Anti-Spoofing Tool

Project Background Report

May 2011

Author: Mohammed Baihan

Supervisor: Dr. Ning Zhang

School of Computer Science
The University of Manchester

Table of Contents

List of Figures and Tables

Figures

Tables

Abstract

Phishing is an online identity theft that makes use of social engineering and technical subterfuge. Using these techniques attackers can gain individuals' confidential data in order to illegally access their bank accounts. The number of victims of phishing attacks increases dramatically in the last decade. This is because attackers continue developing new phishing techniques and the majority of Internet users do not follow security advice. The role of this project is to provide a solution (SpoofGuard++) to the phishing problem. The proposed solution tries to counter new sophisticated phishing techniques, such as Cross Site Scripting and Tabnabbing, as well as simple techniques. In this project, a literature review of the related works is conducted with a critical analysis on different proposed solutions. In addition, an investigation of new sophisticated phishing techniques is also conducted. These activities lead to gather system requirements of the proposed solution. The majority of the proposed solutions try to combat simple phishing techniques or to address HTML4 vulnerabilities and some of these solutions suffer from bypass techniques. However, the investigation suggests that additional work should be done to mitigate the risks of new phishing threats: HTML5, Cross Site Scripting, URL shortening, HTML attachment and Tabnabbing threats.

Chapter 1. Introduction

Phishing, a term coined in 1996, is online identity theft that makes use of social engineering and technical subterfuge. Attackers use these techniques to steal users' confidential data, for example, bank account ID and password, and then consequently cause harm to Internet users by transferring money from their accounts to the attacker's accounts. The first use of this type of online attacks was on AOL accounts [2].

1.1. Project context

In order to better understand the phishing problem, a reader may need to know about phishing attacks, the consequences of such attacks, and how to counter these attacks.

1.1.1. Phishing attacks

As Huang et al. [1] describe in their paper, a typical phishing attack involves five steps (Figure 1) which can be described as follows:
1- A fraudulent website is developed by an attacker.
2- The attacker sends fake emails to a large number of users. These emails include a link to the attacker's website.
3- A number of unsuspecting users will be lured to visit the attacker's website. On this website the confidential data of these users are exposed to be compromised.
4- The attacker can gain users' confidential data from his fraudulent website.
5- The attacker impersonates the users on the target website using their confidential data. Then the attacker can access the victims' financial accounts.

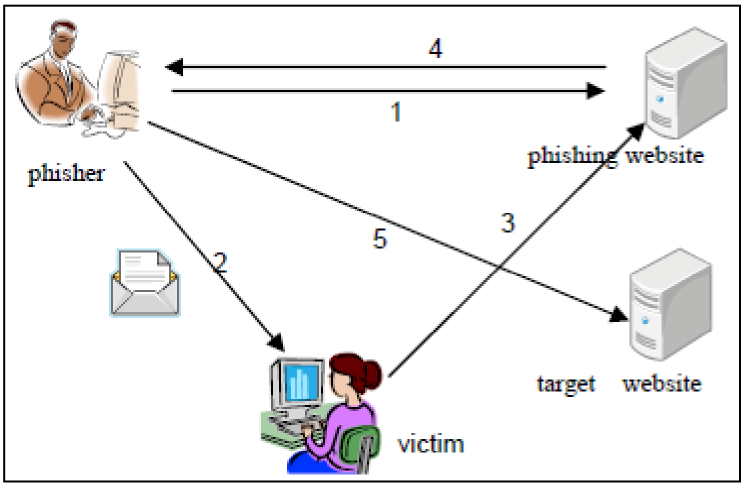


Figure 1: Typical phishing attack steps [1]

1.1.2. The consequences of phishing attacks

Phishing attacks mainly affect individuals. These affects are financially related and legally related [35]. The financial impact of phishing on individuals is the most important issue. Typically, the main target of phishers is gaining access to the individuals' bank accounts. If their attacks are successful, the individuals are likely to lose money from their bank accounts. In addition, an individual may face real legal issues because of phishing. Phishing attacks focus on gathering individual confidential data, such as name and social security number. If these data are used to break the law, the individual will not be able to deny this and hence may face formal penalties.

1.1.3. How to counter these attacks

In order to reduce the risk of phishing attacks, a variety of techniques have been proposed. Some of these techniques are designed to work at server side (server website), while other techniques are developed to work at client site (the browser client). In addition, educating users may contribute in mitigating phishing risks and can be served as a compliment solution to both server and client side techniques. A complete dissection of these techniques can be found in Chapter 2 (literature review).

1.2. Research motivations and challenges

The motivations and challenges of this project are:

1- There are shortcomings in the current anti-phishing techniques and solutions which allow some sophisticated attackers to achieve their targets, for example, blacklist-based solutions are not effective if these lists are not updated [36].

2- The number of victims, whether organisations or individuals, has increased over the last ten years. For instance, in 2008, more than 5 million US Internet users lost major amounts of their money [37].

3- Phishing attackers mainly try to gather users' confidential data [38].

4- Most of Internet users do not follow security advice due to extra effort that security requires [41].

1.3. Aim and objectives

The aim of this project is to provide the anti-phishing industry with a solution that can detect more sophisticated phishing attacks as well as detecting simple phishing attacks. To achieve this project aim, there are some detailed objectives and tasks that are required to be performed:

1- To survey and examine the current techniques and solutions of anti-phishing and gain further knowledge through the understanding of these techniques.

2- To conduct an investigation of new phishing attacks and potential threats.

3- To collect the proposed system requirements.

4- To design the proposed system's architecture.

5- To implement the designed architecture into a working programme.

6- To evaluate the resulting system.

1.4. Project scope

In order to achieve the project' objectives this project's scope should be specified:

1- The development of an Internet Explorer (version 9) plug-in (SpoofGuard++).

2- The Microsoft .NET framework will be used to implement SpoofGuard++ using C# programming language.

3- SpoofGuard++ is an enhanced version of the origin SpoofGuard [3].

4- SpoofGuard++ is intended to mitigate risks of new and sophisticated phishing techniques.

1.5. Report structure

The following sections of this report are organised as following**:**

Chapter 2- Literature review: Previous works in anti-phishing domain are demonstrated in this chapter and are examined to express their advantages and limitations.

Chapter 3- The proposed solution design: The design, implementation and evaluation phases are expressed in details in this chapter.

Appendix A: A Project Gantt chart used for planning.

Chapter 2. Literature review

To mitigate the phishing problem, a number of solutions have been proposed in literature and in industry. These solutions can be grouped into two main categories: server-side and client-side solutions.

2.1. Type 1: Server-side solutions

Server-side solutions are server-based applications that attempt to mitigate the phishing problem. The idea behind server-side anti-phishing solutions is to protect a user from being a victim of a phishing attack by filtering incoming emails, taking action against fraudulent websites, or applying authentication protocols at the recipient's mail server. These solutions make use of email-content analysis, notice-and-take-down, or protocol-based authentication methods.

2.1.1. Email-content analysis method

The email-content analysis method focuses on examining incoming emails to find specific features of fake emails to prevent such emails from reaching the user's inbox. To determine these features, a number of known fake emails are analyzed. These features can be grouped into seven categories: structural, link, element, spam filter-based, style markers-based, structural attributes-based and word-based features. There are a number of techniques which are associated with this method. For example, there is model-based machine learning [14] and property-structure based techniques [18].

