An Anti-Phishing Tool: PhishProof

A dissertation submitted to the University of Manchester for the degree of Master of Science in the Faculty of Engineering and Physical Sciences

2012

TAIMOOR ZAHID

School of Computer Science

Table of Contents

The Final Word Count:

Body of the dissertation: 17,001

Total dissertation: 19,262

List of Figures

List of Tables

List of Equations

List of Abbreviations

I. CSS – Cascading Style Sheet
II. SVM – Support Vector Machine
III. HTML – Hyper Text Markup Language
IV. DMC – Dynamic Markov Chain
V. APWG – Anti-Phishing Work Group
VI. URL – uniform resource locator
VII. DOM – Document object Model
VIII. SSL – Secure Sockets Layer
IX. GUI – Graphical User Interface
X. API – Application Programming Interface
XI. HTTP – Hypertext Transfer Protocol
XII. TP – True Positive
XIII. TN – True Negative
XIV. FP – False Positive
XV. FN – False Negative
XVI. MBS – Manchester Business School

Abstract

Phishing is the art of luring users to a fraudulent page by masquerading as a trustworthy organisation and acquiring their personal information such as credit card number, social security number, password or any other information which can be used to gain benefit. Despite having several anti-phishing tools the number of victims has increased dramatically over last few years as internet users ignore warning alerts and most of the solutions available rely on user input. The proposed solution attempts to help users differentiate between phishing and legitimate web pages with minimal effort.

This project includes a detailed literature review of existing anti-phishing techniques. In addition, advantages and limitations of each technique have been discussed to understand where existing solutions are lacking. On the basis of this literature research, an anti-phishing solution, PhishProof, has been designed, implemented and evaluated. PhishProof uses a combination of blacklist and web page content analysis method to provide 3-level protection against phishing. Level-1 uses blacklist method, level-2 uses a combination of both methods and level-3 analyses 6 features of a web page to compute risk rating which determines whether a page is phishing or legitimate. Each level is only initiated if a web page survives the previous level without being flagged. Voice and message alerts are used to get user's attention if a phishing page is encountered. A Firefox extension is developed using JavaScript, XUL and CSS. A website and backend administrator panel are also developed to assist the Firefox extension. Evaluation of PhishProof shows it has an accuracy of 80% and is easy to use for both expert and naïve users. In addition, PhishProof has negligible effect on Firefox's performance.

The future work for proposed solution includes adding features to counter malware and JavaScript based attacks, making extension compatible with tabbed browsing and extending PhishProof to other commonly used browsers.

Declaration

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual Property Statement

I. The author of this dissertation (including any appendices and/or schedules to this dissertation) owns any copyright in it (the "Copyright") and s/he has given The University of Manchester the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

II. Copies of this dissertation, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has entered into. This page must form part of any such copies made.

III. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the dissertation, for example graphs and tables ("Reproductions"), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

IV. Further information on the conditions under which disclosure, publication and commercialization of this dissertation, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/display.aspx?DocID=487), in any relevant Dissertation restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Guidance for the Presentation of Dissertations.

Acknowledgement

I would specially like to thank my project supervisor, Dr Ning Zhang, for her guidance and support throughout the project.

I would like to thank Noman Ahmed and Ihsan Bhatti for helping me deploy PhishProof website and PhishProof backend administrator panel on to the server. I would also like to thank Mazen, Shradha, Tushar, Rahul, Vartika, Atul and Rincy for participating in user acceptance testing activity. Without their help and cooperation the evaluation of the software would be impossible.

Finally, I would like to express my gratitude to my family and friends and acknowledge them for their indispensable support to make this dissertation a success.

Chapter 1 Introduction

Phishing is a well known social engineering technique carried over emails or other electronic communication with the primary goal of acquiring sensitive user information. Attackers masquerade as trustworthy financial organisations and fraudulently acquire passwords, bank account details or other personal information [25]. The first phishing attack in the 90's on AOL set the trend for attackers to target financial institutions.

1.1 Project Context

The best way to deal with any problem is to understand the problem and its consequences. Similarly in order to better understand phishing and its severity, it is important to know how a phishing attack is done, its consequences and the counter measures.

1.1.1 How a phishing attack is done

A phishing attack involves an attacker, a victim, a phishing website and a target website, as shown in figure1.1 According to Huang et al., [27] a phishing attack involves 5 steps:

1. Attacker sets up a phishing website.
2. Attacker sends the link of the phishing website to the victim via email or instant message.
3. Victim follows the link and enters personal details on the phishing website thinking it is a legitimate website.
4. Attacker retrieves user's personal details from the phishing website.
5. The attacker uses details of the user on the respective legitimate website.

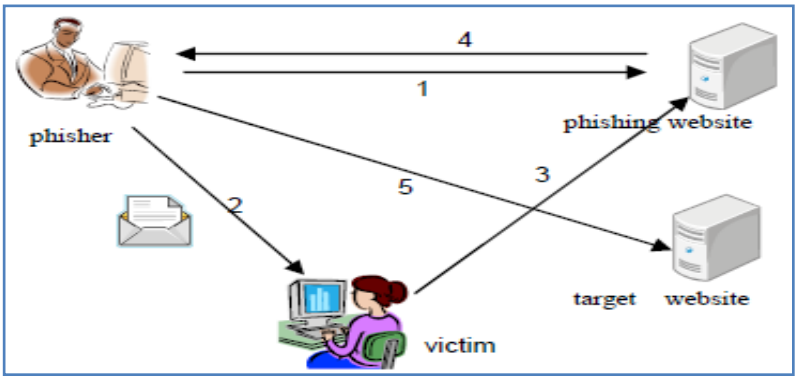


Figure 1.1: Steps involved in a phishing attack [27]

1.1.2 Consequences of phishing attacks

Majority of the phishing attacks have the intention of getting financial benefit and are aimed at getting users' personal details related to financial institutes like banks. From the very first reported phishing attack in the 90's the trend has been set for attackers to target financial institutions. Millions of lives are affected as people lose their money to such scams. An individual can also face legal issues if personal information gathered through phishing is used to violate the law.

The attacks are not just limited to financial institutions as shown in figure1.2 below. 392 brands were targeted by attackers in February and March 2012, which is a new all time high [28].

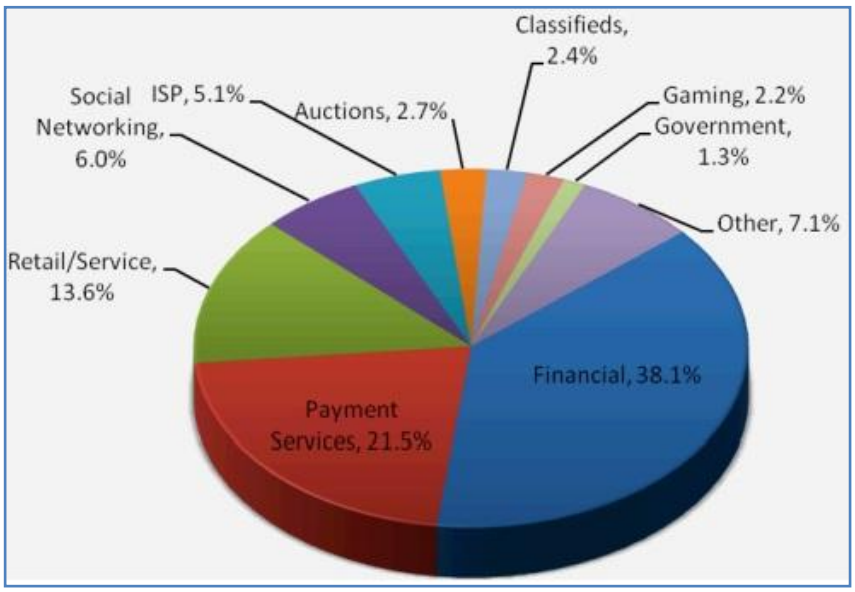


Figure 1.2: Most targeted industry sectors [28].

Apart from its economic affect, phishing also damages an individual's confidence and level of trust. People become reluctant to use online facilities available which affect genuine and legitimate business.

1.1.3 How to counter phishing attacks

In order to counter phishing many client side and server side solutions have been proposed and implemented. Anti-phishing solutions can be divided into 4 basic categories: blacklisting, domain binding, symptom based prevention and content filtering [25]. Besides technical solutions, educating users about phishing also helps in reducing the phishing risk and compliments server and client side solutions. Server and client side solutions are discussed in detail in chapter 2.

1.2 Research Motivation and Challenges

Motivations and challenges for this project are as follows:

1. Current anti-phishing solutions are not effective enough to stop modern and sophisticated phishing attacks. Blacklist based solutions are only effective if the list is updated regularly [7].
2. 56,859 unique phishing sites were detected by anti-phishing work group (APWG) in February 2012, which is a new all time high [28].
3. Phishing attacks have a success rate of over 70% on social networks [30].
4. Each year, in US alone, $350 million to $1 billion is reported as loss due to phishing [26].
5. Most of the current anti-phishing solutions require some form of input from the users who tend to ignore security warnings because of the extra effort required [29].

1.3 Aims and Objectives

The aims of this project are as follows:

1. To design and develop an anti-phishing solution that will help users differentiate between a legitimate and phishing website with minimal effort.
2. To make the design flexible enough to accommodate additions and changes in future.
3. To design and develop a platform for users to interact with the system moderator and give their valuable feedback to improve the system and also report phishing incidents.
4. To design a backend panel for system administrator to update whitelist & blacklist, and also view user reports and feedback.

In order to achieve these aims, the following tasks and objectives need to be performed:

1. Examine existing anti-phishing solutions and get a deeper understanding of how these solutions work.
2. Investigate phishing techniques and understand what the limitations of existing solutions are.
3. Collect the system requirement for the proposed solution.
4. Design the system architecture.

5. Implement the designed architecture.
6. Evaluate the system developed.

1.4 Project Scope

To achieve the project objectives, the project scope is specified as follows:

1. Development of a Firefox (version 12 and later) extension (PhishProof Toolbar).
2. Development of a website (www.phishproof.com) and backend administrator panel for the PhishProof toolbar (www.phishproof.com/admin).
3. Mozilla application framework will be used to implement the PhishProof toolbar using XUL (for toolbar interface) and JavaScript (for toolbar functionality).
4. PhishProof website will be implemented using HTML and PHP.
5. PhishProof website, backend administrator panel and database will be hosted on a shared server.
6. PhishProof is intended to assist users in differentiating between a phishing and legitimate website with minimal effort.

1.5 Dissertation Structure

The remainder of this report is structured as follows: In chapter-2 previous work done in the field of anti-phishing is discussed in detail along with advantages and limitations of each solution. Chapter-3 discusses system requirements, and explains the design of proposed solution and phishing checks to meet these requirements. Chapter-4 presents implementation of proposed solution along with the framework and languages used for implementation. Problems faced during implementation are also mentioned in this chapter. Chapter-5 describes the tests conducted to evaluate proposed solution and also mentions the limitations of system developed. Finally, the project is concluded by providing a summary of this project and highlighting what it has achieved in chapter-6. Suggestions for future work have also been mentioned in this chapter which would help reduce limitations of the proposed solution.

Chapter 2 Literature Review (countermeasures)

Phishing has been an area of interest for both scientist and industries over the past few years. Phishing solutions can be divided into two broader categories: server side and client side. Where client side phishing solutions have been of interest to big software companies the server side solutions have been an area of research focus.

2.1 Chapter Introduction

This chapter provides a detailed study of existing anti-phishing solutions. Both client side and server side solutions are critically analysed and advantages and limitations of each solution are discussed.

The chapter is structured as follows: section 2.2 provides detailed study and critical analysis of existing server side solutions. In section 2.3 existing client side solutions, their advantages and limitations are discussed. Section 2.4 describes what is learned from conducting a detailed literature review and section 2.5 provides a summary of this chapter.

2.2 Server side phishing solutions

Server side phishing solutions involve installing a program or configuring the server to prevent users from being victims to phishing attacks. These programs filter incoming emails and check websites users are trying to connect to etc. Server side solutions mainly use email content analysis or notice and take down methods.

2.2.1 Email Content Analysis

The mechanism of email content analysis method is to analyse incoming emails and filter them on the basis of a set of features. When an email arrives the application installed on the server analyses the content of the email and decides whether the email is ham (legitimate), spam or phishing.

There is a difference between spam and phishing email classification. Spam emails are intended only for informing users about some product, whereas in phishing emails there is a certain level of interaction with the receiver. Phishing emails are more harmful and may contain malicious links, deceptive forms etc through which attacker can gain

access to personal details of users. Table 2.1 lists the differences between spam and phishing emails.

Table 2.1: Differences between phishing and spam emails.

