced phish detection framework. The key insight behind this detection algorithm is to detect phish in a probabilistic fashion with very high accuracy (Guang et al 2010).

Although the process of checking in blacklist and white list is efficient method but only checking in the black list is not an efficient method, because all the phishing URLs are not available in the blacklist. If there is any problem in the blacklist (like connection error) the system cannot give the correct result.

**2.2.2 Visual Similarity**

The look and feel of a website gives the conviction to the victims that they are visiting a legitimate website. The three metrics used to measure the visual similarity are layout similarity, block level similarity and overall style similarity. Webpage segmentation forms the base to define these metrics. Salient blocks forms the structure of a webpage and the weighted average of the similarities between the paired blocks is known as block level similarity whereas the ratio between the total no of blocks and weighted number of matched blocks is known as layout similarity. The histogram of the style feature helps in calculating the overall style similarity i.e. the normalized correlation coefficient of the histograms of two webpages (Liu et al 2005).

The potential phishing pages are compared against the actual pages to assess the visual similarities between them in the metrics of key region, overall style and page layouts (Liu et al 2006).

EMD involves the Human Computer Interaction, text and graphics level into it (Revathy and Guhan 2012). The three steps involved in the preprocessing of web pages are;

* Performing normalization,
* Getting the image of the webpage from the given URL and
* Representing the image of the webpage as the visual signature of the webpage (including the color and coordinate features)

Txt section with its attributes and the visual image contributes to the overall look and feel of the webpage. Comparing the background color, font size width and height of the image and position in the page identifies similarities (Radha et al 2011).

**2.2.3 URL Verification**

URL obfuscation has become a key trick among all the tricks used in phishing activities. Therefore, equipping the user by creating awareness about the obfuscated URLs and how to determine the true nature of the strange URLs is the need of the hour (Ed Skoudis 2012). URL analysis involves all the formats of the URL being analyzed but mostly the links related to login information only (Debra et al 2009). While the difference and similarities between the URLs are identified with the help of the trusted site, generally the Link Guard algorithm is applied for analyzing the common characteristics and hyperlinks (Juan and Guo 2006). Content matching

techniques and DNS queries are also used to identify the malicious URLs apart from using the regular expressions and hash maps for isolating the Symantec variations (Pawan et al 2010). Phish wish is not using any training any training and white list or blacklist (Debra et al 2008). Page Safe is a tool that is based on user input to find the legitimacy of the URL (Sengar and Vijay 2010). Registering a similar domain to trick the user into a fraudulent site is becoming common by using the @symbol for redirection. For e.g., in the case of www.paypal.com@123.123.123.123 the user may still feel that they are visiting the site www.paypal.com, but actually being directed to a site with 123.123.123.123 as the IP address. Therefore checking the URLs for any special characters gains the importance now (Chun-Ying et al 2011).

The following points were considered in the Phish Market Model (Tyler and Tal 2010).

* Only specific URLs requested by the receiving party are shared.
* Providing party was not given information about the URLs, which was given to the receiving party.
* The number of URLs given to the receiving party is tallied in a secured way.
* URLs belonging to the receiving party are not counted.

Guessing a class label for suspect websites is done by a technique by analyzing their textual and visual contents. When these conditions are met, that website will be assumed to be a phishing website. The different types of the URLs are as follows (Haijun et al 2011),

* <https://signin.ebay.com>
* <https://www.paypal.com/c2>
* <https://ssl.rapidshare.com/premiumzone.html>
* <http://www.hsbc.co.uk/1/2/HSBCINTEGRATION/>
* <https://login.yahoo.com>
* <https://www.mybank.alliance-leicester.co.uk/index.asp>
* <https://www.optuszoo.com.au/login>
* <https://steamcommunity.com>

**2.2.4 Content-based Approach**

A novel content-based methodology for detecting phishing web sites CANTINA examines the content of a web page to decide if it is genuine or fake. This is in contrast to other approaches that only take a shallow view of the surface characteristics of a web page, for example the URL and its domain name. CANTINA makes use of the well-known Term Frequency/Inverse Document Frequency (TF-IDF) algorithm that is very commonly used in information repossession and more purposely, the Robust Hyperlinks for surmounting broken hyperlinks. Heuristics can notice phishing attacks as soon as they are launched, without the requirement to wait for blacklists to be updated. However, attackers may be capable to devise their attacks to keep away from heuristic detection. In addition, heuristic approaches frequently turn out to be false positives (wrongly labeling a rightful site as phishing). Blacklists might have an advanced level of accuracy, although usually necessitate human intervention and confirmation, which may use an immense amount of resources. At a latest Anti-Phishing Working Group meeting, it was stated that phishers are beginning to use onetime URLs, which direct someone to a phishing site the first time the URL is used. However, it directs people to the legitimate site afterwards. This and other novel phishing tactics significantly obscure the procedure of compiling a blacklist, and can decrease blacklists’ effectiveness (Yue et al 2007).

**2.2.5 Source code based approach**

The Semantic Link Network (SLN), which is a self–organized data model (Liu et al 2010) for semantically organizing web resources. In this method the SLN is constructed and reasoned in three major steps:

* The associated web pages related to the suspicious webpage are retrieved. They are retrieved from two resources. One is from forward links contained in the suspicious web page, and the other is from a powerful search engine, which returns candidate web pages with similar text content to the suspicious web page.
* The Semantic Link network is constructed from the suspicious web page and its associated web pages.
* Reasoning is conducted to mine the implicit association relations, which are defined as the relations among all web pages, which include the suspicious web page, and its associated web pages.

The method (Mona and Omar 2011) discussed the source code checking in the following ways:

* Searched all the images in the website and if there any images work from another website or have links from another place it will consider it a phishing character, so all images should be in the website folder like this <img src="Logo.PNG" border="0" width="1243" height="302">. 􀁸 The program is checking the login or submit button and should make the button action on the website like: login.php. If the button action link to any IP likes this: 103.838.39.0/login.php or email or script it will consider a phishing character.
* Also it checks for iframe, domain, script tags and popup window and if it is found, it is considered as a phishing character.
* If there are more than one phishing characteristics of the same type such as more than one popup window, it will be considered as phishing character.