Bergholz et al. (2008) [14] propose a model-based machine learning technique. In this technique a new email's features are compared to features of known phishing emails. Then a judgment on the new email is made as to whether this email is fake or normal. This technique uses 27 basic features and different advanced features. The basic features can be grouped into five categories: structural, link, element, spam filter-based and word-based. The structural features are: the total number of body parts, the number of discrete and composite body parts, and the number of alternative body parts. The link features are: the total number of links, the number of internal and external links, the number of links with IP-numbers, the number of deceptive links, the number of links behind an image, the maximum number of dots in a link, and a Boolean, indicating whether there is a link whose text contains one of the following words: click, here, login, update. The element features are four Boolean features with regard to whether or not HTML, scripting, JavaScript, and forms are used. The spam filter-based features are: the filter test score and a Boolean of whether or not an email is considered to be spam. The word-based features are Boolean features of whether or not the words "account, update, confirm, verify, secur, notif, log, click and inconvenien" occur in the email. The advanced features are proposed by the authors. They adaptively trained Dynamic Markov Chains and novel latent Class-Topic Models to generate these features. To compare the new email's features to the proposed features, the technique uses a classifier. Typically, this classifier has two inputs: the values of the phishing emails' features (the training set of the classifier), and the values of the new emails' features (the test set of the classifier). Figure 2.1 gives a general view of the proposed technique.

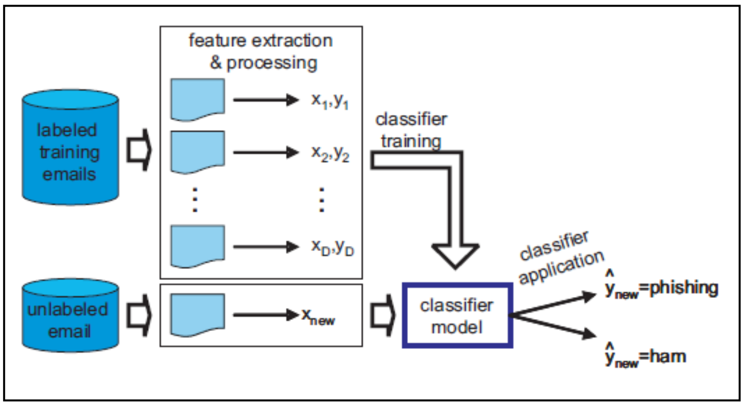


Figure 2.1: The machine learning approach [14]

This technique has one advantage and two limitations:

Advantage

1- The classifier used in the proposed technique can minimize the amount of normal emails that may be classified as phishing emails. The authors [14] claim that the classifier reduces this amount by two thirds in comparison with the work of Cormack et al. [15].

Limitations

1- This technique provides less accurate results in comparison with previous solutions – that of Fette et al. [16] for example. This is because this technique does not use extrinsic-based features such as the age of linked-to domains [14].

2- Since the proposed solution is a statistically-based technique, attackers may bypass it, for example by using HTML layout tricks [17][18].

Another technique is proposed by Chandrasekaran et al. (2006) [18]. This technique makes use of the structural properties of phishing emails to distinguish between legitimate and fake emails. To achieve their target the authors have identified 25 features. These features can be grouped into two categories: style markers-based and structural attributes-based features. The complete list of the features is provided in Table 2.1. The authors used 100 phishing and 100 legitimate emails as input to the simulated annealing algorithm, to identify the useful features. From the relevance between such features, information gain (IG) has been used to rank these features. Based on the candidate features, the authors used the Support Vector Machine (SVM) classifier to classify phishing emails.

Table 2.1: the features used in this technique

Feature category	Feature
style markers-based	Total number of characters
	Total number of unique words
	Word count
	Total number of function words
	Function word frequency distribution:
	Account
	Log
	Access
	Bank
	Credit
	Click
	Identity
	Inconvenience

	Information
	Limited
	Minutes
	Password
	Recently
	Risk
	Social
	Security
	Service
	Suspended
	Total number of words
structural attributes-based	Structure of email subject line
	Structure of the greeting provided in the email body

This technique has one advantage and two limitations.

Advantage

1- The selection of function words features increases the accuracy of this technique as the authors have proved in their experiment [18].

Limitations

1- This technique may not identify some browser vulnerabilities-based attacks [18], such as International Domain Name (IDN) spoofing and pop-up hijacking attacks. This is because the proposed technique focuses only on email-based attacks.

2- This limited number of emails involved in the experiment is not large enough to draw a broader conclusion [18].

2.1.2. Notice-and-take-down method

Another method to combat phishers is to attack their websites before they can start harming any individuals. This can be done by finding these websites' URLs from reported phishing emails, for example, then try to remove these websites from the Internet. Typically, specialist companies play this role as a service to financial organizations. There are a few techniques that follow this method, two of which will be discussed here.

Shah et al. (2009) [19] proposed a technique called Pshark. Essentially, through four stages, Pshark waits for any suspicious emails and, upon detecting a phish website, starts to remove such a website. In the first stage, the system identifies the suspicious email which is currently judged manually. In the second stage, the URL of a phish page will be extracted and a WHOIS query is used to find the host server's IP address and location, and the Server Administrator's details. In the last stage, Pshark sends a message to the host Server Administrator to notify him/her that a phishing website is being hosted on its server. Then the Server Administrator should remove the phishing pages. After that, Phshark periodically checks whether or not the phishing pages have been removed. If such a page still exists, Pshark will act aggressively in one of two ways. Firstly, it will inform the legal authorities that the Server Administrator is responsible for this attack. Secondly, Pshark will apply attacks against the phishing page, for example by flooding the phishing page using false data to reduce the probability of determining correct and false data.

Another notice-and-take-down technique is provided by BrandProtect International Company [20]. It tries indirectly to remove phishing pages upon the receipt of an abuse notification from victims. Using the suspected website's URL, the website is checked and considered as to whether or not it is active, if it still exists, or if it is never-active. If the site is active, the Incident Response Analyst collects information about the ISP and the domain owner and afterwards ensures that the phishing site is removed. In addition, the URLs of such fraudulent websites will be sent to Microsoft, Google and Firefox asking them to add these URLs to their blacklist.

These two techniques have one advantage and three limitations as follows:

Advantage

1- These solutions involve proactive rather than reactive actions, and therefore protect more innocent users from phishing attacks.

Limitations

1- Both of the proposed methods require Server Administrator interaction to remove phishing pages. This dependence on an external body may affect the performance of the solution [19].
2- As these solutions act upon receiving user messages, confidential information of some victims may already have been compromised [21].
3- The Pshark design still lacks an effective email filtering technique [19].

2.1.3. Authentication protocol method

This method tries to solve the phishing problem by adopting authentication schemas. These schemas can be applied on the email protocol (STMP), which is designed without security requirements [6]. Using this method, sender's identity can be examined. This can mitigate phishing risks. A number of techniques, that adopt this method, have been proposed such as senderID [4] and DomainKeys Identified Mail (DKIM) [5].

Microsoft proposes an email authentication technique called SenderID which provides sender authentication based on its path. Typically, before sending an email, a sender can publish a DNS text record which contains allowed IP addresses that can be associated with the sender's domain. Then, before the sender's email reaches its destination, the recipient's mail server can intercept this email and extract the sender's IP Address and the sender's domain by finding the address in the "From:" header. Then, it queries the sender's DNS to retrieve the associated IP address. After that, the recipient's mail server can check the real sender's IP against the associated IP addresses. The email is authentic if it passes this check or is considered to be deceptive otherwise. Figure 2.2 summarizes the senderID technique process.