Spam Emails	Phishing Emails
intended to inform user about a product or promotion	intended to gain access to personal information of the user
less or no interaction with user	more complex interaction with user
no descriptive forms	contains descriptive forms to get user information
General audience – sent to as many as possible	Targeted audience – sent to a carefully chosen list
may contain deliberate typos to bypass spam filters	no deliberate typos in phishing emails
contains links to redirect users to product or promotion website	contains links to redirect users to fake or phished website

Content based email filtering is done on the basis of a set of features which can be categorised as; structural, Link, Element, Spam filter and Word List [1]. Machine learning techniques are used to extract relevant features by capturing the content and structural properties of a number of illegitimate emails, and data mining techniques are used to find hidden patterns within these phishing emails. There has been a lot of research in the past on content based phishing filters; some of which are discussed below.

Chadrasekaran et al., [2] proposed a technique which used structural properties of phishing emails to segregate legitimate emails from fake ones. 25 features comprising of style markers and structural attributes were used, as shown in Table 2.2. A total of 200 emails were tested; which included 100 phishing and 100 legitimate emails, with simulated annealing as feature selection algorithm. According to the relevance between features, information gain was used by author to rank the features. Support Vector Machine (SVM) classifier was used to classify phishing emails which yielded a detection rate of 95% with a very low false positive rate.

Table 2.2: marker and structural attributes extracted from email document. Total 23 style marker features and 2 structural attributes are used [2].

Category	Feature
Style Marker	Total Number of words
	Total Number of characters
	Vocabulary richness
	Function word frequency distribution (18 features) [Table 2.3].
	Total number of function words
Structural	Structure of the email subject line
	Structure of the greeting provided in email body

Table 2.3: List of 18 functional words used in experiment [2].

Keywords	Account	Log
	Access	Minutes
	Bank	Password
	Credit	Recently
	Click	Risk
	Identity	Social
	Inconvenience	Security
	Information	Service
	Limited	Suspended

Advantages:

- ✓ Emails are classified before they reach the user's inbox which reduces human exposure [2].
- ✓ Automated action - No human input required [2].

Limitations:

- × Experiment data set is not large enough to draw a broader conclusion [2].
- × Effectiveness of classification depends on the choice of features and keywords. If words outside Table 2.3 are used, the attacker will be able to bypass the filter.
- × This technique only focuses on email based attacks and won't be able to help users against other attacks [2].

Fette et al., [5] proposed a machine learning based approach PILFER. Random forest is used as a classifier to classify emails as either phishing emails or legitimate emails. A total of 10 features are used: IP based URLs, age of linked-to domain names, Non-matching URLs, "Here" links to non-modal domain, HTML emails, Number of links, Number of domains, Number of dots, JavaScript and spam filter output. 9 out of 10

features can be extracted from emails directly while WHOIS query can be used to get the "age of linked-to domain" feature. A data set with 713 legitimate emails and 860 phishing emails is used to obtain results.

Advantages:

- ✓ PILFER is flexible and can adapt if the nature of phishing attacks changes. New features can be added if they become more important [5].
- ✓ PILFER performs better than a general spam filter [5].

Limitations:

- × Less number of features used [5].

Another model based machine learning technique proposed by Bergholz et al., [3] uses 27 basic features (which can be extracted directly from the email) as input to classifiers, and 2 advanced features which can be viewed as classifiers themselves because they are based on models. The basic features can be classified in 5 categories: structural, link, element, spam filter and word list. The complete list of features is provided in Table 2.4. The 2 advanced features proposed by the author are Dynamic Markov Chain and latent Topic Model features. SVM classifier model is used to classify emails as phishing or legitimate. The 2 inputs to classifier model are phishing email's features (training set) and new email's features (testing set). Figure 2.1 shows a general view of the model machine learning technique proposed.

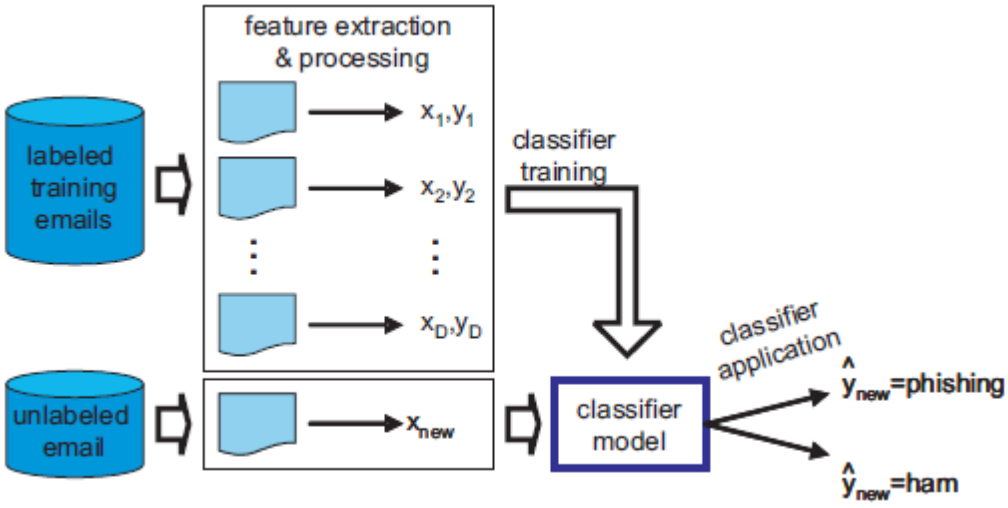


Figure 2.1: The machine learning approach [3].

Table 2.4: Feature categories and features extracted from emails.

Features Category	Features Extracted
Structural: Reflects the body part structure of an email	• Total number of body parts • Number of discrete and composite body parts • Number of alternative body parts
Link: Reflects properties of links contained in an email	• Total number of links • Number of internal and external links • Number of links with IP numbers • Number of deceptive links • Number of links behind an image • Maximum number of dots in a link • A Boolean indicating whether there is a link whose text contains one of the following words: click, here, login, and update
Element: Reflects the kind of web technologies used in an email	• HTML • Scripting • JavaScript • Forms
Spam Filter: Whether email is spam or not	• Filter test score • A Boolean of whether or not an email is considered spam
Word list: Keywords hinting possibility of phishing	• Account • Update • Confirm • Verify • Secur • Notif • Log • Click • Inconvenien

Advantages:

- ✓ Memory requirements compared to standard DMC approach are reduced by two thirds because of the use of Dynamic Markov Chain compression and adaptive training algorithm [3].
- ✓ Classifiers trained using features extracted by Dynamic Markov Chains and Latent Class-Topic Models outperform previous benchmarks [3].
- ✓ Features used are derived directly from the email and do not require information about specific websites like age of domains used in technique proposed by Fette et al., [5].

Limitations:

- If only basic set of features are used, the system performs worse as compared to the system of Fette et al., because extrinsic features (age of domain) are not used.
- Attackers may bypass the proposed solution using HTML layout tricks since it is a statistically-based technique [2] [4].

2.2.2 Take down method

Another server side technique to counter phishing is to take down a phishing website before it harms anyone. In email content analysis phishing emails were filtered but no action was taken against the attacker. This is both preventive and corrective approach as compared to email content analysis which is a preventive solution. Reported websites and URLs found in phishing emails are harmful websites that are removed from the internet in notice and take down method.

Shah et al., [6] extended email content analysis approach and proposed an improved solution called Pshark. Pshark is a proactive approach against phishing websites and works aggressively against attackers rather than just preventing the user from a phishing attack. Pshark methodology can be divided into 4 steps as shown in Figure 2.2; detecting phishing emails, locating host server, reporting to server administrator and aggressive Pshark approach. If the administrator fails to respond to the warning given by Pshark and does not remove the phishing website, in step4 either the server is taken down by reporting to legal authorities or by flooding the page with deceptive information.

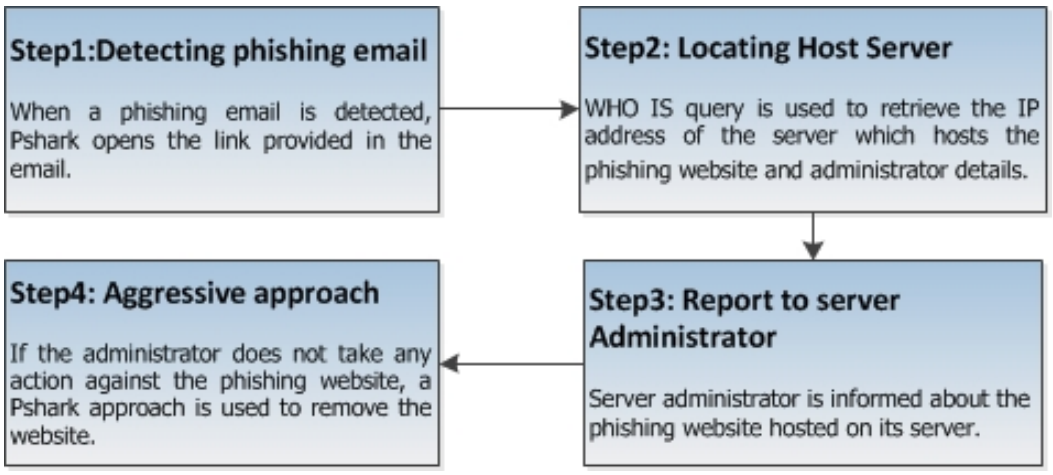


Figure 2.2: Shows the basic 4 steps of the methodology used by Pshark.

Advantages:

- ✓ Pshark is a proactive approach and takes down websites rather than just ignoring phishing attacks [6].
- ✓ Once the phishing website is taken down, all future attacks from that particular website are stopped [6].

Limitations:

- × Most phishing websites have an average age of 3.1 days and many disappear within hours [3]. The attacker might cause damage and remove the website by the time administrator acknowledges the message sent by Pshark.
- × Cannot stop an initial phishing email [6].
- × Pshark does not have an email filtering technique [6].
- × Shutting down a fake website is performed manually and still needs to be automated [6].

Many organizations like Netcraft, BrandProtect, Dell SecureWorks, Cyveillance, PhishLabs, MarkMonitor, Telefónica, VeriSign, FraudWatch International, Easy Solutions, Internet Identity, etc. assist in deactivating fraudulent websites and removing them from the internet. Table 2.5 shows the list of organizations and the services offered.

Table 2.5: list of organizations and services against phishing offered.

Organization	Services						
	Email authentication	Email filtering	Web filtering	Consumer toolbars	Takedown	Fraud analysis	Forensic Services
Netcraft	✗	✓	✓	✓	✓	✗	✗
BrandProtect	✗	✗	✗	✗	✓	✓	✓
Dell SecureWorks	✗	✗	✗	✗	✓	✗	✓
Cyveillance	✗	✗	✓	✗	✓	✓	✗
PhishLabs	✗	✓	✓	✗	✓	✓	✗
MarkMonitor	✓	✓	✓	✗	✓	✓	✓
Telefónica	✗		✗	✗		✗	✗
VeriSign	✓	✓	✗	✗	✓	✗	✗
FraudWatch International	✗	✗	✗	✗	✓	✓	✗
Easy Solutions	✗	✗	✗	✗	✓	✓	✗
Global Sign	✓	✗	✗	✗	✗	✗	✗
Go Daddy.com	✓	✓	✗	✗	✗	✗	✗
Iconix	✗	✓	✗	✗	✗	✗	✗
Symantec	✗	✓	✗	✗	✗	✗	✗
McAfee	✗	✓	✗	✗	✗	✗	✗

These organizations remove fake websites after a complaint has been made about a specific website or a phishing incident. On receiving a complaint, the reported website is checked whether it is a fraudulent website or not. Once that is confirmed, information about the owner is collected and the website is deactivated. Most of these organizations also ask Microsoft, Google and other major companies to update their blacklist. Figure 2.3 summarises the steps taken once a complaint is made.

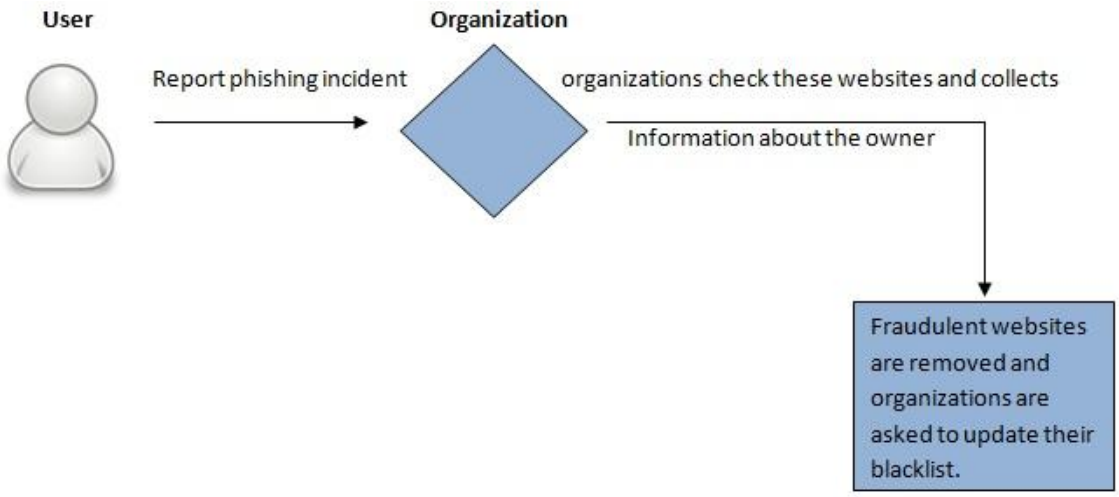


Figure 2.3: Process of taking down a website.