**2.2.7 Data Mining Approach in Phishing**

Data mining techniques are known to develop the evaluation of phishing attacks (Xi Chen et al 2011). The supervised classification technique is a commonly adopted technique; is a stream of data mining applied to assess the intensity of various phishing attacks. The hybrid approach is characterized by a combination of the bit key phrase extraction and the supervised classification method. It utilizes both; the textual data description of the phishing attack and the financial data of the company under study so as to study the intensity of phishing attack. This is done in accordance with the risk level/ financial loss generating potential of the subject. The adversary aware classifier model: Support Vector Machines (SVM) model is basically viewed as a signaling game where the beliefs, strategies and probabilities of the messages types are all rationalized and built-in as prior knowledge, as soon as a new e-mail arrives. This has permitted the classifier to modify the margin error parameter dynamically as the game progresses. More exclusively, this is considered as an important aspect in the miss-classification constraint of the

optimization problem of the SVM algorithm (Sebastian and Gaston 2013).

The Instance Based Learning (IBL) algorithm is a variant of the nearest neighbor algorithm where, in addition to the conventional system, the method

1. Normalizes various attributes ranges,
2. Processes instances incrementally, and (iii) has a simple policy for tolerating missing values (Kruegel et al 2005). In the Decision Tree (DT) algorithm is a simple algorithm that is based on a set of rules which is advantageous owing to the sequential structure of the decision tree branches. The conditions and actions that are significant are inter-linked directly to supplementary conditions and actions. However, insignificant conditions are ignored.

The boosting method is a known to be a well established scheme for improving the performance of any particular classification algorithm. It constructs a highly accurate classification rule by combining various simple and moderately accurate hypotheses into single and independent one (Islam and Zhou 2008). Different types of machine learning algorithms give high accuracy in detecting phishing websites (Daisuke et al 2009). There are also several existing techniques that include several features that can be conceived, while recognizing a relevant and representative subset of features to create a precise classifier (Ram et al 2012).

**2.2.8 Single Password Mechanism**

The recent solution OTP (one time password), is an important financial security measure that is being used to defend session attacks. On the contrary, it is inflexible to execute differentiated OTP creation mechanisms in them (Yunlim et al 2013). The Single Password protocol (SPP) allows a user to use one single password (for a single user name) for all accounts; reducing the risk to malicious server attacks. A typical SPP uses two basic techniques: challenge/ response and one-time server-specific tickets. The succeeding content explains the working mechanism. For example, consider ‘P’ to be a single password that a client ‘C’ remembers. When ‘C’ registers with a server ‘S’, it generates a challenge and the ticket verification information is sent and ‘S’ stores them in its password file. Later on, when ‘C tries to login on ‘S’, and ‘S’ prompts ‘C’ with the stored challenge. Then ‘C’ uses the challenge, the server’s name ‘S’, and password ‘P’ to issue a one-time server-specific ticket, collectively with a new challenge and a new ticket verification information is sent to ‘S’. After that, ‘S’ verifies the received ticket by means of the stored ticket verification data. If the ticket is valid, then the verification of ‘C’ is successful, and ‘S’ swaps the stored challenge. Next, the ticket verification information, new ticket verification information and the one-time server-specific ticket from ‘C’ are essential (Mohamed et al 2007).

****

**Figure 1: Phishing Mechanism**

**2.2.9 E-Mail Phishing**

E-mail is the most used Internet service in the recent past, and has been the main resource exploited in phishing (Cleber et al 2011). The Simple Mail Transfer Protocol (SMTP), which is commonly adopted in sending emails, allows anyone to forge the sender’s address (Herzberg 2009). The methods that are used to spread phishing through e-mails are quiet similar to spam messaging (Kanich et al 2008). Phishing by itself can be viewed as a sub-category of spam; perhaps inter-linked (El-Alfy and Abdel 2011). Some phishing e-mails are restricted according to the other specified requirements (Jasmine 2008).

An integrated information-processing model of showing the risk to phishing (Arun et al 2011) studied how one may understand the process of phishing emails. The factors that manifest the cognitive valuation process and the outcomes of such an evaluation would affect an individual’s risk to phishing. Prejudiced by cognitive and information processing activities individuals levels of motivation, personality, beliefs, knowledge, and their day-to-day experience may be affected. An integrated model has been studied using a structural equation modeling technique, on a sample of intended victims of an actual phishing email that occurred in a well established university in northeast USA. This type of model concentrates on four contextual factors:

1. Technological efficacy
2. The individual's level of involvement
3. Email load, and
4. Domain specific knowledge. The email inbox is undoubtedly a dangerous place. But using the pattern recognition tools it may be convenient to filter a major portion of the elements that would damage the end users (Sebastin 2013).

The features which are identified in e-mail headers (Cleber et al 2011) are mentioned below:

* Hyperlinks with visible text such as a URL, but pointing to a URL dissimilar from the observable text
* E-mail body coded in HTML format
* Too extensive URL
* Number of domains and sub domains in the URL
* Hyperlinks with any visible text, but pointing directly to an IPbased URL
* Images origin as an IP address
* Sender domain that are different from some URL domain in the message body
* Images with external domain different from the URLs in the message body

**2.2.10 Session Hijacking**

Several web-based features use a type of session management in order to create a user-friendly environment. Sessions are saved on a server and then are associated with their respective users using session identifiers (IDs). Generally, these session IDs appear as a simple target for attackers, who by getting access to them, effectively hijack the users’ identities (Mitja 2002).

Even if a token is correctly produced and remains unpredictable, attackers would intercept it. Another means to intercept these tokens is by identifying them from log files. These include: browser logs, server proxy logs and, web server logs. If the token is passed as a URL variable, an

attacker can easily read it on the log. In order to lower the temporal window for such attacks, these sessions require being as short as possible. Some applications supply no means for a session’s running out, which facilitates attackers to use multiple values trial and error before a session expires. When a user logs out, the server can delete such a token from the user’s browser.

However, if the user is capable of sending a previously used token, the server would keep accepting it (Corrado et al 2010). Automatically it performs all the steps in that particular session fixation as how real attackers It can also lighten the load of test operators by only presenting a number of data as follows (Yusuke 2010).

**2.2.11 Limitations of the Existing System**

With regard to the itinerary, the literature survey presented here unveils some limitations in the existing approaches. But client side phishing attack is available in the same side sever side phishing attack is also available. Both play a major role in websites. The views of various researchers are used in visual similarity and E-mail phishing. Researchers have not yet explored the contribution of URL verification with various issues. However, these approaches fall short of addressing the following issues in particular.