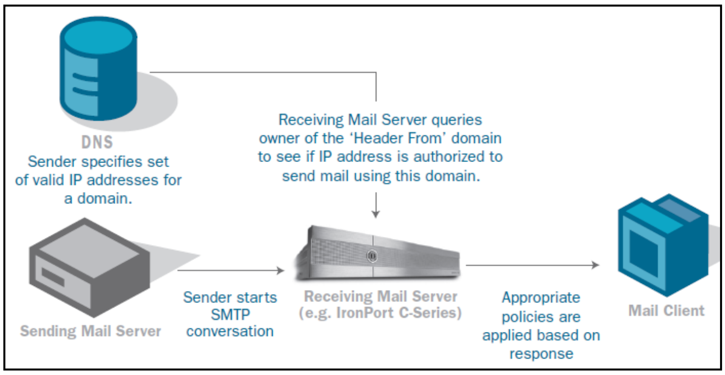

DNS
Sender specifies set of valid IP addresses for a domain.
Receiving Mail Server queries owner of the 'Header From' domain to see if IP address is authorized to send mail using this domain.
Sending Mail Server
Sender starts SMTP conversation
Receiving Mail Server (e.g. IronPort C-Series)
Appropriate policies are applied based on response
Mail Client

Figure 2.2: senderID technique processes [4]

This technique has one advantage and one limitation.

Advantage

1- SenderID is easy to implement since it performs a simple IP address check.

Limitation

1- In a normal email, the “From:” header indicates the sender’s domain, and the IP address that appears in the email is the sender’s IP. However, if the email is redirected using mail forwarding services or mailing lists, the “From:” header still indicates the original sender’s domain, but the IP in that email will be the service provider’s IP. Thus, the sender needs to publish all mail forwarding services and mailing list IP addresses that it may use. Obviously this task is not easy [6].

Yahoo also proposes another email authentication technique called DKIM. It is a cryptographic-based protocol which is used to authenticate the sender’s (server) domain. To do this, a sender can digitally sign an email for authentication purposes. Typically, the sender produces a hash value of each message and encrypts the hash value using the sender’s private key. The corresponding public key is published in a DNS text record. When the recipient’s mail server receives the email, it extracts the sender’s domain that can be found in the “From:” header. Then, it finds the sender’s public key from the DNS text record, and finally checks the signature against the email context. If the signature is valid, the sender is then authenticated. Figure 2.3 summarizes the DKIM protocol processes.

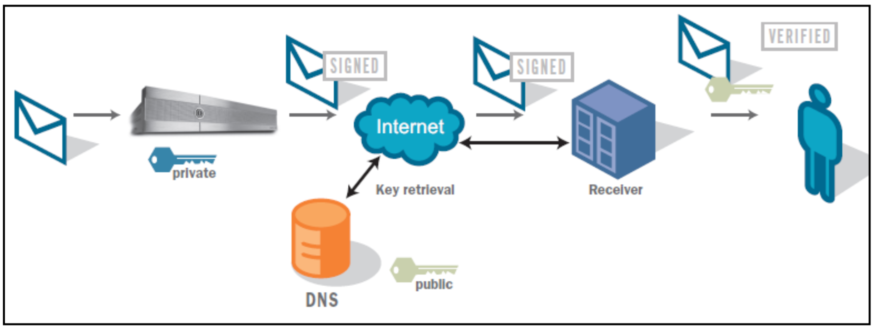


Figure 2.3: DKIM technique processes

This technique has two advantages and one limitation.

Advantages

1- DKIM is a solution for mail forwarding problems. Because this protocol does not check the IP address in an email, this may confuse the senderID protocol. However, instead of an IP address, it verifies a digital signature, which does not change in the case of mail forwarding [6]

2- Before the email reaches its destination, the email contents can be modified. However, using DKIM protocol, the recipient can verify the original message content. First, the original message’s hash value can be produced by decrypting the digital signature using the sender’s public key. Then, using the identical hash algorithm on the current message, the recipient can get the current message’s hash value. Finally, if it is not identical to the original message’s hash value, the authentication will fail [6].

Limitation

1- Sometimes the forwarding services need to modify a message’s content. However, as shown above, this modification will result in authentication fail [6].

2.2. Type 2: Client-side solutions

While the previous category of solutions can be applied on the server side, client-based solutions are designed to work on the Internet users' machines. That is, using plug-ins or browser helper objects (BHOs) which a user can install to monitor visited web pages, and to warn the users if they have entered a fraudulent page. These solutions are different in terms of how to determine if a visited page is fraudulent or not. They can be classified into four groups: blacklist-based or white-list-based, visual-clue-based, webpage-feature-based and information-flow-based solutions.

2.2.1. Blacklist-based method

The majority of anti-phishing methods rely on a blacklist, a list of known phishing domains [1]. This method combats the phishing attempts by preventing user from accessing web pages that appear in the blacklist. To build this list, the method requires retrieving recent uniform resource locators (URLs) of phishing pages from specialist websites such as Anti-phishing Working Group (APWG) or PhishTank, or alternatively may receive these URLs from the users directly. The techniques of Microsoft SmartScreen Filter [22] and NetCraft Toolbar [23] make use of blacklists method.

Microsoft SmartScreen Filter is integrated with the recent Internet Explorer, IE9. This tool uses two methods to determine the nature of a page: blacklist checking and heuristics analyses. Basically, when a user visits a site using IE9, the SmartScreen Filter will compare a page's contents against heuristics characteristics, which are updated periodically using machine learning techniques developed by Microsoft. If suspicious properties are found, the tool will warn the user to avoid providing any confidential data by causing a yellow shield to appear. However, if the page passes the heuristics test, the tool will check its URL against a frequently updated online blacklist. If the URL is found in the blacklist, the page's contents will be blocked, and a red shield will appear in the address bar. The user then has the choice whether to proceed or to close the page. The tool also checks downloaded files against the same blacklist, and the later processes will be applied. SmartScreen Filter provides its user with a reporting feature to notify Microsoft about new fraudulent URLs. In addition, to decrease the false positive detection rate, this tool depends only on verified unsafe URLs provided by reviewers at Microsoft or by employees from third parties. In a network environment, the domain administrator can use a Group Policy feature to prevent users from overriding the SmartScreen Filter. This means that users in this network cannot bypass the warning if it appears, because the option of ignoring such a warning is disabled, thus the users are more secure [24].

This technique has two advantages and one limitation.

Advantages

1- Unlike blacklist-based tools, SmartScreen Filter can protect users from downloadable malicious files that may be used by phishers to collect users' confidential data, for example keyloggers.

2- By preventing users from overriding SmartScreen Filter, an organization network administrator may decrease the possibility for users becoming phishing victims, and hence may protect the organization's confidentiality.

Limitation

1- As with any blacklist-based solution, users are still exposed to new phishing attacks [1]. That is, the URLs of newly established phishing sites may not yet be included in the blacklist.

NetCraft Toolbar is another blacklist-based technique provided to Mozilla Firefox and Internet Explorer users by NetCraft. This tool warns the users through five labels: “since”, “rank”, “country”, “host name” and “risk rating” (see Figure 2.4). Each time a user enters a website, the tool will query the NetCraft Web Server Survey using the website’s URL to retrieve critical information about such a website. The tool then shows the website’s foundation date in the “since” label or prompts “new site” if this website is not found in the Web Server Survey. New sites are given a high risk rating as most phishing sites have this property. The “rank” label indicates how many times a website has been visite0d, and most visited web pages are considered safe by the method. The “country” label displays the place of a website’s host server. For example, if a user enters a barclays.co.uk site, and the country label value is “China”, then the user can identify this site as being fraudulent. In the “host name” label, the website hosting company can be displayed. If the hosting company has a history of hosting phishing sites, the NetCraft Toolbar will increase the risk rating. The “risk rating” label gives an indication of the danger the users face. The tool calculates the rating based on several factors including:

1- The age of the website domain, which NetCraft’s designers consider as the most important factor.
2- Known phishing sites hosted in the same domain as the current website.
3- The appearance of the legal website’s hostname, an IP address or a port number in the current website’s URL.
4- The history of the current website’s hosting company regarding hosting any phishing pages.
5- The history of the current website’s hosting country with respect to phishing websites.
6- The top level domain’s history, for example .biz, regarding hosting any phishing pages in the past.
7- The current site’s rank score.