Table 2.6 summarises the two techniques; email content analysis and notice and takedown, discussed in this section.

Table 2.6: Comparison of Email content analysis and Takedown.

Email Content Analysis	Notice-and-Take-down
Preventive approach	Preventive and Corrective approach
Take no action against attacker	Take action against attacker
Fraudulent website may continue to harm many users	Once a fraudulent website is taken down, all future attacks are stopped from that particular website
Prevents phishing emails from reaching the user	First phishing email cannot be prevented from reaching the user

2.3 Client Side Phishing Solution

Mostly client side solutions are browser plug-ins or extensions that are installed on user's machine. These extensions monitor websites visited by users and informs them if they are about to enter a fraudulent page. There are many different solutions present to help users distinguish a fraudulent page from a legitimate one but each technique has some limitations. Attackers have constantly improved their methods which makes the job of detecting phishing websites even harder. Client side phishing solutions fall into two categories: Blacklist based method and heuristic approach.

2.3.1 Blacklist Based Method

Blacklist approach has been used in many other areas for quite some time and has recently been adopted as an anti-phishing solution. Blacklist is an access control technique that allows access to anything outside the list. An anti-phishing blacklist contains all the entries that are denied access [7]; where as a whitelist contains entries which are legitimate and denies access to anything outside the list.

In blacklist based anti-phishing, creating and maintain the list is the most important task. To create this list, URLs of phishing pages are retrieved from users directly, spam or phishing emails, or from various authentic websites. Anti-Phishing Work Group (APWG), PhishTank, OITC, SURBL, The DNS blackhole, ZeuS Tracker etc. are some of the organizations that serve the anti-phishing cause. Once a URL is reported, it is verified before it is added to the blacklist to reduce false positives. Different organizations have different ways of checking a URL; PhishTank classifies a URL as a phishing threat if it has at least 4 votes from users. Figure 2.4 shows a general model of how a blacklist is formed.

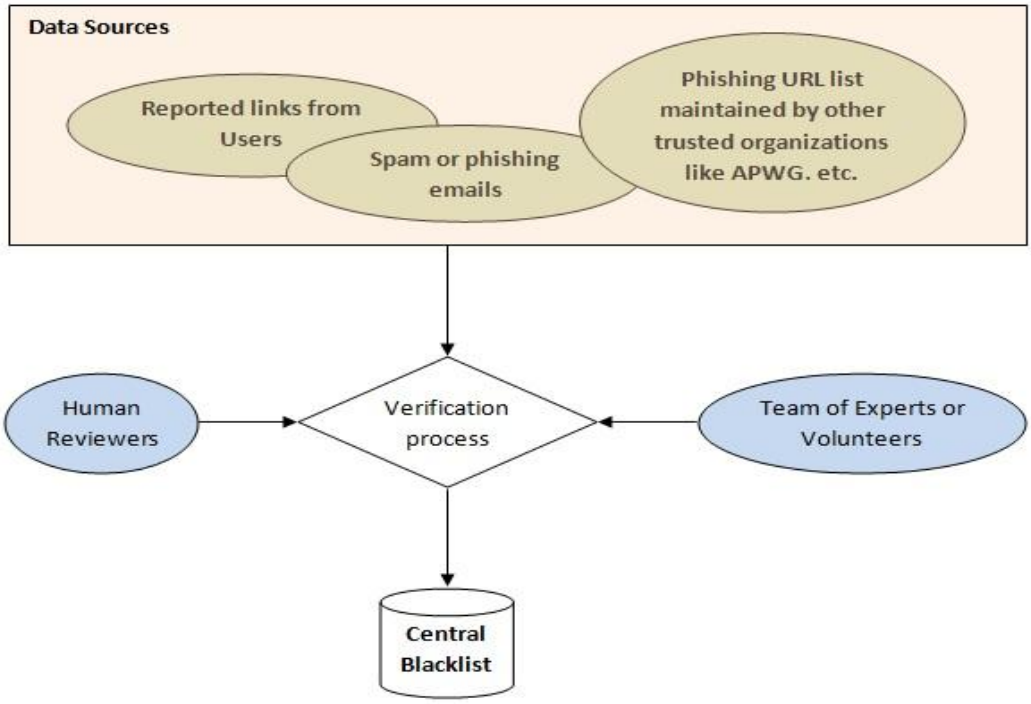


Figure 2.4: Steps involved in blacklist compilation.

Blacklist of IP addresses can also be used to stop attackers but usually a lot of websites are hosted on one server and this will terminate other legitimate websites on the same IP address as well. Therefore, blacklist of domains is preferred. Organizations like Sucuri, Symantec, Google etc. also maintain a blacklist and inform the users when they are

about to visit a blacklisted site. The blacklist method is used by Netcraft Toolbar [8], Cloudmark DesktopOne [9], Microsoft SmartScreen Filter [10], Firefox [15] and many others.

Netcraft toolbar helps Internet Explorer and Mozilla Firefox users against phishing attacks. There are 5 labels; Since, Country, Rank, Host and Risk Ratting, through which Netcraft assists a user in differentiating a phishing website from a legitimate one (see Figure 2.5).

Table 2.7 gives details of the labels and how they contribute in determining the risk rating. Netcraft's designers consider age of the website domain to be the most important factor in determining the risk rating.

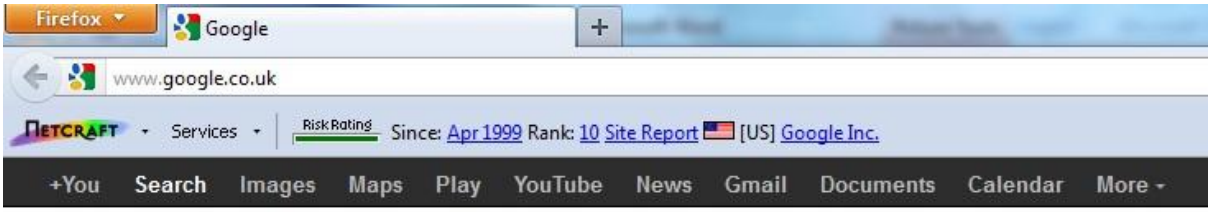


Figure 2.5: Netcraft Toolbar.

Table 2.7: Labels used by Netcraft and how they contribute in calculating Risk Rating.

Label	What is shown in the Toolbar		Risk
Since	The date, when this website was first seen in Netcraft Web Server Survey	Website formation date if the website is available in the Web Server Survey	Low Risk
		"New Website" if it is not available in Web Server Survey	High Risk
Rank	Number of times this website has been visited	most visited pages	Low Risk
		Least visited pages	High Risk
Country	Country where the website is hosted	If the hosting country does not have history of hosting phishing websites	Low Risk
		If the hosting country has history of hosting phishing websites	High Risk
Host	Name of Organization hosting the current site	If the hosting company has no history of hosting phishing websites	Low Risk
		If the hosting company has previously hosted phishing websites	High Risk
Risk Ratting	Calculates the risk involved in visiting this website	Shows how trustworthy this website	

Advantages:

- ✓ Netcraft toolbar can deal with DNS poisoning. If a website that is supposed to be hosted in USA is hosted in India, Netcraft Toolbar will highlight this problem and show the country's name in the toolbar.
- ✓ Protects users against popup windows which hides the address bar and browser navigations [17].

Limitations:

- × Most phishing sites are hosted on hacked servers which host all legitimate websites and have been active for quite some time which means the attackers will be able to bypass the Since and Host labels.
- × Country label is user dependent and users have to notice the fact that a website is not hosted in a country where it is supposed to be.

Microsoft SmartScreen Filter is a tool for Internet Explorer 9 (IE9) users which uses blacklist and heuristic analysis to determine whether the page is phishing or legitimate. When a user visits a page using IE9, the contents of the page are compared against heuristic characteristics. If the page fails to pass the heuristic test, a yellow shield will appear warning the user about the contents of the page and will suggest the user not to enter any confidential information. However, if no suspicious properties are found, the tool will check its URL against a blacklist. If a match is found in the blacklist, a red shield will appear informing users about the blacklisted page. It is then up to the users whether they want to proceed or cancel the page. Users can also report to Microsoft about new fraudulent URLs using SmartScreen reporting feature. SmartScreen blacklist verifies ULRs (see Figure 2.4) before adding them to the blacklist.

Advantages:

- ✓ SmartScreen provides additional security in a network as it allows the administrator to set up a group policy which restricts users from ignoring warning shown by SmartScreen Filter [11].
- ✓ SmartScreen also provides protection against downloadable malicious files like key-loggers.

Limitations:

- Blacklist needs to be updated regularly and users will be vulnerable to newly created phishing websites [7].

Firefox, one of the most used browsers available [16]; also uses blacklist-based approach to protect users against phishing. Other major browsers like safari [14] and Internet Explorer [10] also use this approach. Firefox maintains 2 blacklists, remote and local. The local blacklist is downloaded on the user's browser and is updated every 30 minutes from an update server. This reduces network queries but a delay is introduced in blacklist and performance may suffer. The remote blacklist on the other hand is updated and more comprehensive as well because the local blacklist may be pruned due to size limitations. However, the remote blacklist is hosted on a lookup server and each time a query is made, it is encrypted before sending to increase the security. Response time is the main drawback of remote list. When a blacklisted page is detected, the page is disabled and user is notified about it by a warning sign. The user then has the option to either continue or close the page. Firefox also gives its users the option to report false negatives and false positives.

2.3.2 Heuristic Method

Heuristic approach uses HTML, website content or URL signatures to identify phishing pages. According to Sheng et al., [13], blacklists are less effective as compared to heuristic when protecting users at the start and tools that use heuristic and blacklist together are more accurate than tools with only blacklist.

There has been a lot of research in the past on heuristic approach. Phishing websites are analysed to make heuristics which are then later used to classify websites as either phishing or legitimate using machine learning models. Garera et al., [21] discovered heuristics by analysing existing phishing URLs, while Ludl et al., [20] analysed page structure.

The technique proposed by Garera et al., [21] relies on analysing URLs to differentiate between a benign and phishing URL. A logistic regression classifier classifies a URL as either benign or phishing on the basis of 18 URL features selected. These features can be categorised into 4 groups; page based, domain based, type based and word based. The model was trained using a data set which consisted of a blacklist and a whitelist. 1220 URLs from the blacklist maintained by Google were used as training blacklist

along with 113 most popular URLs as training white list. Experiments preformed with the dataset and logistic regression algorithm yielded coefficients of the URL features, which showed that “White Domain Table” and “host obfuscated with IP” are the most informative features in determining phishing URLs.

Advantages:

- ✓ No need to maintain a blacklist.
- ✓ Can identify and report an attack as soon as it is launched, no need to wait for an updated blacklist.

Limitations:

- × High false positive rate.

According to Ludl et al. [20], the structure of a page can be used to distinguish between a legitimate and phishing website. 18 page properties are defined that can be extracted from the HTML and URL of a page. Table 2.8 shows the features and their sources. C4.5 decision tree is used as a classifier with a dataset that comprises of 680 phishing and 4149 legitimate pages. This model correctly classified 284 phishing websites but misclassified the other 396. The experiment shows that structural properties of websites can be used to determine their nature but the selection of these properties is very important and a larger number of properties will yield better results.

Table 2.8: page properties and their source.

Properties	Source
Number of Forms	HTML
Number of input fields	HTML
Number of text fields	HTML
Number of password fields	HTML
Number of hidden fields	HTML
Number of other fields	HTML
Number of internal links	HTML
Number of external links	HTML
Number of other links	HTML
Number of internal secure links	HTML

Number of external secure links	HTML
Number of secure images	HTML
Number of external images	HTML
Number of external references	HTML
Number of whitelist references	HTML
Number of JavaScript tags	HTML
Suspicious URL	URL of page
Uses SSL	URL of page

Advantages:

- ✓ Like all heuristic approaches, no need to maintain a blacklist and can identify and report an attack as soon as it is launched.

Disadvantage:

- × High true negative rate. Too many phishing websites are classified as legitimate.