* The existing tools are have limited protection, and it will give the load to the client browser and there is no up to date update of URL database.
* The common string in the URL mechanism is not applicable if the phishing URL is well structured.
* The credibility of the webpage content was not checked using reliable parameters. The features of the contents are not checked with the correct classification tools.
* More response time for some of the systems like visual similarity, semantic Link Network Connection. Phishers are acting very speedily and gently so the response time of detecting the phishing should be very low.
* Forms of the various WebPages are not validated properly and all types of Heuristics are not considered. Login forms only validated in most of the systems.
* Finding the phishing target is the most important issue in the websites, but the existing techniques are not using the effective method for finding the phishing target. They did not use the inferring rules and strategies.
* The existing e-mail phishing tools require more user action. The response time is too high. Mails will not be marked as spam by default. Most of the e-mail phishing tools work in Post Office Protocol.
* The features set are not the convinced one in the existing systems. The major issue is the preparation of the feature sets to be used in the data mining algorithms.

****

**Figure 2. Types of Phishing Attack**

****

**Figure 3: Life Cycle of Phishing Attack**

**2.3 MOTIVATION OF THIS WORK**

The work presented in this thesis presents the multi-stage filtering technique, that enhanced the classification of phishing and legitimate websites, and finding the phishing target for preventing the phishing. The client and server side’s phishing attacks are analyzed, and better false negative and false positive results have been obtained. This work implements a multi-stage filtering approach to overcome the limitations of the existing systems, including processing time and similarity and compared the model with the supervised learning methods. Thus, suitable models based on different techniques for better classification of the result and predictions of the phishing targets are presented.

The following are some of the solutions to tackle the detection and prevention of phishing websites and e-mails.

* To identify the different formats of the URL of the websites and the hyperlinks of the source code, with the URL Validator and the black listed and white listed verification.
* To effectively represent the phishing target and phishing websites, the parse tree with the top hyperlinks of the specified domain name, is constructed and validated.
* To detect the phishing website based on the heuristic techniques using this novel approach, which has more precision recall and F-score values over the existing clustering techniques.
* To speed up the phishing detection processing time in comparison with the existing detection methods.
* To effectively prevent the usage of phishing websites based on server side phishing prevention techniques.

**CHAPTER THREE**

**RESEARCH METHODOLOGY**

**3.1 PHISHING TECHNIQUES AND COUNTERMEASURES**

Various techniques are developed to conduct phishing attacks and make them less suspicious. Email spooling is used to make fraudulent emails appear to be from legitimate senders, so that recipients are more likely to believe in the message and take actions according to its instructions. Web spooling makes forged websites look similar to legitimate ones, so that users would enter confidential information into it. Pharming attracts track to those forged websites. Malware are installed into victims' computers to collect information directly or aid other techniques. PDF documents, which supports scripting and callable forms, are also used for phishing.

**3.2 EMAIL SPOOLING**

A spoofed email is one that claims to be originating from one source when it was actually sent from another. Email spooling is a common phishing technique in which a phisher sends spoofed emails, with the sender address and other parts of the email header altered, in order to deceive

recipients. Spoofed emails usually appear to be from a website or financial institution that the recipient may have business with, so that an unsuspecting recipient would probably take actions as instructed by the email contents, such as:

* Reply the email with their credit card number
* Click on the link labelled as \view my statement", and enter the password when the (forged) website prompts for it
* Open an attached PDF form, and enter confidential information into the form (Section 3.5)

**3.3 WHY IT'S POSSIBLE**

Simple Mail Transfer Protocol is the Internet standard protocol used for electronic mails. Its objective is to transfer mail reliably and anciently, but core SMTP doesn't provide any authentication. An important feature of SMTP is its capability to transport mail across multiple networks, referred to as \SMTP mail relaying". Basically, receiving and relaying SMTP servers need to trust the upstream server; so it is feasible for a malicious user to construct spoofed messages, and talk with receiving or relaying SMTP servers directly to deliver such a message.

As RFC 5321 suggests, SMTP mail inherently cannot be authenticated at the transport level; real mail security lies only in end-to-end methods involving the message bodies, such as Pretty Good Privacy (PGP) and Multipurpose Internet Mail Extensions (S/MIME). However, there is a high cost to deploy those digital signature based countermeasures, because users are reluctant to install an additional piece of software, and they don't have enough knowledge on how to manage the trust.

**3.4 SPF**

Sender Policy Framework (SPF) is an open standard specifying a technical method to prevent sender address forgery. Since most SMTP servers are mutually-TCP-addressible hosts on the public Internet, receiving and relaying SMTP servers are able to see the IP address of the sending host. SPFv1 protects the envelope sender address, the HELO domain and the MAIL FROM address, by verifying sender IP addresses: SPFv1 allows the owner of a domain to specify a list of IP addresses that are allowed to send emails from their domain, and publish this information in the domain's DNS zone; a receiving server may query DNS to check whether the message comes from one of those whitelisted addresses.

For example, cs.arizona.edu publishes the following SPF record:

v=spf1 a:gandalf.email.arizona.edu a:frodo.email.arizona.edu a:pacer.email.arizona.edu a:gremlin.email.arizona.edu a:optima.cs.arizona.edu

This SPF record lists 5 hostnames, and these hosts are allowed to send emails on behalf of @cs.arizona.edu; \~all" disallows any other hosts to send emails from this domain.

**3.5 DKIM**

Domain Keys Identified Mail (DKIM) allows an organization to take responsibility for transmitting a message in a way that can be verified by a recipient. The author, the originating sending site, an intermediary, or one of their agents can attach digital signatures onto a message. The message headers and body, including the originator address (the From header held), are signed. The DKIM-Signature header held includes the signature, the signing domain, and information about how to retrieve the public key.

A DKIM signature generated by Gmail looks like:

DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;

d=gmail.com; s=gamma;

h=domainkey-signature:mime-version:received:received:in-reply-to:references:date:message-id:subject:from:to:content-type;

bh=rdk+ZKX52H558uYXf2No2gW+cp8RKaZBZwyOM+LufnE=; b=dw0s8c2uuBIqY8msh1266XyGlTDxYGwIBmuVPpkMEUGh2mrhWaUwSWYUn0KHSh

vlwVBTiLGRQ8t8KYk1XdMveBnE3iaXlOGiGKlQLqIQjyd+sxbc8OSGHxcO05BpO

3Egb/pf+i8m9iktEjN4PPhLKsyiniNO8vy8LqC33zjyiVw=

The signing domain publishes public keys as TXT records in their DNS zone. To verify this signature, a receiving server may query DNS name gamma. domainkey.gmail.com (constructed from tags \s" and \d" of the signature) and get a TXT record such as:

k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDIhyR3oItOy22ZOaBrIVe9m/iM

E3RqOJeasANSpg2YTHTYV+Xtp4xwf5gTjCmHQEMOs0qYu0FYiNQPQogJ2t0Mfx9zNu06rfRBD

jiIU9tpx2T+NGlWZ8qhbiLo5By8apJavLyqTLavyPSrvsx0B3YzC63T4Age2CDqZYA+OwSMWQ

IDAQAB

Then the signature can be veri\_ed using RSA and SHA-256 algorithms, as speci\_ed in tag \a".