In addition, NetCraft Toolbar enforces the browser to show its address bar in every window to combat some of the advances in terms of phishing attacks, in which the address bar is disabled in order to deceive the user. This technique has one advantage and one limitation.

Advantage

1- NetCraft Toolbar copes with DNS poisoning. That is if the local DNS have been altered. For example, if www.facebook.com, which is supposed to be hosted in the USA, is assigned to an IP address from Turkey, the NetCraft Toolbar will display Turkey in the “country” label and the user can identify the problem.

Limitation

1- Some phishing sites are hosted on compromised servers in which the domain names of such servers have a clean history in NetCraft Web Server Survey regarding phishing sites and these domain names have been registered on the Internet since 2001, for example. As NetCraft depends heavily on the age property of websites’ domain names, the tool will consider these fraudulent sites as trusted sites.

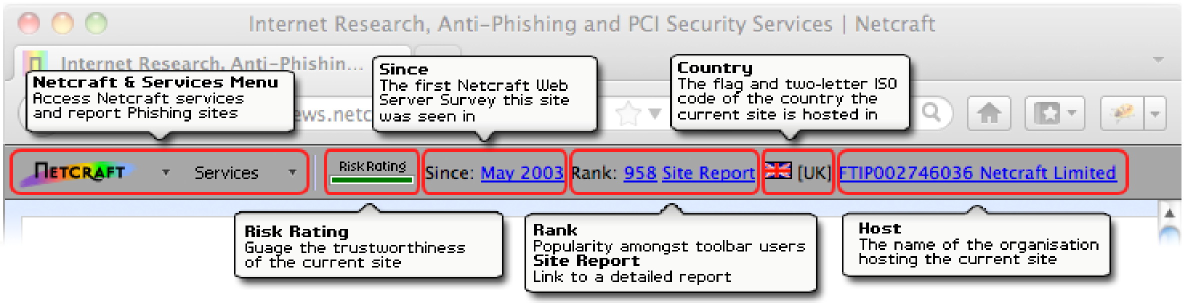

Internet Research, Anti-Phishing and PCI Security Services | Netcraft
Internet Research, Anti-Phishin...
Netcraft & Services Menu
Access Netcraft services and report Phishing sites
Since
The first Netcraft Web Server Survey this site was seen in
Country
The flag and two-letter ISO code of the country the current site is hosted in
NETCRAFT
Services
Risk Rating
Since: May 2003
Rank: 958 Site Report
[UK]
FTIP002746036 Netcraft Limited
Risk Rating
Guage the trustworthiness of the current site
Rank
Popularity amongst toolbar users
Site Report
Link to a detailed report
Host
The name of the organisation hosting the current site

Figure 2.4: NetCraft Toolbar [23]

2.2.2. Visual-clue-based method

Visual-clue-based method applies the idea of using images as a base for the solution to combating phishing attacks. This method relies on the fact that phishing attackers try to lure users by imitating visual features of target websites. This method tends to use images as authentication evidences that the server should present. Dynamic security skin [25] and Visible Watermarking [26] are two visual-clue-based techniques.

Dhamija and Tygar propose a technique called dynamic security skin [25]. In their design, a user needs to remember only one simple password throughout the whole session, and performs two image matches in order to authenticate a remote server. Basically, for the first time, the user selects one image from a list as a background to the login window. The authors call this the trust password window. This image proves to the user that the window knows the shared secret. The authors adopted the Secure Remote Password protocol (SRP) to achieve a mutual authentication between the user and the remote server, in which the two parties do not have to share a secret password. To do this, the user first chooses a password, a random salt and performs a one-way function to generate the verifier. This verifier and the salt should be sent to the server, which will store this information and consider the verifier as the user's password. To access the server, the user provides his or her username, and the server finds the corresponding verifier and salt. Then the user's browser and the server separately generate two random values and exchange them. Then, using the random values and the verifier, each party separately computes an identical session key and generates a hash value of this session key. After that, each party sends each other the hash value of this session key and the random values exchanged earlier. At this stage every party has proved to the other party that it knows the shared secret. However, the user needs to identify an authenticated web page. The authors propose the idea of automated custom security indicators in which random generated images are used. In the last stage of the authentication, the server generates the hash value of the session key. The server can use a visual hash algorithm, Random Art, which takes this hash value and generates a random mathematical formula that determines a color value for every pixel in an abstract image. Using the same hash value, the user's browser can generate the same abstract image. Then, the browser presents this image, for example as a window's border, on the trusted password window. Similarly, the server presents the same image on its webpage. The user then compares the two images on the trusted password window and the server's webpage. If there is a match, then he can trust the server's webpage. This technique has one advantage and two limitations.

Advantage

1- The proposed technique provides the server with a way to prove its identity which is easy for a user to recognize, as he or she only needs to perform two image matches, and it is hard for an attacker to spoof since the attacker has neither the verifier nor the random values [1].

Limitations

1- This technique requires the user to have some knowledge of phishing attacks and how to identify spoofed pages in order to distinguish between an authentic and a spoof webpage. As a result of the leak knowledge, more than 20% of users ignore webpage's visual clues and even professional users may be victims of visual-based attacks [27].

2- This solution is vulnerable to the visual man-in-the-middle-attack [25]. That is, an attacker may be able to create a pop up fraudulent window on the front of an authenticated window and the trust password window**.**

Topkara et al. propose another technique called visible watermarking (ViWiD) [26]. It is an integrity check technique in which the user needs to verify a watermark within the company webpage's logo to authenticate this webpage. This watermark consists of two parts: a shared secret, which the user selects at the registration stage in a secure manner, between the user and the company's sever, and the current date and time of the user's time zone determined by the IP address of the user's machine. This watermark is designed to be unique for every user in order to combat a "one size fits all" attack. The company's logo can appear to the user in two ways: after the user login into his or her account, or by using a cookie. The last choice is preferred since the user need not to enter his or her confidential data on the login webpage to avoid revealing this data on a forged webpage. The user can trust the server's webpage since its logo includes the shared secret. The process of adding the watermark to the company's logo is done on the company web server, and the user need not install any tool or store any data on his or her local machine. This technique has two advantages and two limitations.

Advantages

1- If the user prefers to use cookies to access sensitive web pages through ViWiD, the chance of attackers stealing his or her confidential data is significantly reduced.
2- Since the watermark is different for each user at a specific time, and includes a shared secret between the user and the company, it is hard to design a fraudulent webpage that displays the correct watermark for each user.

Limitations

1- This technique requires the user to be involved in the verification process.
2- The users have to be trained to expect what information should appear in the company's logo in order to distinguish between real and fake webpages.

2.2.3. Webpage-feature-based method

Another method depends on analyzing the webpage's contents to find fraud symptoms, and then warning the user of a potential phishing attack. A number of techniques adopt this method have been proposed, for example SpoofGuard [6] and a framework for the detection and measurement of phishing attacks [8].

Chou et al. proposed and implemented a technique called SpoofGuard [6] to mitigating simple phishing attacks. Typically, when a user visits a webpage, several evaluations on this webpage and a check on outgoing post data will be applied to compute a webpage's spoof index or a total spoof score (TSS). If this spoof index is greater than a threshold which has previously been specified by the user, it indicates that such a webpage is a spoof and the user will be warned. Some of these evaluations are done after downloading the webpage: URL, link, image and domain checks. In addition, some evaluations are conducted when the user interacts with such a page: password, outgoing password, referring page, outgoing post data checks. Table 2.2 summarizes these evaluation functions.