Chou et al., proposed and implemented a client side anti-phishing browser plug-in, SpoofGuard [22, 23]. SpoofGuard evaluates a spoof index for each page which determines whether a page is legitimate or phishing. The spoof index is computed on the basis of page properties: domain name, link, URL, password, outgoing password, referring page, post data and image checks. If the computed spoof index is greater than a pre defined threshold value, the page is classified as phishing and the user is notified about the threat. If the spoof index is less than threshold value, the page is classified and legitimate.

Advantages:

- ✓ Minimum or no user input required.
- ✓ High true positive rate.

Limitations:

- × High false positive rate.
- × Not effective against modern phishing attacks.

Mohammed Baihan [24] extended SpoofGuard's functionality and implemented SpoofGuard++. He enhanced the existing SpoofGuard functionality and added new functions as well to counter new phishing attacks. Existing SpoofGuard Image, URL and link check functionality was enhanced. HTML5 threat detection, Cross site

scripting, URL shortening threat detection, HTML attachment attack detection and tabnabbing attack detection functions were added to enhance SpoofGuard. 2 main components of SpoofGuard++ are DOM tree extractor and Phishing assessment manager. The output from DOM tree is used as input to Phishing assessment manager. Unlike SpoofGuard; which was available for both Firefox and Internet Explorer users, SpoofGuard was developed only for Internet Explorer 9 users.

2.4 Things learned from existing solutions

Most of the anti-phishing solutions discussed have limitations and require some sort of user input. In order to alleviate the growing phishing problem, it is important to identify a fake website and notify users when they encounter one. According to Jagatic et al., [29] users tend to ignore warning messages because extra effort is required. Majority of the existing industry and research solutions rely on either the blacklist model or heuristic based approach. According to Sheng et al. [13], blacklist and heuristic based approach when used together is the most effective approach against phishing. Therefore, the focus of this work will be to combine the two client side anti-phishing techniques and devise a solution that not only efficiently identifies fraudulent websites but also ensures that users pay attention to the warning message.

2.5 Chapter Summary

In this chapter most of the anti-phishing solutions; both industry and research, were discussed in detail. A thorough critical analysis of available solutions was carried out and advantages and limitations of each solution were discussed. There is no doubt that these are some of the best available solutions, but are effective against some phishing attacks and not all. In addition some of these solutions lack in signifying the importance of warning messages which are ignored by the users.

Computer security is a field where complete security cannot be achieved as attackers develop new techniques frequently to bypass security. This project tries to minimize phishing threat and implement a system that is flexible enough to accommodate future changes.

Chapter 3 PhishProof design

3.1 Chapter Introduction

This chapter gives an overview of the proposed anti-phishing solution, PhishProof. In order to design an efficient and effective solution, requirements for the system are identified. The design of PhishProof, its building blocks and architectural components are also explained in detail.

This chapter is structured as follows: Section 3.2 presents the overview of PhishProof. Section 3.3 describes the requirements of the proposed system. Section 3.4 explains the architectural design and building blocks of PhishProof system. To conclude design chapter, section 3.5 summarises design of proposed anti-phishing solution.

3.2 PhishProof Overview

PhishProof is an anti-phishing tool designed to help Firefox users distinguish between phishing and legitimate websites. PhishProof does not require any effort from the users to identify a phishing website. When the system evaluates a website as a phishing website, users are notified immediately via an alert message which makes the system easy to use even for naïve users. PhishProof can be installed on any system that has Firefox (version 12 and later). After installing PhishProof toolbar, users will be able use the browser normally, and will be notified if they visit any potential phishing website. Risk rating for each website visited by the user will be displayed in the toolbar along with the risk rating percentage. To constantly improve the system, a website (www.phishproof.com) and a website management admin panel (www.phishproof.com/admin) is also developed along with the toolbar. Users will be able to report phishing URLs via this website and also assist in improving PhishProof by giving their valuable feedback.

The proposed solution is a combination of two client side anti-phishing methods; blacklist and heuristic, discussed in chapter 2. As its name (PhishProof) suggests, it aims at making users phishing proof by using the best of each method to compliment the limitations of other.

3.3 Requirement specifications

The first step of finding a solution is to understand the problem, which in this case is phishing. Since a lot of work has been done in both industry and research, the best way forward is to do a detailed literature review and study the work done in the past to find out where existing solutions are lacking. Once the system requirements are identified, the system can be designed to meet the requirements specified. The design phase is a pre-requisite for the implementation phase. After designing the system, the proposed design is implemented using a programming language.

In light of the literature review, the main requirements of PhishProof can be narrowed down to functionality and performance.

3.3.1 Functionality Requirements

PhishProof should be able to detect and report phishing websites, and help users distinguish between a phishing and legitimate website with minimum user input. In order to do so the following functional requirements must be met:

1. **Ability to identify websites with short life time**: According to APWG [18], phishing websites have an average age of 3.6 days and most of them disappear within hours. Anti-phishing solutions relying only on blacklist are not able to perform against websites that are active only for a brief period of time. PhishProof should be able to identify websites with short life time.
2. **Ability to identify websites hosted on compromised domains**: Netcraft Toolbar is a client side anti-phishing solution and like many others its risk rating functionality heavily relies on age of domain [8]. Attackers can bypass this by hosting their phishing site on hacked servers which have been active for long. PhishProof should be able to identify websites on compromised domains.
3. **Ability to grab user's attention with warning messages**: When users come across a blacklisted site, most solutions shows a warning message with "Yes" and "No" options asking users whether they want to continue or not. According to a research by Jendricke et al., [19] most users will dismiss the warning without reading it. PhishProof should be able to grab user's attention with its warning notifications.
4. **Ability to identify and report websites with links to phishing websites**: Most attackers redirect users to a phishing website from any legitimate website.

PhishProof should also be able to report pages that contain blacklisted URLs in their body.

5. **Ability to check if a website uses encryption to transfer passwords**: Most websites that request passwords have secure certificate installed. Attackers often ignore this installation as some cost is involved. PhishProof should be able to recognize domains that do not have secure SSL installed.
6. **Ability to identify the referrer to a website:** Most attackers use email messages to redirect users to phishing websites. PhishProof should be able to identify the referring page and increase risk rating if it is an email provider.
7. **Ability to identify blacklisted websites**: There are many website that have already been identified as phishing websites. PhishProof should be able to restrict access to these blacklisted websites.
8. **Ability to identify most visited websites**: There are many trusted websites which are visited regularly by majority users. PhishProof should be able to identify these white-listed websites and allow access without performing additional checks.

3.3.2 Performance Requirements

PhishProof toolbar is a Firefox extension that uses both blacklist and content analysis to help users distinguish between phishing and legitimate pages. Therefore it will require browser memory and time to calculate risk rating. The performance of Firefox should not be affected by PhishProof operations and web-page access time should be kept to minimum.

3.4 PhishProof Architectural Design

This section explains the mechanics of PhishProof. Section 3.4.1 presents a brief overview of PhishProof's architecture, while section 3.4.2 describes the building blocks and architectural components.

3.4.1 Architecture Overview

PhishProof uses blacklist and content analysis method to evaluate a page as either phishing or legitimate. The combination of these two techniques allows PhishProof to fill some of the gaps left by previous anti-phishing solutions. It requires browser memory and time to perform all checks and calculate risk rating of a page.

When a user opens Firefox, PhishProof is initialised and starts performing its operations by loading whitelist, blacklist, mail list, countries list and hosting company list. Whitelist and blacklist are maintained on main PhishProof server and managed using website administrator panel while other 3 lists are part of installation package. Since whitelist and blacklist are downloaded at the start of each browser session, any changes made to either of the lists will be available to users when they start a new browser session. Whitelist includes most commonly visited websites like Facebook, Google etc. There are two reasons why these URLs are included in the whitelist. First, these are global organizations and can be trusted that they won't host phishing websites to con users. Second, these organizations are leading technology companies and take phishing very seriously [31] [32] [33] [34]. Whitelist helps in improving performance of the toolbar as it allows white-listed websites to be loaded without additional checks. Blacklist contains URLs that have already been identified as phishing websites. This helps in filtering out websites that have already been classified as phishing websites without performing additional checks. This reduces the time and number of operations required to assess a page, thus improving performance of PhishProof.

If the URL entered by the user is not found in either of the lists, the website is evaluated on the basis of its content. A risk rating percentage is calculated by performing content analysis. Risk rating value determines whether the website is a potential threat to the user or not. Website content is analysed using 6 different labels. Each label is evaluated individually and its score is multiplied by weight assigned to that label. The total risk rating is the sum of all individual labels. If the value of total risk rating is above a threshold value, the page is evaluated as a potential threat and users are notified about it.

PhishProof also gives its users the added advantage of being able to report suspicious URLs via the PhishProof website. These URLs are then examined by PhishProof administration team via the backend administrator panel. If these URLs are evaluated as potential phishing threats, they are then added to PhishProof blacklist via same administrator panel. Figure 3.1 shows PhishProof architecture.

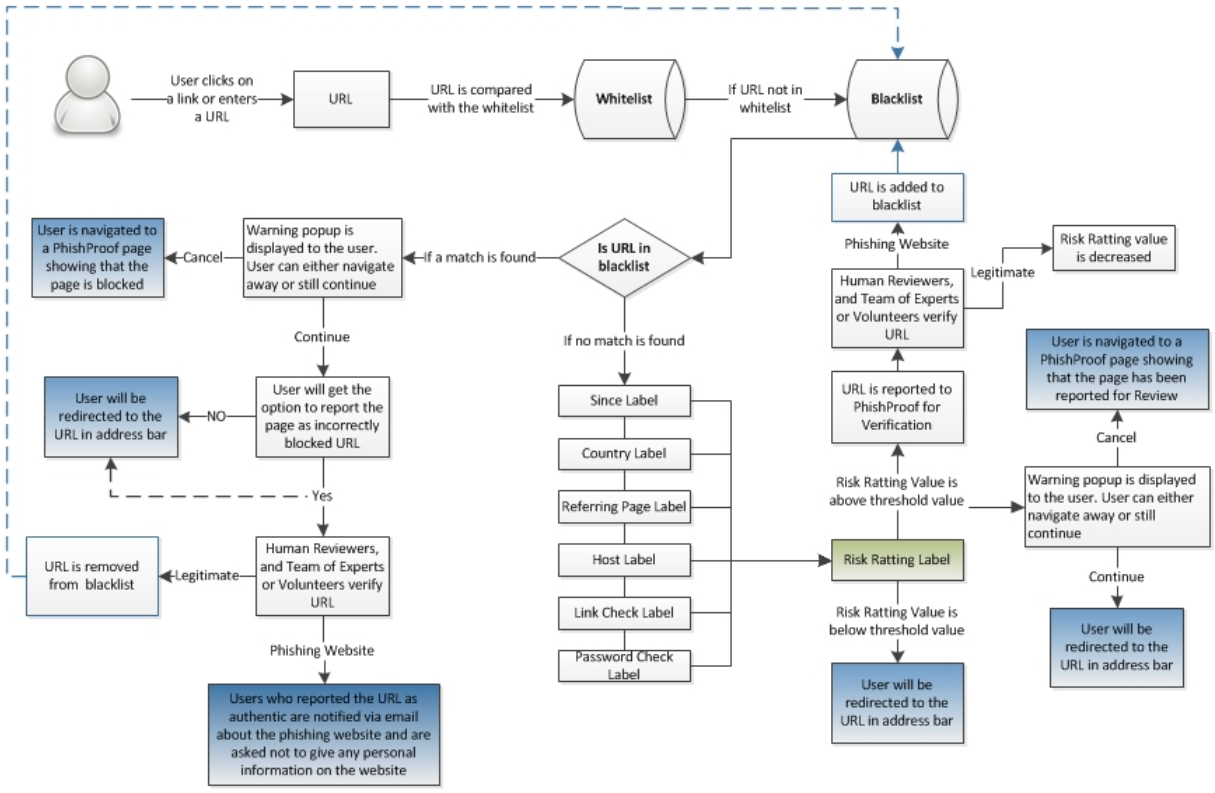


Figure 3.1: PhishProof Architecture.

3.4.2 Architectural Components – Building Blocks

PhishProof system can be divided into 3 main components; PhishProof toolbar, PhishProof website and PhishProof backend administrator panel. PhishProof toolbar performs all phishing checks while the other two components are used to support and manage it. Figure 3.2 shows the architectural components of PhishProof.

Figure 3.2: PhishProof Architectural Components.

3.4.2.1 PhishProof Toolbar

PhishProof toolbar is the main component of proposed anti-phishing solution. All phishing checks and anti-phishing techniques discussed in the literature review are implemented in this component. It is also responsible for calculating and displaying the risk rating percentage. The warning alerts used to notify users about potential threats are also generated and managed by this component. PhishProof toolbar uses blacklist and web page content analysis techniques to counter phishing.