**3.6 OTHER DETECTION METHODS**

Microsoft's SenderID validates the sending server's IP addresses against a TXT record published

in the originator address's DNS zone.

Heuristic-based detection techniques are proposed to identify phishing emails. For example, a simple heuristic is the observation that emails generated by the same toolkit show a high degree of similarity. Once the heuristic identi\_es a kind of phishing emails, it can be entered into a

blacklist, and further emails will be blocked.

**3.7 WEB SPOOLING**

A phisher could forge a website that looks similar to a legitimate website, so that victims may think this is the genuine website and enter their passwords and personal information, which is collected by the phisher.

Modern web browsers have certain built-in security indicators that can protect users from phishing scams, including domain name highlighting and https indicators. However, they are often neglected by careless users.

**3.7.1 How web spooling is done?**

Creating a forged website It's trivial to clone the look of a website by copying the front-end code; a little bit of web programming is necessary to redirect user's input into a \_le or database, then show a \website under maintenance" notice.

Proxy software such as squid or Fiddler2 could be extended to create a fully functional clone. Users can successfully sign in and use all the services provided by the original website, while all the inputs are collected by the server, and all the pages may be modified by the server.

**3.8 DOMAIN HIJACKING**

A more advanced pharming attack is domain hijacking. In domain hijacking, the DNS delegation

record at the domain registrar is changed to a name server controller by a hacker, so that all track can be redirected globally.

Baidu, the largest search engine in China, was hacked by Iranian Cyber Army in January 2010.

1. The hacker chatted with technical support of Register.com, the domain registrar of baidu.com, to change the email address on \_le. The change was approved without careful verification.
2. Account password was reset with the new email address.
3. Delegation record was changed to a name server controlled by the hacker.
4. Millions of users were redirected to hacker's server for 4 hours.

A phisher could also use similar techniques to gain control over a domain.

**3.8.1 Pharming in smaller scope**

DNS cache poisoning and domain hijacking are effective in a large scope, so they would be quickly found and axed. There are techniques for pharming in a smaller scope, such as the local computer or a home network, that can possibly remain unnoticed for a longer term.

The hosts le is a text on local computer that contains hostname-to-IP mappings. This \_le is located at /etc/hosts in UNIX systems, or %WINDIR%nsystem32ndriversnetcnHOSTS in Windows systems. TCP/IP stack consults this \_le before querying DNS. This could be written by malware for pharming.

ARP spooling can manipulate track in local Ethernet, including redirecting track to phishing

server. It can be implemented in malware.

In regions of world that Internet is restricted, some people solutions that claim to provide

uncensored access to Internet. These solutions usually come as software, VPN, proxy server, or

hardware home router; they could be either paid or free. A dishonest provider could over such

a solution with pharming built-in, and users looking for uncensored access would end up visiting

forged websites without realizing this.

**3.9 MALWARE**

Malware is a piece of software developed either for the purpose of harming a computing device

or for deriving benefits from it to the detriment of its user. Malware can be used to collect

confidential information directly, or aid other phishing techniques.

Client security products are able to detect and remove malware and other potentially unwanted

programs, but phishers can make malware undetectable. Financial institutions and online game

vendors distribute security programs to protect their customers.

**3.9.1 Phishing with malware**

Malware can be used to collect confidential information directly, and send them to phishers. Keystrokes, screenshots, clipboard contents, and program activities can be collected. Password input box, where letters are shown as asterisks, can be easily read with a program. Malware can also display a fake user interface to actively collect information. Collected information can be automatically sent to phishers by email, ftp server, or IRC channel.

Malware can also aid other phishing techniques. For web spooling, it can install phisher's CA public key into local computer's trusted CA list. For pharming, it can change the hosts \_le or DNS settings, or even run ARP spooling on local Ethernet. Malware can also enlist the computer into botnets, to send spoofed emails or act as a webserver of forged websites.

**3.9.2 Detecting malware with client security products**

Client security products are widely deployed. Microsoft Update also pushes \Malicious Software

Removal Tool" monthly, which is a lightweight malware scanner.

However, they are not always effective. It's easy to modify a program so that it doesn't contain any known signature, to bypass signature-based detection. There are also techniques to bypass certain behavior-based detection.

**3.10 Phishing through PDF Documents**

Adobe's Portable Document Format is the most popular and trusted document description format.

This makes PDF documents more susceptible to phishing threats, owing to their portability and interoperability on multiple platforms. In addition to being a powerful document format, PDF is a comprehensive programming language of its own dedicated to document creation and manipulation with strong execution features. Some critical functions of a PDF language could be misused by an attacker or a hacker to design a PDF document to his/her own advantage and extract the desired information from the victim, thereby creating a new worldwide threat. These potentially dangerous functions include Open Action and Submit Form.

Although Adobe has implemented some security mechanisms in Adobe Reader and Adobe

Acrobat in order to alert the user in case of (potentially) malicious attempts, these alert measures

are just message boxes, asking the user to allow or block an action. Unfortunately, such message

boxes are often neglected by users, and it is possible to bypass these security mechanisms by

modifying RdLang32.FRA and AcroRd32.dll \_les with malware.

Load Page

Accept URL

Download page source

Analyzer page

Is it phishing?

No

Yes

Raise alert

Is there a response to the alert

Calculate the percentage of phishing

Yes

No

Percentage >=80%

Percentage >=50%

Phishing website

Doubtful website

Secure website

Stop

***Figure 1: The program workflow***

**CHAPTER FOUR**

**SYSTEM DESIGN AND IMPLEMENTATION**

**4.1 OVERVIEW OF DESIGN**

A total of 21 subjects with previous experience in online shopping, 11 females and 10 males, were recruited at a college campus. Thirteen subjects (62%) were college students from 11 different majors. All subjects had at least a college education. The average age was 24 (the range, 19 to 34). Fourteen subjects were randomly assigned to use the Web Wallet and the other 7 were in the control group.

To gauge subjects’ experience with online shopping, we asked them which of our 20 selected e-commerce sites they had visited.