Table 2.2: A summary of SpoofGuard's evaluations functions

Check type	Function
URL	If a webpage's URL includes "@" or an IP address, then increase the spoof index
Link	If 25% of a webpage's links fail an URL check then increase the

	spoof index
Image	If an image on a webpage is in imageDataBase, then check if the two images are associated with different domains, then increase the spoof index
Domain	If a webpage's host domain is similar to a host domain in the history file or in commonly spoofed sites file, then increase the spoof index
Password	If a function of a webpage requests the user's password and this webpage does not use HTTPS, then increase the spoof index
Outgoing password	When the user enters a password on a webpage, a hash value of this password and the webpage's host domain will be compared against a database. This includes hash values of previous entered passwords and their corresponding webpage's host domain. If there is a password match with a different host domain, then increase the spoof index and warn the user
Referring page	If the user is redirected to a webpage, then check if the referring page is an email provider, then increase the spoof index
Post data	If a webpage's function requests any data, then a hash value of the data and the webpage's host domain will be compared against a database. This includes hash values of previous entered passwords and their corresponding webpage's host domain. If there is a password match with a different host domain, then run password check

This technique has three advantages and two limitations.

Advantages

1- If this method is adopted by the majority of Internet users, the phishing attackers will need to develop more sophisticated attacks [25].

2- Such a method presents high accuracy rates (90%) when it comes to identifying phishing pages [9].

3- This method provides a user with a monitoring system without requiring user involvement.

Limitations

1- As this method is developed to address simple phishing attacks, it can be fooled using sophisticated phishing attacks such as Cross-Site Scripting (XSS) [10]. That is, using a script code, an attacker can construct input forms in order to gain user confidential data.

2- SpoofGuard has a relatively high false alarm rate [9][11]. That is, identifying a number of genuine websites as fraudulent.

Garera et al. propose a technique for the detection and measurement of phishing attacks [8] which depends heavily on analyzing URLs to distinguish between benign and phishing web pages. In this solution, a logistic regression filter takes a URL as its input, and applies 18 URL feature tests to determine the webpage's nature, - whether it is benign or phishing. The authors collected most of these features from some Google infrastructures such as the White Domain Table and Google's index infrastructure. The URL feature tests can be classified into four types: page-based features, domain-based features, type-based features and word-based features. Then, they use the Weka data mining library to analyze 2,508 URLs (1,245 phishing and 1,263 non-phishing) using the logistic regression algorithm. From this experiment they obtained the coefficients of the 18 URL features (see Table 2.3). From these results, the authors found that “host obfuscated with IP” and “White Domain Table are the most useful features to identify phishing ULRs. The authors claim that their method has an accuracy rate of 97.31% with a true positive rate of 95.8% and false positive rate of 1.2%.

Table 2.3: The 18 URL features and their coefficients

Feature	**Logistic coefficient**	**Odd Ratio $_e$coefficient**
Is URL in White Domain Table?	-3.82	0.0219
Quality Score II	-1.9543	0.1417
PageRank of Host	-1.8812	0.1524
PageRank of URL	-1.2606	0.2835
PageRank in Crawl Database	-0.536	0.5851
Quality Score I	0.0443	1.0453
Number of characters after organization in host	0.2306	1.2594
Word secure presence	0.3328	1.3949
Word account presence	0.8589	2.3605
Is Page in Index?	0.8738	2.3961
Word webscr presence	0.9969	2.7099
Word login presence	1.8587	6.4155
Word ebayisapi presence	2.1659	8.7221
Word signin presence	2.5404	12.685
Word banking presence	2.6361	13.9593
Word confirm presence	2.7586	15.777
Is target organization in path but not in host?	2.9464	19.0378
Is host obfuscated with IP?	6.3933	597.8151
Constant	**-0.5881**	

This technique has one advantage and two limitations.

Advantage

1- Like SpoofGaurd, this method provides a user with a phishing detection solution without requiring user involvement.

Limitations

1- As this method tries to identify phishing pages based on heuristics texts, it could not stop Man in the Middle Attacks [12]. For example, an attacker may use a Man-in-the-Middle Phishing Kit to serve as a proxy between the user and the provider site [13].

2- For sophisticated attackers it is easy to bypass this detection method [11].

2.2.4. Information-flow-based method

Information-flow-based method tries to protect users from being victims of phish attacks by tracking their sensitive information to make sure that they provide this information on trusted websites. A user will be warned, if she is about giving away her confidential data on fake websites. One technique that follows this method is AntiPhish [42]. This technique detects phishing by examining the current webpage’s domain when a user starts to enter sensitive data.

The AntiPhish technique’s main purpose is to protect users’ confidential data. This can be done by monitoring where the users’ confidential data is been entered and informing the user in the case of a phishing attack. Typically, when a user enters confidential data in a web page's form for the first time, she may ask AntiPhish to capture this data and stores it in an encrypted form. AntiPhish uses the DES encryption algorithm to encrypt users' confidential data by a master password. AntiPhish also stores a web page's domain to be mapped with the user data. AntiPhish uses a domain rather than a web page' address because some websites are hosted in more than one server. However, if AntiPhish uses the address, false attack detection may be triggered. The user needs to provide the master password the next time in order to automatically fill in the previous web page's form. To monitor the users' confidential data, AntiPhish examines text field elements of any form in a web page and interrupts any user event. If the user interacts with a text element, AntiPhish will compare the element value against a list of previous stored user's confidential data. If it finds a match, domains comparison will started. If there is no match, AntiPhish will consider the current webpage as phishing. AntiPhish runs same test if the user generates evens on test elements: press a key, load new page, click or focus. JavaScript gives an attacker the ability of accessing form's text elements before a user submits inputs. To combat this problem, AntiPhish deactivates JavaScript if the focus is on a text element and reactivates it when the focus is lost. AntiPhish has two advantages and two limitations.

Advantages

1- AntiPhish may reduce the false positive rate by using the web page's domain rather than the address in mapping user's confidential data.
2- AntiPhish provides an effective method to combat event-based JavaScript attacks and permits legitimate event-based JavaScript functions in the same time.

Limitations

1- The user needs to inform AntiPhish to capture her confidential data.
2- Key-press JavaScript functions are not allowed by AntiPhish, since it prevents such functions.

2.3. The best way forward

In order to mitigating the identity theft problem in the future it is important to address new security threats. These threats may result from vulnerabilities in new development technologies, for example HTML5 and URL shortening, or from new phishing techniques. These techniques are: Cross Site Scripting or (XSS), HTML attachment, Tabnabbing. HTML5 technology provides web developers with advanced web application features. However, these HTML5's new features allow the attackers to exploit new security threats [28]. Unfortunately, there is no solution have been proposed in literature or industry to address these threats. For this reason, the focus of this work will be on these security threats and on proposing a framework to mitigate the expected resulting problems.

2.4. Summary

In this chapter a verity of proposed anti-phishing solutions both in literature and in industry have been discussed and critically analyzed in order to show their advantages and limitations. Most of these solutions are leading in the anti-phishing field. From the discussion, it was clear that all of these solutions can stop some phishing attacks but not all attacks. In addition, attackers have developed new phishing techniques, for example Tabnabbing. This project tries to address some of these techniques' threats.

Chapter 3. The proposed solution design

3.1. Solution methodology overview

SpoofGuard is a well-known solution in the literature. This solution proves its ability to combat simple phishing attacks [9]. Due to this reason, the functions of SpoofGuard will be adopted in the proposed solution. Some of these functions will be modified to provide better resistance against simple phishing attacks. In addition to these functions, a variety of proposed functions will be added to address the concern of sophisticated and new phishing attacks. A combination of these functions will be used to form the proposed solution, SpoofGuard++.