3.4.2.1.1 Blacklist Based Method & Web Page Content Analysis Based Method

An anti-phishing method is only good if it's able to detect and stop phishing attacks. Therefore blacklist and web page content analysis methods will be investigated to see how effective these two techniques are. Blacklist based methods stop phishing attacks by restricting access to websites that have already been identified as phishing (section 2.3.1). Web page content analysis method (also known as heuristic method) prevents users from phishing attacks by analysing web page features and comparing them against a set of features. A webpage is identified as a potential threat if its features are similar to the features of a phishing page (section 2.3.2).

Blacklist based solutions require storage space as lists need to be managed and updated periodically for the solution to be effective. Web page analysis does not require any storage space to detect phishing attacks. Blacklist and whitelist are used to eliminate the overhead of analysing the contents of a web page that has already been identified as a phishing threat. According to Sheng et al., [13], web content analysis is more effective as compared to blacklist method but the most effective countermeasure against phishing is a combination of web content analysis and blacklist method. Therefore PhishProof toolbar implements both the methods together to give users 3 level protection against phishing as shown in figure 3.3.

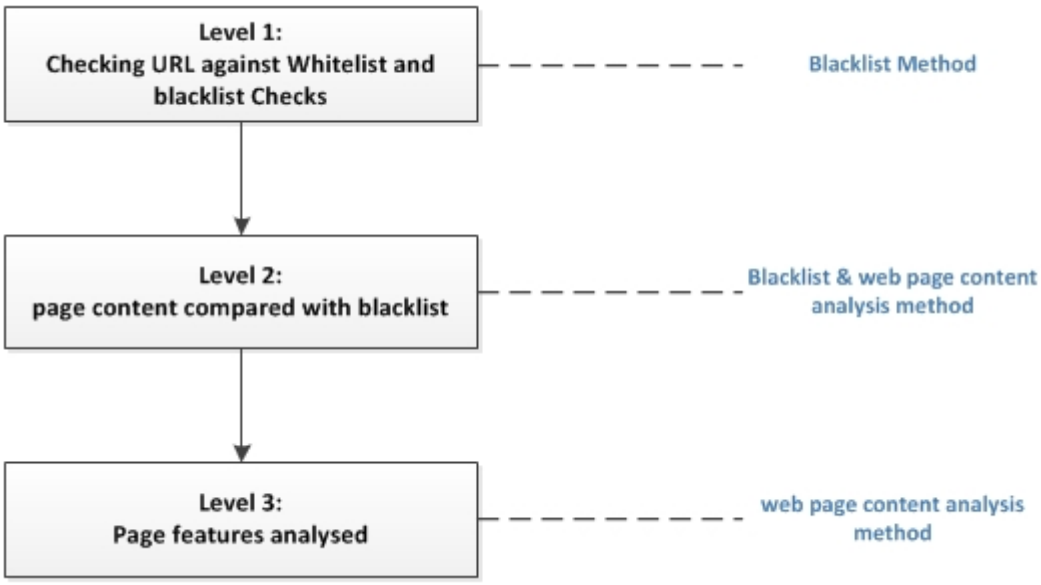


Figure 3.3: PhishProof's 3 level protection against phishing.

3.4.2.1.2 Level 1

In level 1 blacklist based method is used to prevent users from phishing attack. This acts as the first line of defence against phishing. When a user opens Firefox browser, PhishProof starts its operations by loading the whitelist, blacklist, country list, hosting company lists and Mail list into user's browser as shown in figure 3.4. By the time Firefox default screen is loaded, all lists are loaded into user's browser.

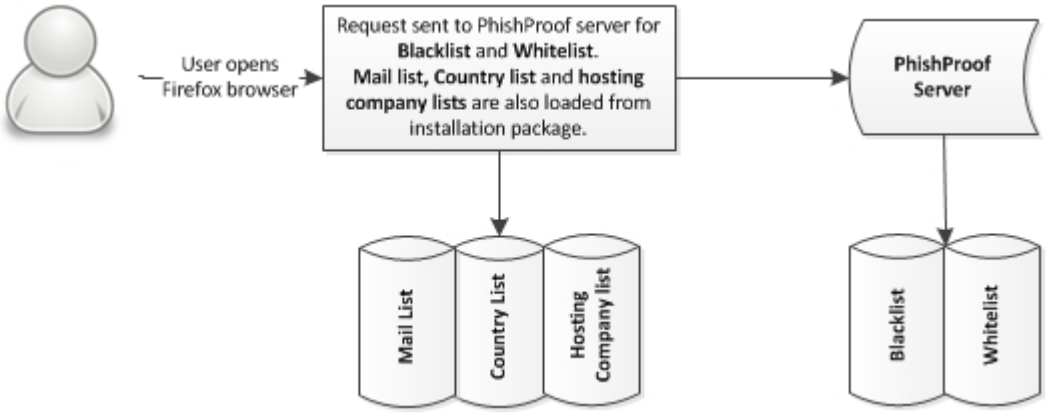


Figure 3.4: Loading lists at start of a browser session.

The lists are loaded in negligible time and users won't have to wait for lists to be loaded. Even if lists take more than expected time to load, the browser will function normally as loading process runs in background. Whitelist and blacklist are loaded from PhishProof server while the other three lists are part of the installation package. The lists are loaded at the start of each browser session so that the performance of Firefox is not affected and users don't have to wait for a network query to execute each time a

page is loaded. This also significantly reduces load on the server as lists are loaded for each browser session and not for each webpage viewed.

Only whitelist and Blacklist are used in first level protection while the other three lists are used in level 3 protection (see section 3.4.2.1.4).

3.4.2.1.2.1 Whitelist

Whitelist is a list stored on main PhishProof server that contains domain name of websites that are trusted by PhishProof. Websites such Google.com, Facebook.com etc. (see table 3.1) have been added because these are most commonly visited websites [35]. Whitelist improves the performance of PhishProof as it does not perform additional checks for a website found in whitelist. It can be updated from the backend administrator panel which gives options to add, delete and edit domain names.

Table 3.1: PhishProof Whitelist.

Domain Names
facebook.com
google.com
login.live.com
paypal.com
uk.msn.com
accounts.youtube.com
accounts.google.com
youtube.com
twitter.com
en.wikipedia.org
uk.yahoo.com
login.yahoo.net
login.yahoo.com
amazon.com
bbc.co.uk
ebay.com
ebay.de
bing.com
linkedin.com
google.co.uk

3.4.2.1.2.2 Blacklist

Like whitelist, blacklist is also stored on the main PhishProof server. Blacklist contains domain name and URLs of websites that have already been identified as phishing websites (see table 3.2). PhishProof restricts access to blacklisted pages and no

additional checks are required to evaluate a page. This significantly reduces the number of operations required to access a website and improves performance. Blacklist can be managed from the backend administrator panel. It provides option to add, delete and edit domain names and URLs. Users can also help in improving the blacklist by reporting suspicious URLs via PhishProof website.

Table 3.2: PhishProof Blacklist.

Domain names & URLs
http://85.17.93.40/pp/42e964e3057530ceeda70d42dd991e36/
http://www.batameasyhotel.com/pypalscm/
http://refbeg.info/bey4I1zj58mp7h913027/amF6em1hbjY2QHQtb25sa W5ILmRI/xDEsdR/a2x1bXRvLmNvbQ=Message-I
www.sialda.pt
http://refbeg.info/aTcntrIde/webscr_prim.php?cmVmYmVnLmIuZm8= uhsdsusu5485757kUJHNN546221oPLKj988777A
http://sherriffshomoeopathy.com/paypal.com/secure-code106/security/

3.4.2.1.2.3 Whitelist & Blacklist Check

In Level 1, whitelist and blacklist are used to protect users against potential phishing threats. When a URL is entered by a user, PhishProof toolbar extracts the domain name and compares it with preloaded whitelist. If the domain name is found in PhishProof whitelist, the website is considered legitimate and toolbar stops all operations. If the domain name is not found in whitelist, it is compared against the blacklist. If the domain name is found in PhishProof blacklist, the website is a phishing website and users are notified via alert popup. If users still choose to ignore the alert message and continue to visit the page, an audio alert is played by PhishProof to get users attention as shown in figure 3.5.

Alert Notification

As mentioned above, Phish Proof toolbar uses alert messages to notify users about websites that are considered a potential threat. PhishProof's alert message has "Cancel" and "Ok" buttons, where cancel button discards the warning message and "OK" button stops users from viewing blocked pages. "OK" button is used to restrict users from visiting blocked pages because according to Aza Raskin [47], most of the users click "OK" without reading the alert message. Hence, users who click OK without paying attention to warning message will be protected against websites identified as phishing

pages. Users who click “Cancel” will be allowed to view restricted page and an audio warning message will also be played to get their attention.

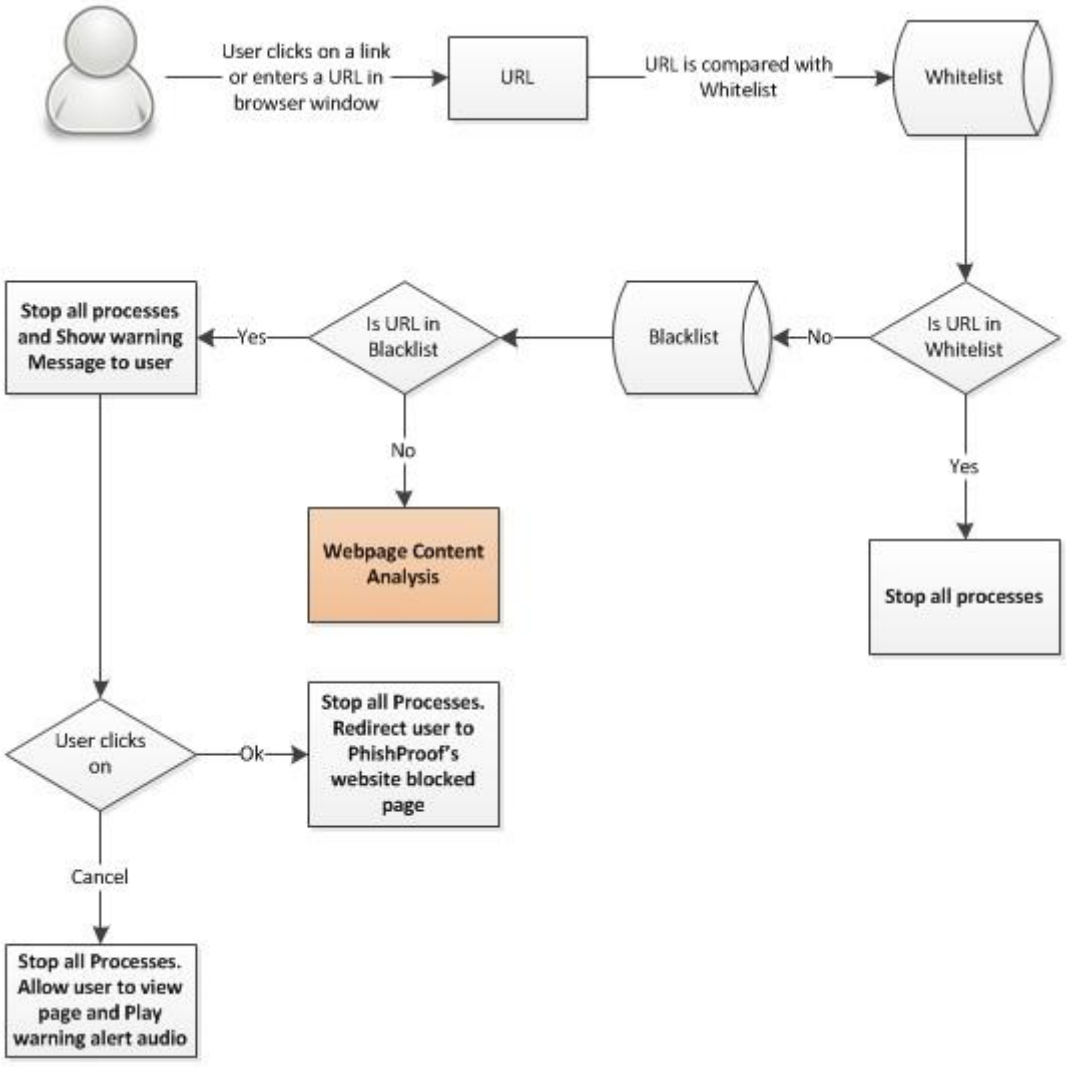


Figure 3.5: Whitelist & blacklist checks.

If the website’s domain name and URL are not found in either of the lists, PhishProof toolbar will advance to Level 2.

3.4.2.1.3 Level 2

In level 2 a combination of web page content analysis and blacklist based methods is used to counter phishing attacks. Level 2 has two main checks, referrer check and links check. Figure 3.6 shows an architecture overview of level 2.