All 21 subjects had used Amazon and Yahoo, and 15 or more had used Apple, Target, Travelocity and Bestbuy. On average, each subject had used 9 sites in our study. Before the study, subjects were briefed about the scenario and their role as John Smith's assistant. The subjects were told to be careful with John Smith’s account information during the study.

We personally observed the subjects’ browsing behaviors during the study. We did not interrupt the study except when subjects clicked the “report fraud” button, at which point we asked them to explain why they reported fraud and told them to stop the task at the current phishing site. At the end of the study, we interviewed the subjects by going over the unrecognized attacks to find out why they did not recognize them.

We define the *spoof rate* as the fraction of simulated attacks that successfully obtain John Smith’s username and password without raising the subject’s suspicion. Figure 9 shows the spoof rate of the normal attack with and without Web Wallet protection, and the spoof rate of all the attacks in the Web Wallet study. The Web Wallet protection significantly lowers the spoof rate of the normal attack from 63% to 7% (one-tail *t*(42) = 5.09, *p* < 1e-05).

Of the seven subjects in the control group, two of them reported all the phishing attacks based on (1) the odd URLs and (2) the fact that the login page is not SSL-protected. Note that one subject

believed that four good sites were attacks because of the lack of SSL. The other five subjects were tricked by at least three attacks, but we should also note that the subjects were still tricked by the fake interfaces under the attacks. In particular, five attacks were fake-suggestion attacks and were reported because “*A window popped letting me know that the website was using fraudulent methods to conceal its identity from me.*” But this warning window was itself fraudulent. The other one attack was the fake wallet attack and was reported: “*Not sure how to resolve the disagreement between the Web Wallet UI which reports ‘radioshack.com’ and the Address Bar of IE which reports ‘radioshack.no-ip.info’. This seems suspicious.*” Again, this subject did not seem to suspect that the Web Wallet interface itself was fake.

**4.2 SYSTEM IMPLEMENTATION**

Detect the phishing websites by checking the webpage source code, we extract some phishing characteristics out of the W3C standards to evaluate the security of the websites, and check each character in the webpage source code, if we find a phishing character, we will decrease from the initial secure weight. Finally we calculate the security percentage based on the final weight, the high percentage indicates secure website and others indicates the website is most likely to be a phishing website.

The environment built by using Microsoft Visual Studio 2010 and use C# programming language.

**Table 1: Phishing characteristics classification**

|  |  |
| --- | --- |
| **Phishing characteristics** | **Phishing characteristics risk** |
| Https | Medium |
| Images | Low |
| Suspicious URLs | High |
| Domain | Medium |
| Email | High |
| Iframe | Low |
| Script | High |
| Popup window | Low |

**Data Mining Analysis**

**Study of the Anti-Phishing Tool with Class Imbalance Problem**

In the proposed system, we have divided collected websites into a training data set and a testing data set for training the different classification model. The reason for taking different classification algorithms is to train the proposed model, to compare their detection performance and select one of that to perform the best result. Table 1 shows the classification result of the hitting websites for all the assigned servers.

Table 2: **Classification Results of Hitting Websites**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | **Legitimate /Phishing Sites** | **TP** | TN | **FP** | **FN** |
| **Server-1** | 2250 / 580 | | 2248 | 579 | 01 | 02 |
| **Server-2** | 1560 / 470 | | 1554 | 466 | 04 | 06 |
| **Server-3** | 1480 / 410 | | 1478 | 409 | 01 | 02 |
| **Server-4** | 1150 / 540 | | 1142 | 536 | 04 | 08 |
| **Server-5** | 1250 / 280 | | 1249 | 275 | 05 | 01 |
| **Average:** | **7690 / 2280** | | **7671** | **2268** | **12** | **19** |

Where TP – True Positive, TN – True Negative, FP – False Positive and FN – False Negative.

In the above result, the value of Precision, Recall and F1-measures can be calculated by using the equations of Class Imbalance Problem. The results are as follows:

Table 3: **Comparison of Precision, Recall and F1-Measure**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-measure (%)** |
| **Server-1** | 0.999555 | 0.999111 | 99.93332 |
| **Server-2** | 0.997433 | 0.996154 | 99.67928 |
| **Server-3** | 0.999324 | 0.998649 | 99.89861 |
| **Server-4** | 0.99651 | 0.993043 | 99.47735 |
| **Server-5** | 0.996013 | 0.9992 | 99.76038 |
| **Average:** | **0.99776** | **0.99723** | **99.75** |

The reason for taking 6 digit values after decimal to show the accuracy of the result. The above result shows the proposed system is producing around 99.75% success result at a particular time period.

Table 4: **Phishing Characteristics and the Risk Status of the Accessed Website**



**4.3 SYSTEM REQUIREMENT**

Although the Web Wallet is security software, it will be used by the human user. Therefore, before we implement a full-featured

Web Wallet, we started with a prototype and ran a controlled user study to test both its usability and its effectiveness at preventing phishing attacks. The tested Web Wallet prototype only supports login information, not credit card or bank information. And the backend is mostly hard-coded for the web sites used in the study. We used the same scenario from previous studies of anti-phishing toolbars. A subject was told to act as the personal assistant of John Smith. John Smith forwarded 20 emails to the subject and asked her to go to 20 different web sites, log in with his password, and add items to his wish list. We did not include any tutorial about the Web Wallet in the study, since few users in the real world read tutorials. The Web Wallet interface must be selfexplanatory.

Five of the 20 forwarded emails were attacks, with links leading the subject to phishing web sites. Phishing attacks were simulated by connecting to the real web site but changing the browser’s address bar to display a different hostname (indicating that the web page was from an unusual source). We simulated the ideal phishing attacks whose content is a perfect copy of the actual site.

Previous studies showed that attacks using a similar hostname (*e.g.*, www.amazon-department.com to spoof www.amazon.com) have the highest spoof rate, compared with attacks using an IP address or a totally different hostname. In this study, all the attacks displayed a URL in the address bar with a hostname similar to the legitimate site. All the attacks did not use an SSL connection. Six out of the 20 tested web sites in the real world do not use SSL to protect their login pages.

**4.3.1 Simulated Phishing Attacks**

Among the five attacks, one attack represents a normal phishing attack. In this attack, the phishing site, either copying the legitimate site or acting as a man-in-the-middle between the user and the real site, uses the same HTML login form as the one in the legitimate site. Therefore, the login form can be detected and disabled by the Web Wallet. The user has to open the Web Wallet to login to the phishing site. Any new security interface should also be tested with new potential attacks. The other four attacks are designed to specifically target the Web Wallet interface.