To achieve this solution three main phases are involved: designing the proposed system, implementing this design and evaluating the resulting system. These three phases will be performed in a sequential manner. That is design then implementation and finally evaluation.
Phase 1: In order to obtain good design system requirements a collection step is needed. These requirements can be collected by performing two activities: previous work analysis and new phishing threats investigation. The previous work analysis, or literature review activity, is important to gain thorough understanding of the phishing problem. An investigation into the activity of new phishing threats is needed to obtain a good knowledge of the recent phishing attacks that need to be addressed. After obtaining the system requirements, the design phase can commence. The main purpose of the design phase is to convert the system requirements into a conceptual form. This conceptual form can be implemented in a programming language.
Phase 2: Upon finishing the previous phase, the implementation phase can be started. Essentially, the architecture diagrams of the initial design will be converted into a working programme.
Phase 3: After implementing the system design successfully, evaluation activities can be performed. These activities are needed to ensure that the resulting system achieves the system requirements and works in the desired way. In addition, if any problems appear in this phase, the design and implementation phases will be repeated after addressing such problems.

3.2. Requirment specifications

In light of the literature review and new phishing threats investigation activities, the main requirements of the proposed system are: functionality, performance and reducing false detection rate requirements.

Functionality requirements: The proposed system should be able to detect simple and sophisticated phishing attacks through its functions. These functions should include:

1- Enhanced functions of SpoofGuard: Some of SpoofGuard's functions can be fooled by the attackers [6]. These functions are: URL, image and link checks.

2- HTML5 threats detection functions: Some of HTML5's new features allow the attackers to exploit new security threats [28]. These threats are: cross-document messaging, local storage, attribute abuse, inline multimedia and SVG and input validation. The proposed solution should provide effective detection functions to address these threats.

3- Cross Site Scripting or (XSS) detection function: XSS attacks occur when a user supplies malicious inputs to a web application [29]. Typically attackers inject JavaScript into bad programmed web applications using these inputs in order to disclose users' confidential data. The proposed solution should provide an effective detection function to address this threat.

4- URL shortening threat detection function: URL shortening service is developed to avoid using long URLs. This service enables users to use services, such as Tiwter and Identi.ca, in which the usage of

URLs is limited to 140 characters per message. However, this service provides attackers and spammers with an ability to bypass protection techniques, such as URLs check [30]. The attackers are then able to redirect unsuspecting users to malicious sites in order to gain users' confidential data. The proposed solution should provide an effective detection function to address this threat.

5- HTML attachment attack detection function: Attackers have found a new way to bypass blacklist-based anti-phishing tools in modern browsers. This is by using HTML attachments rather than URLs in their fake emails [31]. The proposed solution should provide an effective detection function to address this threat.

6- Tabnabbing attack detection function: Tabnabbing is a new phishing attack in which the contents of a webpage can be changed after the user has left it open for a while [32]. This action can be performed using a simple JavaScript code. The proposed solution should provide an effective detection function to address this threat.

Performance requirement: The proposed solution, SpoofGuard++, is an Internet Explorer 9 extension. The functionalities of SpoofGuard++ should not, however, degrade the performance of Internet Explorer 9.

Reduce false detection rate requirement: In order to provide Internet users with a useful phishing monitoring system, Spoofguard++ should produce false attack detection as little as possible.

3.3 System architecture design

SpoofGuard++ is an Internet Explorer 9 extension or a Browser Helper Object that uses the same memory of the explorer. It can access the explorer's events and data in order to detect any malicious activity or content in a visited web page. To give a general idea about how SpoofGuard++ works, an overview if its architecture and architecture's components are discussed below.

3.3.1 Architecture overview

SpoofGuard++ consists of two main components: DOM Tree Extractor and Phishing Assessment Manager. When the explorer finishes loading a web page, the DOM Tree Extractor can be used to get the DOM Tree of this page. The outputs of the DOM Tree Extractor are used as inputs of the Phishing Assessment Manager. This component examines the web page's tags and data to find any phishing attack's feature and gives an indication about this page whether it is a phish or a normal page. Figure 3.1 shows the architecture of SpoofGuard++.

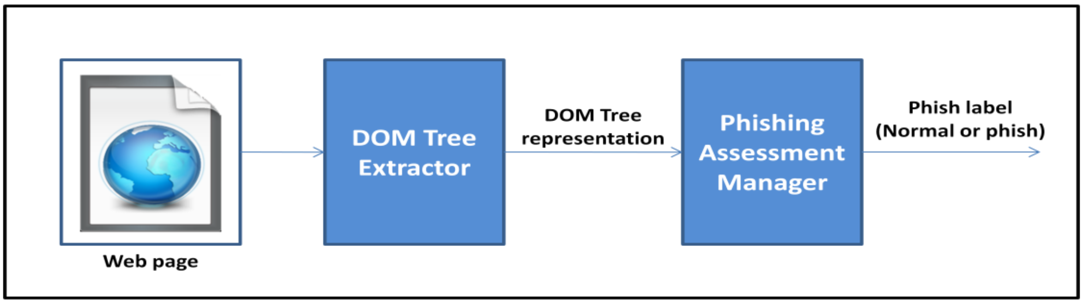


Figure 3.1: SpoofGuard++ architecture

3.3.2 Architecture's components

The DOM Tree Extractor and Phishing Assessment Manager components involved a number of internal functions.

DOM Tree Extractor: Internet Explorer 9 represents web pages as DOM Trees. These trees' contents are filtered to organize these threes' data. DOM Tree Extractor involves images, links, scripts, styles and filters and advertisements remover. The filters' functionalities are proposed by E. Kaasinen, et al [39] and advertisements remover function is proposed by S. Gupta et al. [40]. The advertisements remover function is used to reduce the processing time in order to get better performance. The results of these functions serve as parameters of Phishing Assessment Manager' functions. **Phishing Assessment Manager**: This component involves domains, password, outgoing password, referring page and outgoing post data check functions, which are adopted from the work of N.Chou et al. [3]. In addition, this component involves the proposed functions: URLs, links, images, HTML5 threats, XSS, URL shortening, Tabnabbing attack checks.

3.4. Implementation methodologies

In order to translate the system design into a working system, implementation platform and programming language selection steps are needed.

3.4.1. Implementation platform

As the majority of users surf the Internet using Microsoft Internet Explorer [33], it is worth producing an anti-phishing tool that will work on this platform. The latest version of Internet Explorer, IE9, is the selected platform for this project. To integrate the proposed solution into IE9, the development of a Browser Help Object (BHO) in needed. BHO has two advantages: its flexibility and continuity [33]. Native Windows codes can be involved within a BHO, since a BHO is an independent Windows thread. Thus, it provides developers with direct ways to create process, files, and network connections, in addition to the ability to invoke an existing code. BHO also benefits from Internet Explorer support. That is, a BHO can work perfectly on the current Internet Explorer version and also the later versions.

3.4.2. Programming language

Microsoft has proposed the .NET framework in which any high level code (for example C#, VB.NET, C++) is compiled into a Common Intermediate Language (CIL) [33]. The resulting code is called a managed code. This code is executed inside a sandbox component. This component prevents the code inside it from calling any code outside the .NET framework. This technology provides a safe implementing environment. For this reason the .NET framework and C# programming language have been selected to implement SpoofGuard++.

3.5. Evaluation methodologies

After implementing SpoofGuard++, it is necessary to evaluate it against the system's requirements. Three main evaluation criteria can be used to test SpoofGuard++: detect phishing attacks, reduce false detection rate and determine the impact on Internet Explorer 9.

3.5.1 Detect phishing attacks

As the main aim of SpoofGuard++ solution is to identify and detect phishing attack attempts, this solution should be tested against real and new phishing attacks. The number of correct detections should then be calculated.

3.5.1.1 Test it against phishing attacks

Fresh and real phishing emails and websites are needed to test SpoofGuard++. Anti-Phishing-working-Group (APWG) and PhishTank provide phishing feeds containing recent phishing emails and websites. These emails and websites can be used for testing purposes.

3.5.1.2 Count how many attacks the tool successfully detected

The effectiveness of SpoofGuard++ can be initially determined by counting the number of correct detections. During the process of testing SpoofGuard++, the calculating processes can be performed manually.