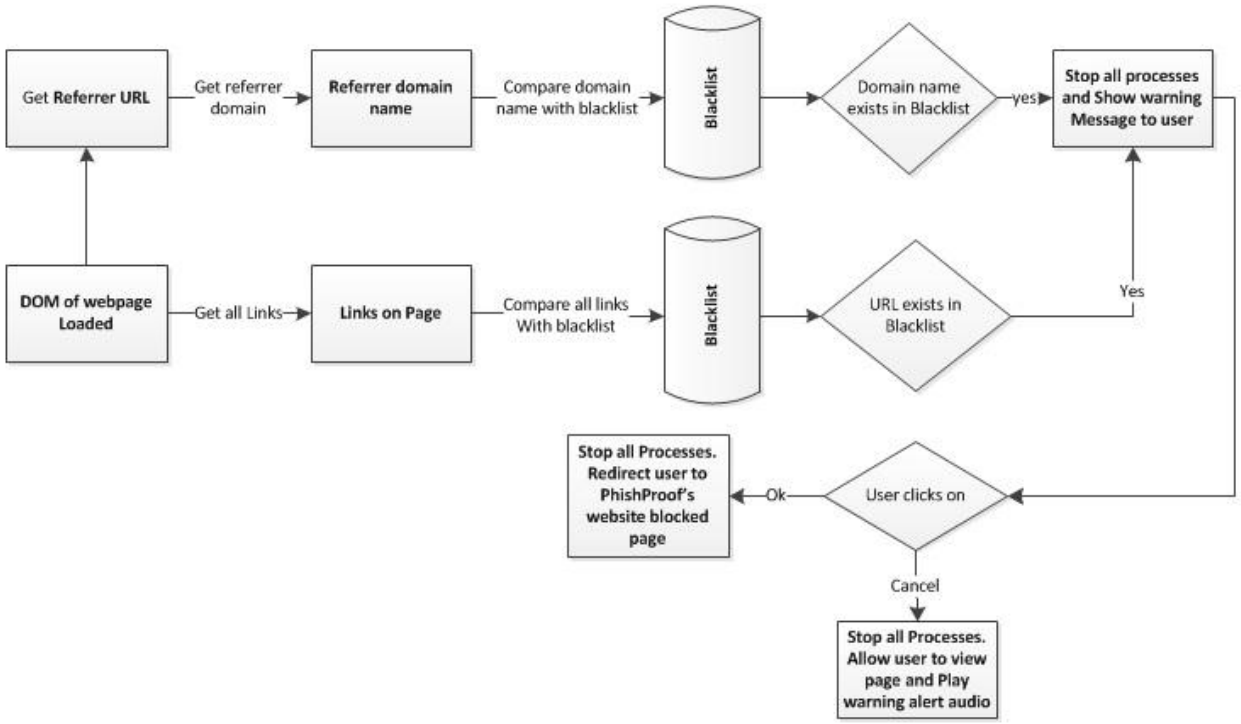


Figure 3.6: Level 2 Architecture Overview.

3.4.2.1.3.1 Referrer Check

This check examines details of referrer website. Most websites redirect users to other domain addresses and the URL typed in the address bar is not the final location where users land. To counter this phishing technique PhishProof checks the domain name of referrer website. Referrer check extracts referrer URL from Document Object Model (DOM) of current loaded page and compares it against PhishProof blacklist loaded in Level 1. If the referring page is blacklisted, users are notified via alert message. If users still proceed to web page by ignoring the alert message, an audio warning message is played to get their attention. If users acknowledge the warning alert, they are redirected to another page which shows that requested website has been blocked by PhishProof.

3.4.2.1.3.2 Link Check

This check analyses all links on current webpage. Most phishing attacks are carried out on compromised websites and servers and therefore attackers might be able to bypass blacklist checks. The compromised websites and phishing emails involve one bad link which redirects users to a phishing page. PhishProof checks all referring links on a website against an up to date blacklist and classifies the current page as a potential

threat if any link is found in blacklist. This helps in identifying compromised domains and restricts users from giving any confidential information.

3.4.2.1.4 Level 3

Level 3 is the last level of protection against a phishing attack. This involves web page content analysis and calculates a risk rating percentage which determines whether the web page is a threat or not. If risk rating calculated is above a threshold value, the current page is classified as a potential threat and users are notified via alert message. 6 Labels; Referrer, Password, Encryption, Age, Country and Hosting, contribute to the final risk rating percentage. Level 3 protection can be divided into 2 rounds; first round where all labels are computed individually and second round where all labels combine to compute risk rating percentage.

3.4.2.1.4.1 First Round

In first round referrer, password, encryption, age, country and hosting checks are done. These checks are done individually and their individual scores are assigned. All Labels (L_i) are assigned score, either 0 or 1. $L_i = 0$ means it is a legitimate page, whereas $L_i = 1$ means the current page maybe a phishing page. All labels are assigned value in this round and are added together to give risk rating percentage in second round.

The web page features required to compute labels are extracted using DOM and analysed to find any characteristics identical to those of a phishing website. Figure 3.7 shows the architecture overview of First round.

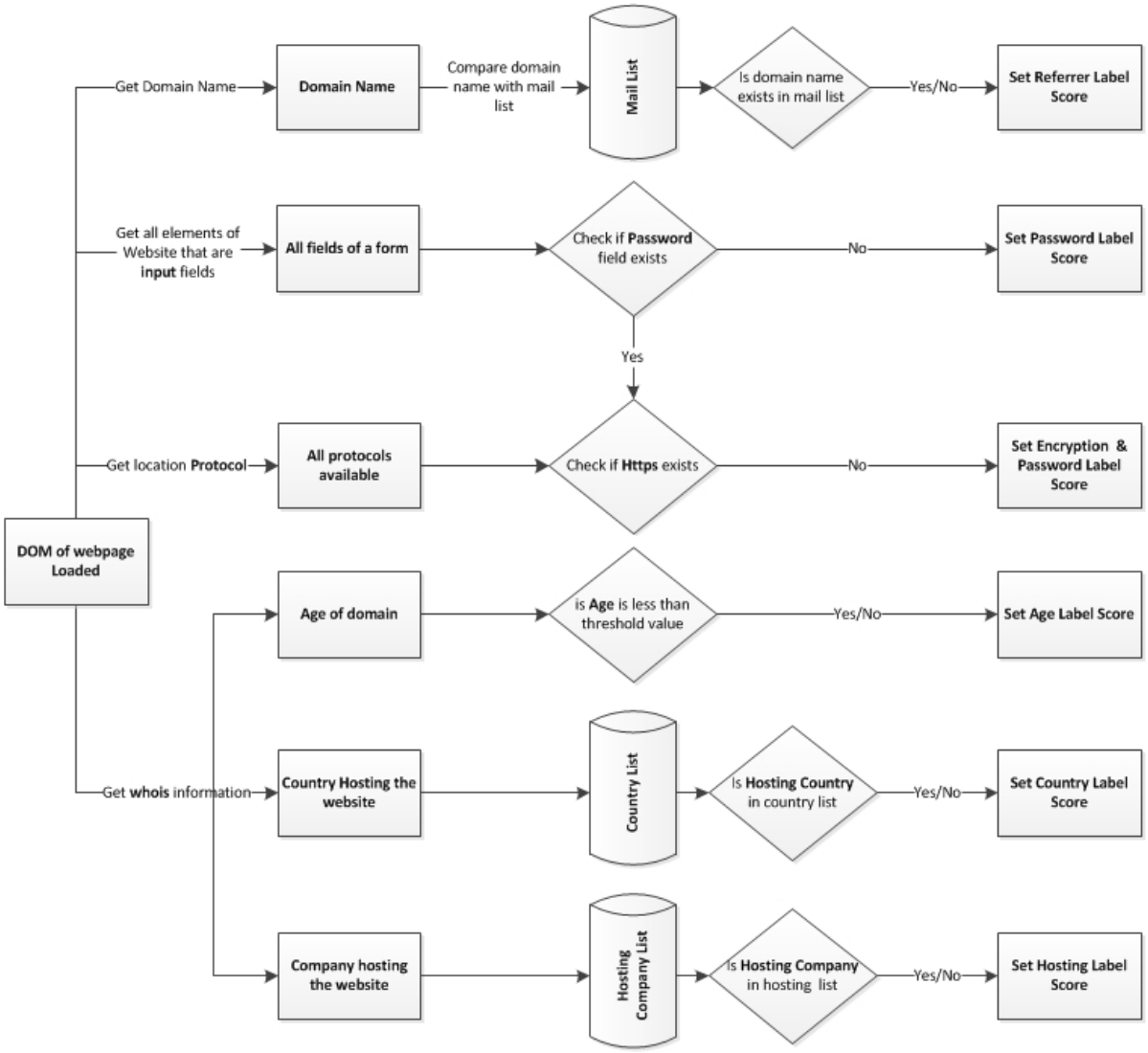


Figure 3.7: First Round Architecture Overview for Level 3.

Referrer Label

Referrer Label checks how the user has reached current page. Most phishing attacks are initiated by sending fake emails to a large number of internet users [36]. Therefore pages referred from emails are considered a higher potential threat as compared to pages referred from other websites. This check gets the value of referring page from DOM of current page and compares it against the Mail list loaded in level 1. Mail list contains the domain names of email providers like Gmail, Hotmail, Yahoo etc (see table 3.3). If value of referrer domain matches with any value in Mail List, referrer label score is set to high risk.

Table 3.3: PhishProof Mail List.

Domain Names
login.live.com
mail.google.com
toast.cs.man.ac.uk
mail.yahoo.com
hotmail.com

Password Label

Since phishing is about getting personal details of users, therefore almost all phishing pages have a password field. Password label check scans the current webpage to find any input field of type password. If PhishProof does not come across any input field of type password, it sets a low score for password field. Whether a page has more than one password field or not does not affect password label score. If any single input field with type password is found, the domain is also checked for encryption protocol SSL (Secure Sockets Layer). All web pages that take users sensitive information should be encrypted and user credentials should be protected [37]. Websites such as Facebook, HSBC, and Google etc. that require sensitive information from users use SSL to encrypt user information. If SSL is not used by a page requesting password, there is a high probability of website being a phishing website. Hence password label score is increased.

Encryption Label

Majority phishing pages don't use SSL authentication protocol to transfer user information. Attackers avoid installing this service because some cost is involved. Phishing websites are active only for a short time and attackers avoid paying anything for such short time. Therefore SSL is considered as an important property in determining risk rating of a page. If a webpage requests password and does not use SSL to encrypt user data, encryption label score is set to high risk.

Age Label

Phishing websites have a very short life time. They have an average age of 3.1 days and many disappear within hours [3]. This is one of the reasons why blacklists need to be updated regularly. PhishProof takes into account this property of phishing websites and

calculates the age of each website it encounters. If the age of the website is less than a threshold value, age label score is set to high risk.

Country Label

There are countries that have a higher probability of hosting phishing websites as compared to other countries [38]. Countries are classified as potential threat if they have a high ratio of hosting phishing websites to legitimate ones. If the hosting country has a history of hosting phishing websites the country label score is set to high risk. If the country of hosting does not have a history of hosting phishing websites, country label scores is set to low risk. Table 3.4 shows the list of countries considered as high risk by PhishProof.

Table 3.4: PhishProof Countries List [38].

Countries	Flag	Country Code
Bhutan		BT
Morocco		MA
Cameroon		CM
Djibouti		DJ
Algeria		DZ
Cote D'Ivoire (Ivory Coast)		CI
Malawi		MW
Virgin Islands (US)		VI
Zambia		ZM
Brunei Darussalam		BN

The country label also assists expert users in identifying phishing websites. PhishProof toolbar displays the country code and flag for each website visited by the user. An expert user will be able to identify a phishing attack if a website such as Facebook.com is hosted in any country other than USA.

Hosting Label

Some hosting companies have a higher probability of hosting phishing websites as compared to other hosting companies [39]. Hosting companies that have a history of hosting phishing websites are considered as high risk as compared to other hosting companies. If the hosting company has a history of hosting phishing websites, the hosting label score is set to high risk else hosting label score is set to low risk. Table 3.5 shows the list of hosting companies considered as high risk by PhishProof.

Table 3.5: PhishProof hosting companies list [39].

Hosting Companies
Lunarpages
UK2 Group
SingleHop, Inc
main-hosting.com
IQ PL Sp. Ltd
Softlayer Inc
iWeb Technologies inc
ionity.com
Lycos
almouroltec.com
OVH Net
Atlantech Online, Inc
HostDime.com
jump.ro
LiquidWeb Inc
Gigenet
UOL

3.4.2.1.4.2 Second Round

In second round, all individual labels (L_i) set in first round are combined with preset weights (W_i) to compute risk rating as shown in figure 3.8. Weights are assigned to give each label different weightage as some labels are more important in categorising a page as compared to others. For instance, a website requesting password and not using authentic encryption will be considered more of a threat as compared to a website hosted in any country from table 3.4.