**Undetected-Form Attack***:* The Web Wallet uses some heuristic rules to analyze the HTML source code of the current page to detect the login form. It is possible that a phishing site manages to bypass the Web Wallet detection.

In this attack, the login form is not disabled. Note, however, that any typing at the undetected form still makes our negative visual cues appear – *e.g.*, the password characters zoom out of the screen

**4.4 HARDWARE REQUIREMENT**

When the user presses F2, the Web Wallet is opened as a browser sidebar. The main interface contains two parts: a card presentation area and a card folder. The card folder displays stored login cards. The stored login cards are encrypted using the Web Wallet’s master password. Users will not be prompted for the master password until they first interact with the card folder, *i.e.*, when they are retrieving a stored card or saving a new card. When there is a stored card that matches the request from the web page (this case happens when the user has sent the same card to this site before and has agreed to save that card into the Web Wallet), the stored card is displayed in the card presentation area and the stored information is automatically filled into the login form. The user then clicks the submit button in the login form. This procedure is free from phishing because the user has submitted the same information to this site before.

When there is no stored card matching the request, a new login card is displayed (figure 3b). The Web Wallet does not require users to save their login information. They can only use the Web Wallet interface to do secure submission. The new login card shows the domain name of the site and a description of its trustworthiness. The new login card provides input fields for the user to type in her username and password.

The new login card has a “save card” checkbox, which tells whether or not to save the new login card in the Web Wallet for future use.

Below the checkbox is a submit button. Depending on the site’s trustworthiness and the user’s history, the button displays different labels and performs different actions. If the site is rated as trusted, the button shows “Fill in” and clicking it fills the typed information into the login form. If the site is not rated as trusted, the button shows “Continue”. Note that this does not mean that the site is guaranteed to be phishing; it may simply not meet our criteria in terms of the trustworthiness.

Pressing the button has two sub cases. In one case, if the user is remembered to have logged into this site using the same login information, the Web Wallet simply fills the information into the web page. In the other case, if the user has not logged into this site before, the Web Wallet needs to confirm with her about her intended site, which will be elaborated in the next section.

**4.5 SOFTWARE REQUIRMENT**

The requirement that are used for this project are;