3.5.2 Evaluate false positive rate

Any anti-phishing solutions may lose the users' trust when the solution identifies a large number of normal sites as phishing sites. For this reason, it is important to count the number of false detections identified by SpoofGuard++ during the test.

3.5.2.1 Count tool's false detections

During the process of testing SpoofGuard++, the calculating processes can be performed manually.

3.5.3 Impacts on Internet Explorer

One important factor that can make an anti-phishing plug-in useless is overhead on Internet Explorer. Therefore, SpoofGuard++ performance should be monitored and reported. This can be done by surfing certain websites with and without SpoofGuard++ and measure the execution time in both cases. A number of most visited web pages, provided by NetCraft [34], can be used in this experiment.

3.5.3.1 Open specific sites on the explorer without the tool

Without a SpoofGuard++, three time records should be observed for each webpage: the time before surfing the webpage, the time after downloading the webpage and the duration between these two times.

3.5.3.2 Open the same sites on the explorer with the tool

Now with a SpoofGuard++, three time records should be observed for each webpage: the time before surfing the webpage, the time after downloading the webpage and the duration between these two times.

3.5.3.3 Measure the excution time in both cases

After conducting the previous experiments, the performance of SpoofGuard++ can be determined. To perform this, for each webpage a comparison between surfing time with and without SpoofGuard++ is conducted.

3.6. Work organization

The project organization step is an important factor in producing a solid project. Specifying project deliverables and project plan are needed in order to organise the project.

3.6.1. Project deliverables and outcomes

The expected outcomes of this project are:
1- A full implementation of an Anti-Spoofing tool (SpoofGuard++).
2- An evaluation of the effectiveness of SpoofGuard++ against its requirements.
3- A recommendation list of future work.
4- A documentation of all works of developing the proposed solution.

3.6.2. Project plan

Planning of this project is needed in order to achieving better outcomes for this project with respect to the limited time. To gain a thorough understanding of the problem and the previous work in the anti-phishing domain, an intensive reading of literature is needed. This required about 35 days. After that 4 days are allocated to conduct an investigation on new and sophisticated phishing techniques. This activity and the literature reading activity lead to gathering the proposed system requirements. More 5 days are dedicated for designing the proposed solution's architecture and main functions. The previous activities are needed to achieve the first phase of the project components as describe in section 3.1 solution methodology overview.

After successfully fishing the first phase, the proposed solution is ready to be implemented. In this phase the initial architecture and components of SpoofGuard++ will be traslated into a working program as stated in both section 3.3.1 architecture overview nad 3.3.2 architecture components. The time allocated for these activities is about 30 days. Upon implementing SpoofGuard++, evaluation activities are conducted to test its functionality. The expected time to perform the evaluation activities is about 6 days. For more details about evaluation methodology, the reader is advised to read section 3.5 evaluation methodologies.

3.7. Summary

SpoofGuard++ solution development should be processed through three main phases: designing the proposed system, implementing this design and evaluating the resulting system. The design phase involves the collection of the system requirements and system architecture design. In the implementing phase, the initial architecture design is converted into a working program. Finally, the evaluation phase involves activities that test the resulting system against its requirements.

4. References

[1] H. Huang, J. Tan, L. Liu, "Countermeasure Techniques for Deceptive Phishing Attack", International Conference on New Trends in Information and Service Science, 2009, pp. 636-641.

[2] Anti-Phishing Working Group, **"**http://www.antiphishing.org/**"**, 2011, retrieved on 20/03/2011.

[3] N. Chou, R. Ledesma, and Y. Teraguchi, et al, "Client-side Defense against Web-based Identify Theft", In: Proc. of 11th Annual Network and Distributed System Security Symposium, 2004, pp.1-16.

[4] Microsoft, "Sender ID Home Page", **"**http://www.microsoft.com/mscorp/safety/tehnologies/senderid/default.aspx**"**, 2009, retrieved on 23/03/2011.

[5] Yahoo, "Yahoo! AntiSpam Resource Center", **"**http://antispam.yahoo.com/domainkeys**"**, 2009, retrieved on 23/03/2011.

[6] Cisco Corporation, "IronPort Email Authentication", 2008, pp. 1-14.

[7] R. Dhamija, and J. D. Tygar, "The Battle against phishing: Dynamic Security Skins", In: Proc. of ACM Symposium on Usable Security and Privacy, 2005, pp.77-88.

[8] S. Garera, N. Provos, and M. Chew, et al, "A Framework for Detection and Measurement of Phishing Attacks", In: Proc. of the 5th ACM Workshop on Recurring Malcode, 2007, pp.1-8.

[9] Y. Zhang, S. Egelman, L. Cranor, and J. Hong, "Phinding Phish: Evaluating Anti-Phishing Tools", 2010, pp. 1-16.

[10] H. Shahriar and M. Zulkernine, "PhishTester: Automatic Testing of Phishing Attacks", Fourth International Conference on Secure Software Integration and Reliability Improvement, 2010, pp. 198-207.

[11] S. Bin, W. Qiaoyan, L. Xiaoying, "A DNS based Anti-Phishing Approach", Second International Conference on Networks Security, Wireless Communications and Trusted Computing, 2010, pp. 262-265.

[12] Y. Joshi, D. Das, S. Saha, "Mitigating Man in the Middle Attack over Secure Sockets Layer", IEEE, 2009, pp. 1-5.

[13] J. Milletary, "Technical Trends in Phishing Attacks", US-CERT, 2006, pp. 1-17.

[14] A. Bergholz, J.-H. Chang, G. Paaß, F. Reichartz, and S. Strobel, "Improved phishing detection using model-based features". In Proceedings of the Conference on Email and Anti-Spam (CEAS), 2008, pp. 1-10.

[15] G. V. Cormack and R. N. Horspool, "Data compression using dynamic markov modelling", The Computer Journal, 30(6), 1987, pp.541–550.

[16] I. Fette, N. Sadeh, and A. Tomasic. "Learning to detect phishing emails", In Proceedings of the International World Wide Web Conference (WWW), 2007, pp. 649–656.

[17] Gregory L. Wittel and S. Felix Wu, "On Attacking Statistical Spam Filters", first conference on E-mail and Anti-spam, 2004, pp. 1-7.

[18] M. Chandrasekaran, K. Narayanan and S. Upadhyaya, "Phishing E-mail detection based on structural properties", NYS Cyber Security Conference, 2006, pp. 1-7.

[19] R. Shah, J. Trevathan, W. Read and H. Ghodosi, "A Proactive Approach to Preventing Phishing Attacks Using a Pshark", Sixth International Conference on Information Technology, 2009, pp. 1-7.

[20] BrandProtect International, "BrandProtect's Phishing takedown process", "http://www.brandprotect.com/phishing-take-down-process.html", 2010, retrieved on 25/04/2011.

[21] T. Moore and R. Clayton, "Examining the impact of website take-down on phishing", Anti-Phishing Working Group eCrime Researcher's Summit (APWG eCrime), 2007, pp. 1-13.

[22] Microsoft, SmartScreen Filter, "http://windows.microsoft.com/en-US/internet-explorer/products/ie-9/features/smartscreen-filter", 2011, retrieved on 24/03/2011.

[23] NetCraft, Netcraft Tolbar, "http://toolbar.netcraft.com", 2011, retrieved on 21/03/2011.

[24] Microsoft, SmartScreen Filter and Resulting Internet Communication in Windows 7 and Windows Server 2008 R2, "http://msdn.microsoft.com/en-us/library/ee126149(v=ws.10).aspx", 2009, retrieved on 24/03/2011.

[25] R. Dhamija, and J. D. Tygar, "The Battle against phishing: Dynamic Security Skins", In: Proc. of ACM Symposium on Usable Security and Privacy, 2005, pp.77-88.