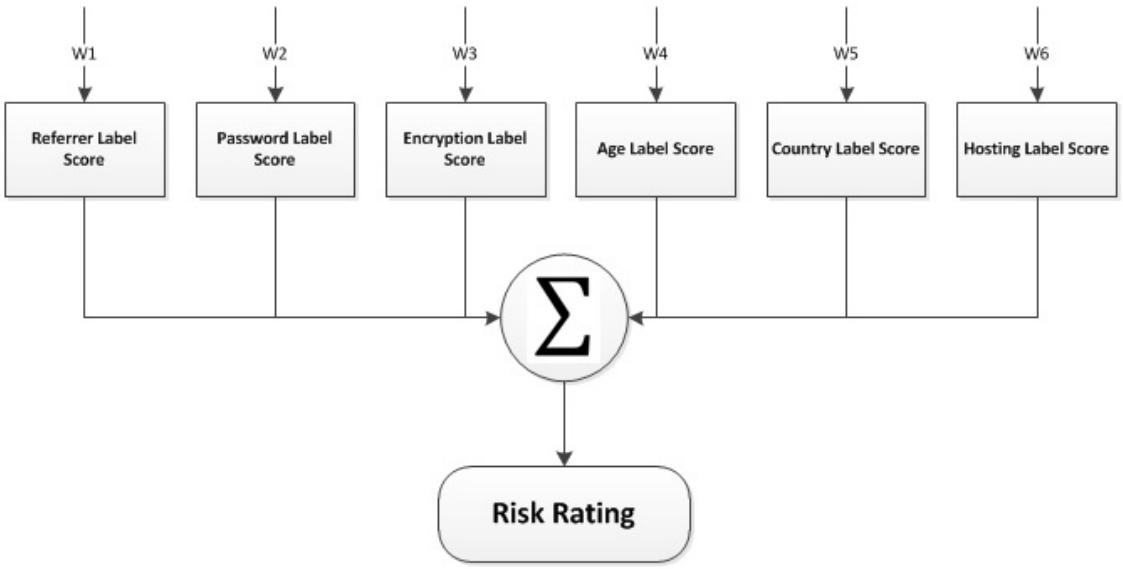


Figure 3.8: Architecture of Second Round in level 3.

Risk rating is the ratio of the weighted scores to total score as shown in Equation 3.1.

$$\text{Risk Rating} = \frac{\sum_{i=1}^{n}[(w_i)*(L_i)]}{\sum_{i=1}^{n}(w_i)}$$ **(Equation 3.1: Total Risk Rating Calculation)**

Where, W_i and L_i represent the preset weight and individual score of *i-th* label. The values for L_i are set in round one, where L_i is 0 is for legitimate pages and L_i is 1 for phishing pages as explained in section 3.4.2.1.4.1. The value of risk rating obtained is between 0 and 1. A web page is considered as a phishing threat if the risk rating value, as defined in equation 3.1, exceeds a certain threshold δ, i.e.: $\frac{\sum_{i=1}^{i=n}[(w_i)*(L_i)]}{\sum_{i=1}^{i=n}(w_i)} > \delta$.

This equation is adopted from a journal on detecting phishing pages by Rosiello et al., [52]. It is used to compute risk rating because when certain features are present, the probability of a webpage being phishing increases. The risk rating computed is multiplied by 100 to get risk rating percentage.

Table 3.6 summarises the labels used by PhishProof in Level 3.

Table 3.6: Level 3 labels summary.

Label	What the label means		Label Score
Age	When the domain was registered/activated	If age of domain is more than "Minimum age limit" value set by PhishProof	Low Risk (0)
		If age of domain is less than "Minimum age limit" value set by PhishProof	High Risk (1)
Referrer	From where users were redirected to the current website	if referrer is any website not in blacklist and not an email provider	Low Risk (0)
		If referrer is any email service provider **(Table 3.3)**	High Risk (1)
Country	Country where the website is hosted	If the hosting country does not have history of hosting phishing websites	Low Risk (0)
		If the hosting country has history of hosting phishing websites **(Table 3.4)**	High Risk (1)
Hosting	Name of Organization hosting the current site	If the hosting company has no history of hosting phishing websites	Low Risk (0)
		If the hosting company has previously hosted phishing websites **(Table 3.5)**	High Risk (1)
Password	Checks whether input field of type password exists or not	If current page contains no input field of type password	Low Risk (0)
		If current page contains input field of type password	High Risk (1)
Encryption	checks whether current page uses encryption to transfer user information or not	If current page uses SSL authentication protocol	Low Risk (0)
		If current page does not use authentication SSL protocol	High Risk (1)

3.4.2.2 PhishProof Website

PhishProof websites is hosted on a shared server and is connected to the central database which contains blacklist and whitelist. This component helps in improving PhishProof toolbar and provides users with option to report phishing URLs, incorrectly blocked URLs and any Bug in the system. PhishProof Website also allows users to download Firefox extension package and provides instructions for installing and using the toolbar. Website consists of 4 tabs; Home, About Us, Report and Contact Us.

The website header is common for all pages, which contains PhishProof toolbar logo and a navigation bar. The navigation bar contains links to all four tabs and a “Download Now!” link which allows users to download PhishProof toolbar installation package.

3.4.2.2.1 Home

This is the landing tab for PhishProof website. It contains a small tutorial to help users understand and use anti-phishing toolbar. Home tab also contains three report buttons which direct users to different reporting sections on website.

Tutorial consists of a screenshot of Firefox with PhishProof toolbar installed on it and some text explaining what different things on the toolbar mean. The tutorial also gives users an understanding of how they can get the most out of PhishProof website and toolbar.

Home tab also contains 3 buttons; A phishing site, Blocked URL and A BUG. Clicking on any button will direct the user to other sections of the website where users can report accordingly (see section 3.4.2.2.3).

3.4.2.2.2 About Us

"About Us" helps users understand what PhishProof is all about and what the main purpose behind this solution is. It also contains information about people associated with PhishProof and what responsibilities they have.

3.4.2.2.3 Report

This is the most important part of this Component. This section allows users to report URLs they consider as potential phishing websites or URLs they think have been blocked incorrectly. This section also allows users to report any bug or issue faced while installing or using PhishProof toolbar. These reports can be made through 3 sub tabs; A Phishing website, incorrectly blocked URL and A BUG. Each sub tab allows users to report about that particular issue by filling a short form. The first two sub tabs have a similar form with following fields:

- Name
- Email Address
- URL to report
- Description
- Code

All 5 fields are mandatory and no form will be submitted if either field is missing. Names of first four fields are self explanatory, whereas the 5^{th} field requires users to input the value displayed in CAPTCHA. CAPTCHA is a challenge response check used

to ensure that the form is being submitted by a person and not any computer program [40]. This also protects PhishProof Server against Denial of Service (DoS) attack [41].

Report a Bug form has all fields described above except "URL to report". All information entered by users is stored in PhishProof central server and can be managed from backend administrator panel (see section 3.4.2.3).

3.4.2.2.4 Contact Us

"Contact Us" allows users to contact PhishProof team regarding any queries or help. It contains a small form with the following fields:

- Name
- Email
- Phone
- Comments
- Code

All 5 fields are mandatory and need to be filled by users. When users submit this form, an email is generated and sent to PhishProof support services email address.

3.4.2.3 PhishProof Backend Administrator Panel

This component manages blacklist, whitelist and all incoming reports & queries from PhishProof website. Like PhishProof website this is also hosted on a shared server. Backend admin panel is password protected because of its sensitive nature. Anyone with unauthorized access to backend admin panel will be able to alter blacklists & whitelist and harm PhishProof and its user. PhishProof admin panel has only one account and no other user except administrator will be able to login. PhishProof admin panel also has functionality to change admin panel password. Administrator can change existing password once logged in by clicking on "change password". Administrator will be asked to type new password twice to make sure there is no typo and new password has been typed correctly.

Once logged in administrator can manage PhishProof toolbar and Website through 4 tabs: URL reports, Blacklist, Whitelist and Reported Bugs.

3.4.2.3.1 URL Reports

This section contains reports made by users on PhishProof website. URL reports section is divided into two parts; first part contains potential phishing websites reported by users while the second one contains incorrectly blocked URLs reported. For the first part admin panel provides administrator with the option to add any reported item into blacklist or cancel user's suggestion. Second part gives administrator option to remove suggested items from blacklist or cancel user queries.

3.4.2.3.2 Blacklist

Blacklist contains all websites considered as potential threat by PhishProof (see section 3.4.2.1.2.2). PhishProof blacklist can be managed by administrator using this section. Administrator can perform the following operations on any record in blacklist:

- Add
- Edit
- Delete
- Add to whitelist

3.4.2.3.3 Whitelist

Whitelist contains all websites trusted by PhishProof (see section 3.4.2.1.2.1). This section gives administrator the option to manage whitelist and perform following operations:

- Add
- Edit
- Delete
- Add to Blacklist

3.4.2.3.4 Reported Bugs

This section contains all issues reported by users using the report bugs tab on PhishProof website. Administrator can check issues faced by users through this tab. This section helps administrator and PhishProof team constantly improve PhishProof toolbar and user experience.

3.5 Chapter Summary

In this chapter the requirements of proposed solution have been described along with the design of PhishProof. This chapter explains how the system requirements are met by PhishProof design. In addition, three components of proposed anti-phishing solution, PhishProof toolbar, PhishProof website and PhishProof Backend administrator panel, have also been explained. 3–level protection against phishing has been introduced to protect users against phishing attacks. First level uses blacklist based method to check the webpage loaded against a whitelist and blacklist. These lists are maintained from backend admin panel. Second level uses a combination of blacklist based method and webpage content analysis method to protect users against potential phishing websites. Web page features are extracted and compared against the blacklist to check if the page was refereed from a blacklisted website or the current page contains any blacklisted links. If the website survives first two levels, level 3 uses webpage content analysis to classify the current page as either phishing or legitimate. Level 3 consists of 2 rounds; first round analyses 6 features and assigns individual score to each label. While in second round all individual scores are joined along with their pre-defined weights to compute risk rating. The proposed solution also explains how alert messages and audio warnings are used to get user's attention. Apart from PhishProof toolbar, PhishProof website and backend admin panel architecture is also explained in this chapter. In the next chapter, implementation and issues faced while implementation of proposed PhishProof system will be discussed.

Chapter 4 PhishProof Implementation

4.1 Chapter Introduction

This chapter focuses on implementation of the PhishProof system design explained in chapter 3. It describes the platform for which the toolbar is developed and different languages used to implement the proposed solution. In addition, the difficulties and issues faced during implementation are also discussed in this chapter along with the code generation of the PhishProof toolbar.

This chapter is structured as follows: In section 4.2 implementation platforms are discussed. Section 4.3 describes the programming languages used for implementing proposed anti-phishing solution. Section 4.4 presents the low level design of the PhishProof system. Section 4.5 illustrates the graphical user interface for all three components. Section 4.6 highlights difficulties and issues faced while implementation. Implementation chapter is concluded by summarising entire chapter in section 4.7.

4.2 Implementation Platforms

Firefox (version 12 and above) is chosen as the platform to host PhishProof toolbar because it is one of the most widely used internet browser available [16]. To integrate PhishProof toolbar into Firefox browser, a Firefox extension will be developed. Firefox extensions can be updated easily to work with future Firefox versions and offer developers extensive support in analysing features of the webpage. Firefox extensions give developers freedom to customise application according to personal needs. Through extensions, developers can easily create network connections and processes. Firefox extensions are required to follow certain standards which are explained in detail in section 4.4.1.

4.3 Programming Languages

Proposed anti-phishing solution consists of 3 components; the PhishProof toolbar, the PhishProof website and the PhishProof admin panel (see section 3.4.2 for details). To implement these components 5 languages; XUL, CSS, JavaScript, HTML and PHP, are required. A Firefox extension can only be developed using XUL, CSS and JavaScript. HTML and PHP are used to implement user interface and backend coding for PhishProof website and PhishProof admin panel.

XML–based User–interface Language (XUL) is a type of XML that makes use of predefined widgets. XUL is used for developing extension's interface. It defines logic rather than style [45]. The best part of XUL is the use of dynamic overlays. Through dynamic overlays, developers don't have to worry about changing original code. Developers can simply modify the behaviour of a window's interface by adding that specific overlay. This helps developers focus on current task in hand rather than reinventing the wheel.

Cascading Style Sheets (CSS) is used to describe presentation of HTML, XUL or any other markup language. The style sheet has two major advantages; its style capabilities are more than HTML and XUL and it can be modified easily. To modify style using a style sheet, developer only needs to make a change in the style sheet file, whereas in XUL or HTML each line of code using particular style needs to be modified.