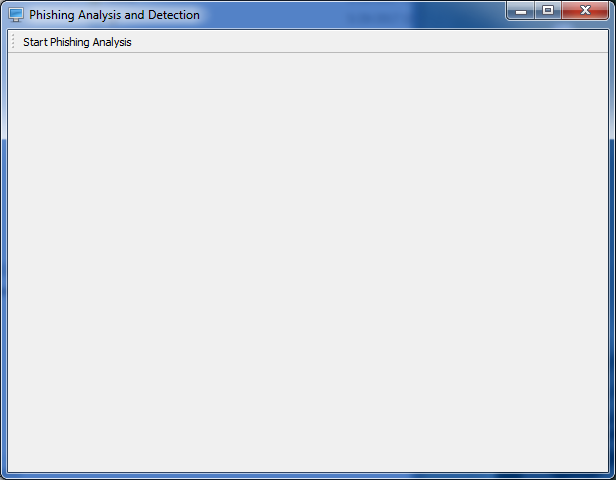
HTML

CSS

PHP

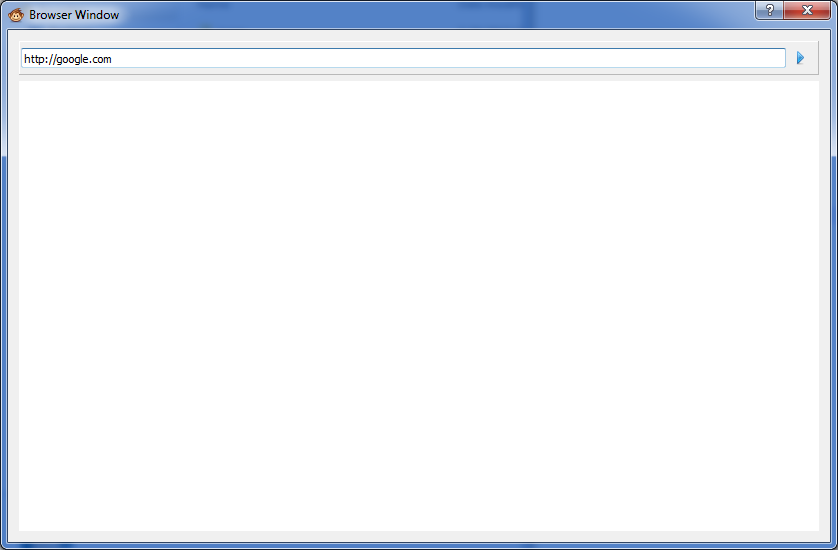
PERL

**4.6 DISPLAY OF THE GRAPHICAL USER INTERFACE**

****

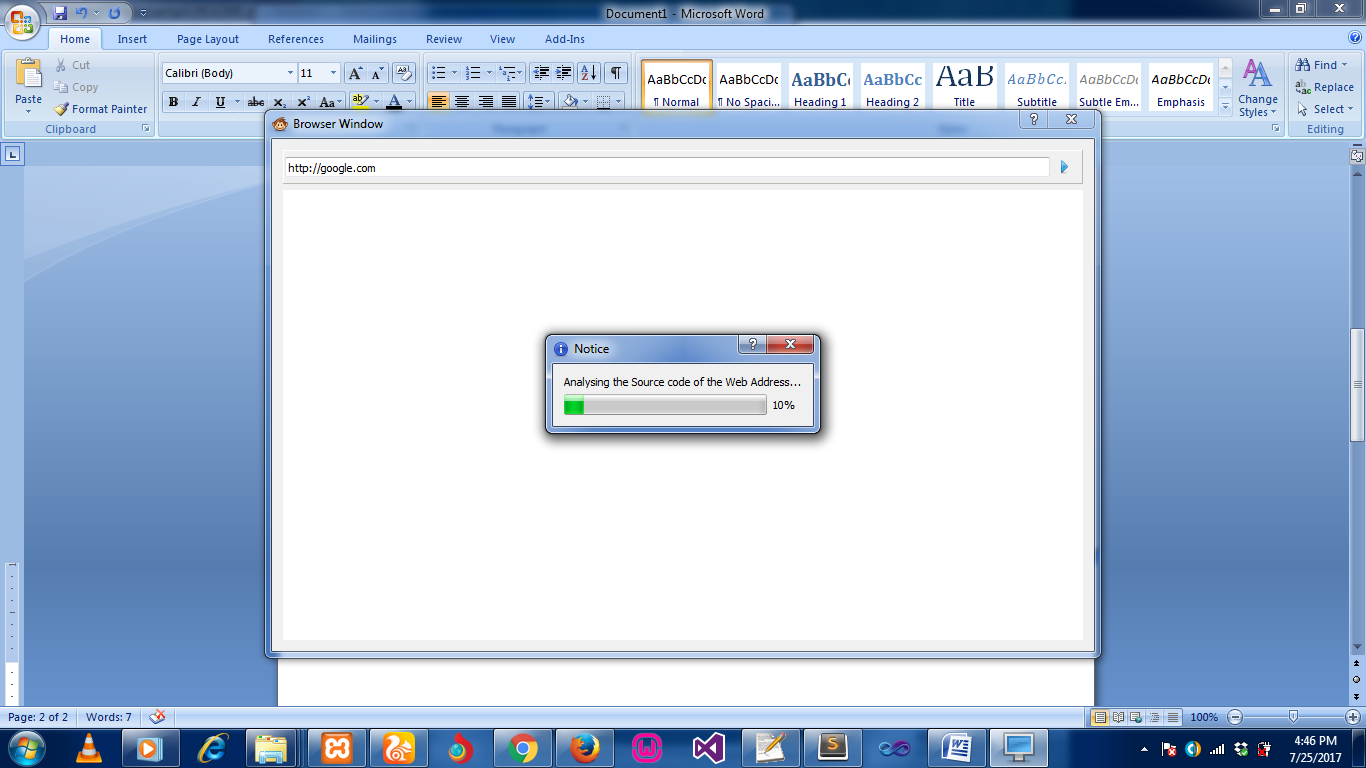
**Figure 1: starting phishing analysis**