[26] M. Topkara, A. Kamra, and M. J. Atallah, et al, "ViWiD: Visible Watermarking Based Defense against Phishing", Lecture Notes in Computer Science, Vol.3710, 2005, pp.470-483.

[27] R. Dhamija, J. D. Tygar, and M. Hearst, "Why Phishing Works? ", In: Proc. of the SIGCHI conference on Human Factors in computing systems, 2006, pp.581-590.

[28] A. Weiss, "Top 5 Security Threats in HTML5", "http://www.esecurityplanet.com/features/article.php/3916381/Top-5-Security-Threats-in-HTML5.htm", 2010, retrieved on 14/03/2011.

[29] Cgisecurity, "The Cross-Site Scripting (XSS) FAQ", "http://www.cgisecurity.com/xss-faq.html", 2002, retrieved on 13/03/2011.

[30] M. Kassner, "URL shortening: Yet another security risk", "http://www.techrepublic.com/blog/security/url-shortening-yet-another-security-risk/1044", 2009, retrieved on 05/05/2011.

[31] E. Mills, "Phishers use HTML attachments to evade browser blacklists", "http://news.cnet.com/8301-27080_3-20043960-245.html", 2011, retrieved on 06/05/2011.

[32] A. Raskin, "Tabnabbing: A New Type of Phishing Attack", "http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/**",** 2010, retrieved on 03/05/2011.

[33] T. Raffetseder, E. Kirda, and C. Kruegel, "Building Anti-Phishing Browser Plug-Ins: An Experience Report", ICSE Workshop on Software Engineering for Secure Systems (SESS), IEEE Computer Society Press, 2007, pp. 1-7.

[34] NetCraft, "Most Visited Web sites", "http://toolbar.netcraft.com/stats/topsites", 2011, retrieved on 06/05/2011.

[35] AgainstPhishing, **"**The Dangers of a Phishing Attack**"**, **"**http://www.againstphishing.com/dangers-of-phishing.html**"**, 2010, retrieved on 01/05/2011.

[36] Internet Technologies Workshop: Tel-Aviv University, "Current Anti Phishing Methods", "http://tau-itw.wikidot.com/deleted:saphe-current-anti-phishing-methods", 2009, retrieved on 14/02/2011.

[37] Gartner, "Number of Phishing Attacks on U.S. Consumers Increased 40 Percent in 2008", "http://www.gartner.com/it/page.jsp?id=936913", 2009, retrieved on Sunday 13/02/2011.

[38] L. Phifer: E-Security planet, "top ten phishing facts", "http://www.esecurityplanet.com/views/article.php/3875866/Top-Ten-Phishing-Facts.htm", 2010, retrieved on 13/02/2011.

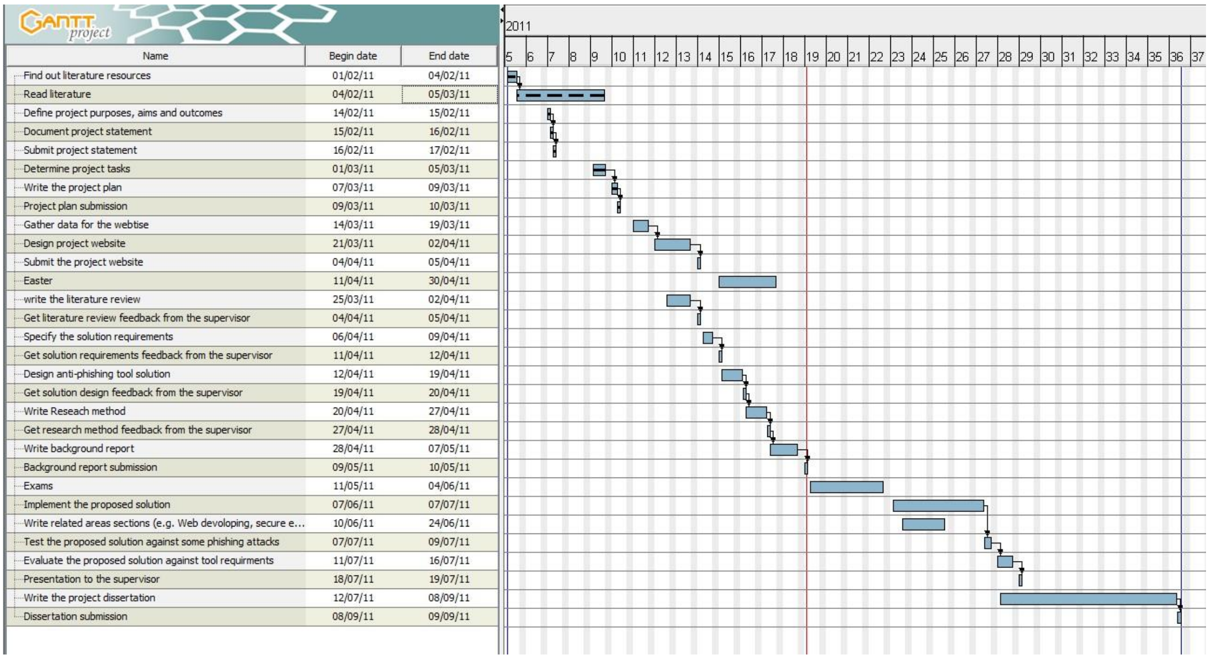
[39] E. Kaasinen, M. Aaltonen, J. Kolari, S. Melakoski and T. Laakko, "Two approaches to bringing Internet services to WAP devices", 2000.

[40] S. Gupta, G. Kaiser, D. Neistadt and P. Grimm, "DOM-based Content Extraction of HTML Documents", 2003, pp. 1-10.

[41] C. Herley, "So Long, And No Thanks for the Externalities: The Rational Rejection of Security Advice by Users", Association for Computing Machinery, Inc., 2009, pp. 1-12.

[42] E. Kirda, and C. Kruegel, "Protecting Users against Phishing Attacks with AntiPhish", In: Proc. of the 29th Annual International Computer Software and Applications Conference, 2005, pp.521-534.

Appendix A : The Project Gantt chart



Name	Begin date	End date
Find out literature resources	01/02/11	04/02/11
Read literature	04/02/11	05/03/11
Define project purposes, aims and outcomes	14/02/11	15/02/11
Document project statement	15/02/11	16/02/11
Submit project statement	16/02/11	17/02/11
Determine project tasks	01/03/11	05/03/11
Write the project plan	07/03/11	09/03/11
Project plan submission	09/03/11	10/03/11
Gather data for the webtise	14/03/11	19/03/11
Design project website	21/03/11	02/04/11
Submit the project website	04/04/11	05/04/11
Easter	11/04/11	30/04/11
write the literature review	25/03/11	02/04/11
Get literature review feedback from the supervisor	04/04/11	05/04/11
Specify the solution requirements	06/04/11	09/04/11
Get solution requirements feedback from the supervisor	11/04/11	12/04/11
Design anti-phishing tool solution	12/04/11	19/04/11
Get solution design feedback from the supervisor	19/04/11	20/04/11
Write Reseach method	20/04/11	27/04/11
Get research method feedback from the supervisor	27/04/11	28/04/11
Write background report	28/04/11	07/05/11
Background report submission	09/05/11	10/05/11
Exams	11/05/11	04/06/11
Implement the proposed solution	07/06/11	07/07/11
Write related areas sections (e.g. Web devoloping, secure e…	10/06/11	24/06/11
Test the proposed solution against some phishing attacks	07/07/11	09/07/11
Evaluate the proposed solution against tool requirments	11/07/11	16/07/11
Presentation to the supervisor	18/07/11	19/07/11
Write the project dissertation	12/07/11	08/09/11
Dissertation submission	08/09/11	09/09/11