Where to insert the URL



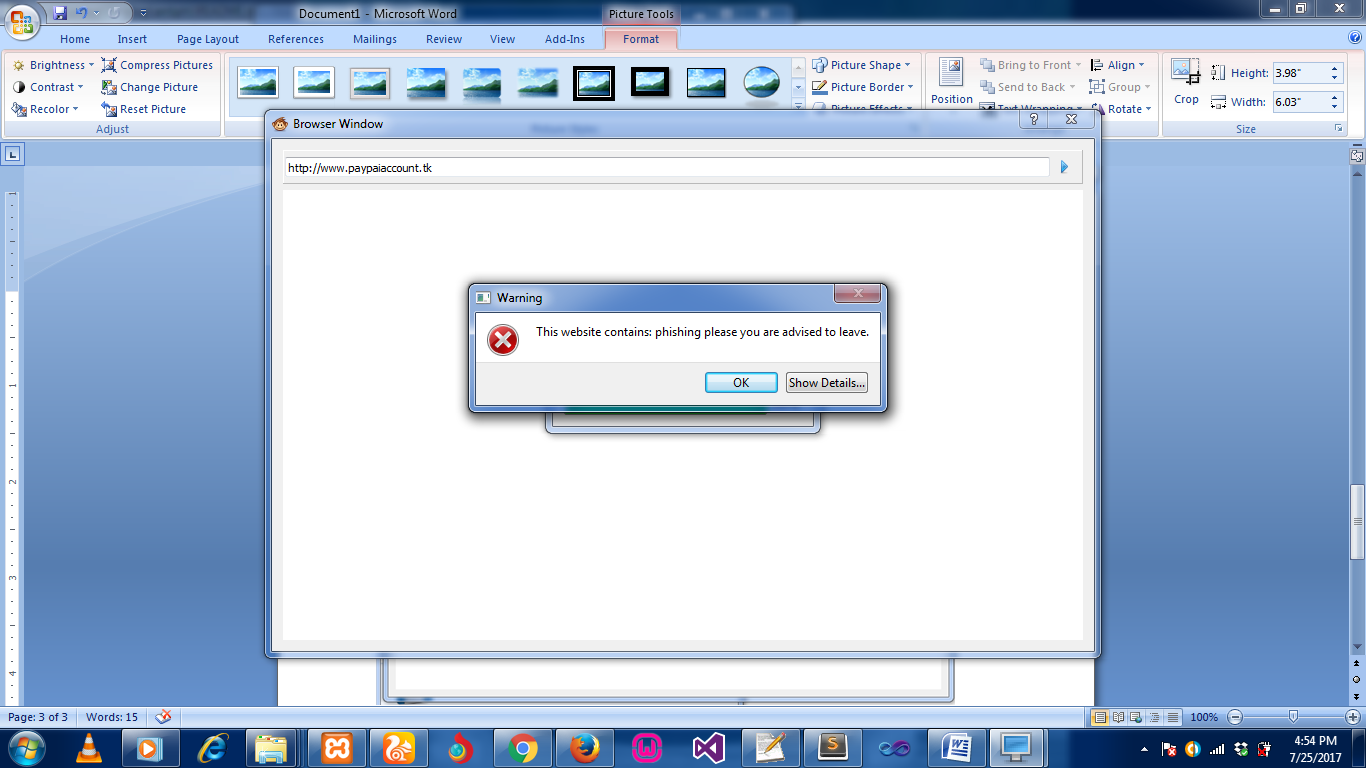
**Figure 2: Insert URL**

Analyzing the url



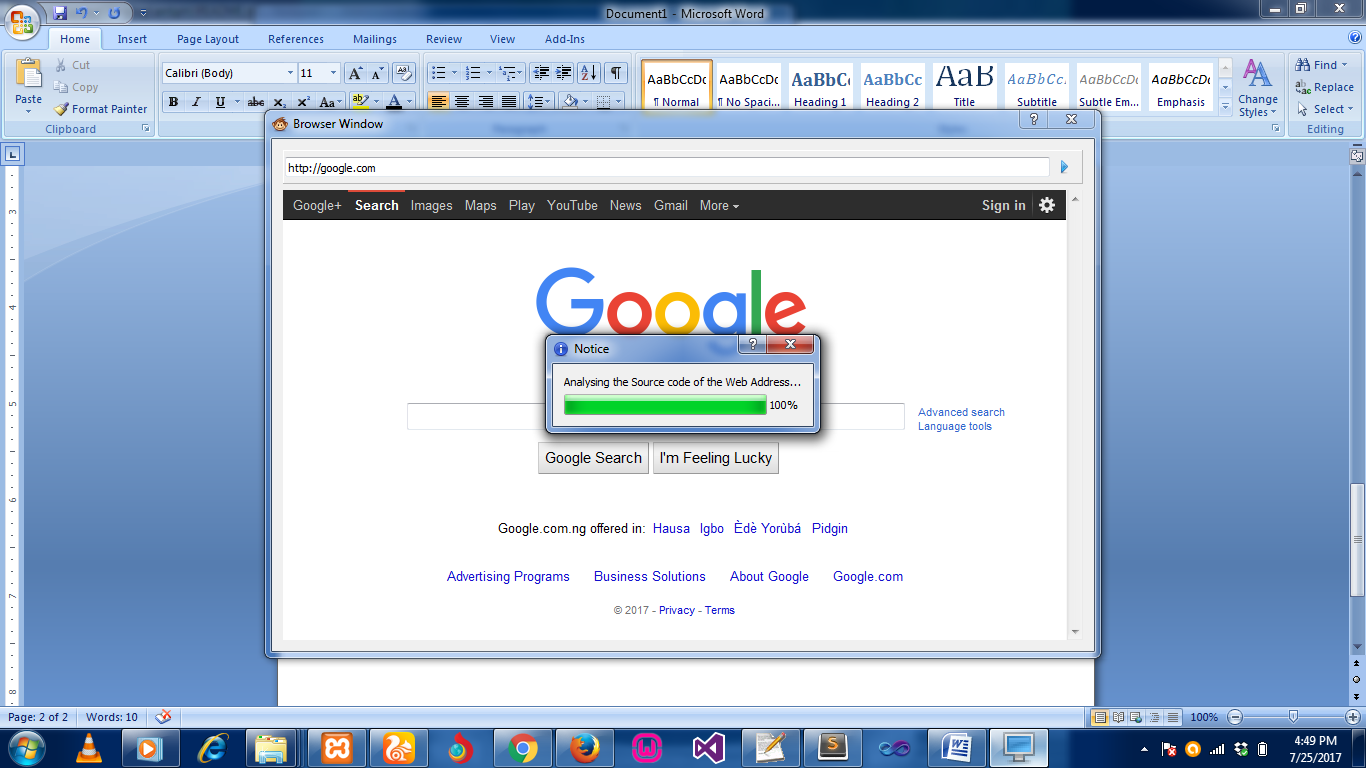
**Figure 3: Source code of the web**

**Output**

****

**Figure 4: Warning of the Page**

**Output**

****

**Figure 5: The result of Source code of the web**

**CHAPTER FIVE**

**CONCLUSION AND RECOMMNEDATION**

**5.1 CONCLUSION**

In this paper we proposed a phishing detection approach that classifies the webpage security by checking the webpage source code, we extract some phishing characteristics out of the W3C standards to evaluate the security of the websites, and checked the webpage source code, if we find a phishing character, and we will decrease from the initial secure weight. Finally we calculated the security percentage based on the final weight, the high percentage indicates secure website and others indicates the website is most likely to be a phishing website. We checked two webpage source codes for legitimate and phishing websites and compare the security percentages between them, and we found the phishing website is less security percentage than the legitimate website. In Future work we can add other checks in the program and check more source codes contains many languages in it like PHP, CSS, Perl, HTML etc. Also, we can develop a browser plug-in to check the webpages and informs the user if there any possible attack*.*

**5.2 RECOMMENDATIONS**

Phishing is an appalling threat in the web security domain. In this attack, the user inputs his/her personal information to a fake website which looks like a legitimate one. We have presented a survey on phishing detection approaches based on visual similarity. This survey provides a better

understanding of phishing website, various solution, and future scope in phishing detection. Many approaches are discussed in this paper for phishing detection; however most of the approaches still have limitations like accuracy, the countermeasure against new phishing websites, failing to detect embedded objects, and so forth. These approaches use various features of a webpage to detect phishing attacks, such as text similarity, font colour, font size, and images present in the webpage. Text based similarity approaches are relatively fast, but they are unable to detect phishing attack if the text is replaced with some image. Image processing based approaches have high accuracy rate while they are complex in nature and are time-consuming. Furthermore, most of the work is done offline. These involve data collection and profile-creation phases to be completed first. A comparative table is prepared for easy glancing at the advantages and drawbacks of the available approaches. No single technique is enough for adopting it for phishing detection purposes. Detection of phishing websites with high accuracy is still an open challenge for further research and development.

**REFERENCES**

Felten, E.W., Balfanz, D., Dean, D., Wallach, D.S. Web Spoofing: An Internet Con Game. Proceedings of the 20th National Information Systems Security Conference, 1997

Fight Identity Theft Blog. ING Direct Fights Keystroke Loggers. December 11, 2005. http://fightidentitytheft.com/blog/?p=23

GeoTrust. TrustWatch Tools. <http://www.trustwatch.com/>

Herzberg, A. The ‘Unprotected Login’ Inter-Net Fraud League (I-NFL) Hall of Shame.2005. http://www.cs.biu.ac.il/~herzbea//shame/

Herzberg, A. TrustBar: Re-establishing Trust in the Web. 2006. http://www.cs.biu.ac.il/~herzbea/TrustBar/

<http://www.oit.umn.edu/safe-computing/topics/phishing-scams/index.htm>

http://www.w3.org/TR/WAI-WEBCONTENT/

<http://www.passmarksecurity.com/>

Jakobsson, M., Myers, S. Stealth Attacks and Delayed Password Disclosure. http://www.informatics.indiana.edu/markus/stealthattacks.

Johns, M. A Guide to Supporting InfoCard v1.0 Within Web Applications and Browsers. March, 2006. http://www.identityblog.com/?page\_id=412#infocardg\_topic5a

Krebs, B. The New Face of Phishing. The Washington Post. Feb 2006.

Liu, W., Deng, X., Huang. G., Fu. A.Y. An Antiphishing Strategy Based on Visual Similarity Assessment. IEEE Internet Computing, Vol. 10, No. 2, pp. 58-65, March/April, 2006. PassMark Security. Two-Factor Two-Way Authentication.

M. Aburrous, M.A. Hossain, F. Thabatah and K. Dahal, "Intelligent phishing website detection system using fuzzy techniques", in 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA), pp. 1-6, 2008.

N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, "Client–side defense against web–based identity theft", In Proceedings of 11th Annual Network and Distributed System Security Symposium, 2004.

Y. Pan and X. Ding, “Anomaly Based Web Phishing Page Detection", Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06), Computer Society, 2006.