JavaScript is a scripting language used to provide enhanced user interfaces. It supports functional, object oriented and imperative programming. It gives access to objects within a host environment. It is dynamic and has first-class functions, which means that all functions are objects. Unlike C and C++, JavaScript has garbage collector that automatically frees up memory. It detects when an object is not needed anymore and removes it. Based on these reasons the PhishProof toolbar functionality is implemented using JavaScript.

4.4 PhishProof System – Low Level Design

PhishProof toolbar is a Firefox (version 12 and above) extension. An extension is a type of add-on which allows users to add new features to an application or modify existing ones. Extensions are installed in Firefox browser and initiated when browser starts. An extension can also utilize same memory as browser. Browser events such as a page loaded or new tab opened can be monitored through an extension.

The first and foremost step of developing an extension for Firefox is to set up the environment for programming. Firefox extensions require specific file structure and file names, which are explained in the following section.

4.4.1 Setting up Firefox Extension

All Firefox extensions are required to maintain a specific file structure [44]. The top level folder must always be the name of extension, which in this case is "PhishProof".

After this a folder named “chrome” is created inside the main folder. Chrome folder contains two sub folders named “content” and “skin”. All letters should be lowercase for sub folders; chrome, content and skin. The internal file structure for PhishProof toolbar is shown is figure 4.1.

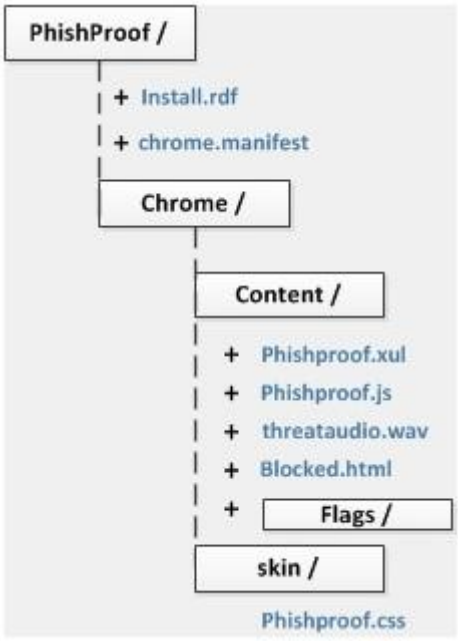


Figure 4.1: Internal File Structure of PhishProof Toolbar.

The main folder, PhishProof, contains two files “install.rdf” and “chrome.manifest”. The install.rdf file is used to register the extension at specific location by the Extension Manager. This file contains the following information about extension: extension ID, extension name, version number, minimum and maximum versions of Firefox on which extension works, creator details, description of extension, etc. Some tags such as id, name, type etc. are compulsory and extension won’t be installed on users system if they are missing or not declared correctly. Tags such as creator, description, homepageURL etc. that provide extra information to users can be omitted. Figure 4.2 shows the install.rdf file for PhishProof. All XML tags are shown in blue colour for readers ease, whereas values and comments are shown in black and green.

```
<?xml version="1.0"?>

<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:em="http://www.mozilla.org/2004/em-rdf#">

    <Description about="urn:mozilla:install-manifest">

      <!-- Required Items -->
      <em:id>taimoor@phishproof.com</em:id>
      <em:name>PhishProof Toolbar</em:name>
      <em:type>2</em:type>
      <em:version>1.0</em:version>

      <em:targetApplication>
            <Description>
                  <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
                  <em:minVersion>4.0</em:minVersion>
                  <em:maxVersion>15.*</em:maxVersion>
            </Description>
      </em:targetApplication>

      <!-- Optional Items -->
      <em:creator>PhishProof</em:creator>
      <em:iconURL>chrome://phishproof/skin/toolbar-logo.png</em:iconURL>
      <em:description>PhishProof is an anti phishing toolbar that detects phishing websites.
      </em:description>
      <em:homepageURL>http://www.phishproof.com/</em:homepageURL>

    </Description>
</RDF>
```

Figure 4.2: Install.rdf file of PhishProof Toolbar.

The chrome.manifest file tells Firefox what overlays and packages are provided by our extension. Figure 4.3 shows the chrome.manifest file of PhishProof toolbar. The first line uses package name specified (PhishProof) to register a content package and location to the content folder. Lines 2 & 3 register an overlay through which Firefox's main window can be modified using phishproof.xul. Line 5 creates a locale while line 7 sets up a default skin provider.

```
Line 1:  content PhishProof chrome/content/
Line 2:  overlay chrome://browser/content/browser.xul
Line 3:  chrome://phishproof/content/phishproof.xul
Line 4:
Line 5:  locale PhishProof en-US chrome/locale/en-US/
Line 6:
Line 7:  skin PhishProof classic/1.0 chrome/skin/
```

Figure 4.3: chrome.manifest file of PhishProof Toolbar.

Apart from install.rdf and chrome.manifest, PhishProof folder also contains a sub folder named chrome. This folder contains content and skin subfolders. Skin sub folder contains a CSS file which is used to style PhishProof toolbar. The content sub folder is the most important folder because it contains both, user interface and functionality files.

The “phishproof.xul” file is responsible for interface of PhishProof toolbar, whereas phishproof.js handles toolbar functionality. Phishproof.xul embeds JavaScript through the `<script>` tag. GUI of the PhishProof toolbar is explained in detail in section 4.5.1. The following section describes implementation of PhishProof toolbar functionality discussed in section 3.4.

4.4.2 PhishProof Toolbar Functionality

The “phishproof.js” file implements all anti-phishing operations and checks described in section 3.4.2.1. The anti-phishing checks are implemented as follows:

PhishProof toolbar uses event listeners to check whenever a page is loaded in browser. Listeners are objects that listen and are triggered when a particular event happens. A single listener only registers events on a single target, which can be a window, document or XMLHttpRequest. The PhishProof toolbar uses `addEventListener` method to register an event for `window` using the following statement:

```
window.addEventListener(eventType, listener)
```

Where, `eventType` is a string variable that represents which event to listen to. In this case its value is set to “load”. `Listener` is the function triggered when the specified event occurs. PhishProof uses this to initialize toolbar whenever Firefox is opened. This event is only fired once, when toolbar is loaded, and is removed immediately since it is not needed any more. The event is removed using `removeEventListener` method.

When PhishProof toolbar is initialized it declares global variables and loads whitelist, blacklist, mail list, country list and hosting company lists into browser memory (see section 3.4.2.1.2).

4.4.2.1 Loading Lists

Mail list, country list and hosting company list are part of the installation package. They are initialized inside the code and added like a normal array. Whitelist and blacklist is maintained using PhishProof backend administrator panel. These lists are retrieved from PhishProof server each time a browser session is initiated. XMLHttpRequest is used to connect to PhishProof server. XMLHttpRequest is an Application Programming Interface (API) available in JavaScript and other browser scripting languages. This API not only sends HTTPS and HTTP requests to a server, but also loads the response directly into the script [48]. XMLHttpRequest object allows data from a server to be

received in background and does not affect the processes running in foreground. Hence, users can use browser normally while PhishProof loads whitelist and blacklist.

Almost all modern browsers support XMLHttpRequest objects [49]. Whether a browser supports XMLHttpRequest objects or not can be tested using `Window.XMLHttpRequest` command. If this command returns true if the browser supports XMLHttpRequest objects, else it returns false. For Firefox and other browsers that support XMLHttpRequest objects, an XMLHttpRequest object can be created using the following command:

```
var variableName = new XMLHttpRequest ()
```

XMLHttpRequest objects have 3 important properties; `Onreadystatechange`, `readyState` and `status`. `Onreadystatechange` stores a function which is called whenever `readyState` property changes. `readyState` holds the status of XMLHttpRequest. When the request is finished and response is ready, the values of `readyState` and `status` are 4 and 200 (see table 4.1). PhishProof checks the function in `Onreadystatechange` for theses values and once these values are found, response from the server is retrieved as a string using `responseText` property. This is stored in a variable and converted into an array using `eval` command. XMLHttpRequest object uses `open` method to assign method and destination URL to pending requests. The Method assigned for loading lists is "`Get`" and destination URL is the URL of whitelist or blacklist on PhishProof server. Once these values are assigned, `send` method is used to send an HTTP request to server and receive response. Figure 4.4 sows the pseudocode for loading whitelist and blacklist.

Table 4.1: readyState and status values for XMLHttpRequest objects.

Property	Value	Description
readyState	0	Request not initialized
	1	Connection established
	2	request received
	3	processing request
	4	request finished and response is ready
status	200	OK
	404	Page not found

```
IF XMLHttpRequest object is Supported by browser
        create new XMLHttpRequest object
EndIF

BeginFunction when XMLHttpRequest object changes state

        IF object readyState is 4 AND object status is 200
                GET response from server
                CONVERT response from server into Array

        EndIF
EndFunction

ASSIGN method and destination URL to pending requests
SEND Http request
```

Figure 4.4: Pseudocode for loading whitelist and blacklist.

4.4.2.2 Phishing Checks

PhishProof toolbar performs phishing checks when a page is loaded into browser. The following command is used to check when a page is loaded in browser:

```
gBrowser.addEventListener ("DOMContentLoaded", Check Function)
```

gBrowser is a global object that exists for all browser windows. The `addEventListener` is attached to it so that phishing checks can be initiated whenever a web page is opened in browser. The first argument is event for which check functions will be initiated. `DOMContentLoaded` is used instead of `load` which was used previously because it is fired when parsing of page is completed. Whereas `load` event is fired when all files have been loaded, this also includes images and ads. The second argument is function which will be initiated when page parsing is completed. This function contains all phishing checks.

4.4.2.2.1 Level 1 Checks

The PhishProof toolbar uses blacklist based method as first line of defence against phishing attacks (see section 3.4.2.1.2). A variable `Flag` is used to check status of current web page. Flag value determines whether page is white-listed, blacklisted or undefined. The default value for `Flag` is 0 (see table 4.2).

Table 4.2: Possible Flag values.

Flag Value	Description
0	undefined
1	Blacklisted
2	White-listed

PhishProof starts level 1 phishing checks by extracting domain name of web page loaded in browser using `document.domain` command. This value is compared against whitelist. If this value is found in whitelist, all operations are stopped and `Flag` value is set to 2. If domain name is not found in whitelist, blacklist is checked for domain name and URL of current web page. If either domain name or URL is found in blacklist, `Flag` is set to 1 and all operations are stopped. Figure 4.5 shows the pseudocode for level 1. If no match is found either of the lists, PhishProof proceeds to level 2.

```
//Whitelist check
Get domain of web page in browser
For each domain in whitelist
        IF website domain is found in whitelist
                SET Flag = 2
                Stop all processes
        EndIF
EndFor
//Blacklist Check
IF Flag is 0
        Get URL of web page in browser
        For each domain and URL in blacklist
                IF website domain OR website URL is found in blacklist
                        SET Flag = 1
                        Stop all processes
                EndIF
        EndFor
EndIF
```

Figure 4.5: Pseudocode for Level 1 Checks.

4.4.2.2.2 Level 2 Checks

The PhishProof toolbar uses a combination of blacklist and web page content analysis method in level 2 (see section 3.4.2.1.3). In level 2 PhishProof toolbar performs referrer check and links check to protect users against phishing. Referrer check is executed only if web page has a referrer. This is checked using `document.referrer.length` command which returns length of referrer. If length of referrer is 0, it means web page

in browser does not have a referrer. If referrer for web page is available, referrer of current webpage is extracted using the `document.referrer` command. This command returns the URL of referrer. A parse URI (Uniform Resource Identifier) open source function [50] is used to extract domain name from URL. This is done by removing all characters and variables before and after the domain name. Referrer domain is then compared against blacklist to check whether referrer is blacklisted or not. If referrer is blacklisted, all processes are stopped and `flag` set to 1.

Links check examines all links available on a web page. This check is executed only if the webpage has links to other pages. This is checked using the `document.links.length` command which returns length of an object containing all links. This value is equal to number of links available on a web page. If length of object is 0, it means web page in browser does not have any links. If length of object is greater than 0, links check is implemented by comparing URL of each link on web page with blacklist. The `document.links[i].href` command is used to access URL of each link, where `"i"` is index number from 0 to length of object. If any link is found in blacklist, all processes are stopped and `flag` is set to 1. Figure 4.6 shows the pseudocode for level 2 checks. If web page passes level 2 phishing checks, PhishProof proceeds to level 3.

```
//Referrer check
IF Referrer for web page exists AND Flag is 0
        Get referrer URL of web page in browser
        Get domain name of referrer from referrer URL
        For each domain in blacklist
                IF referrer domain is found in blacklist
                        SET Flag = 1
                        Stop all processes
                EndIF
        EndFor
EndIF
//Links Check
IF Web page has links AND Flag is 0
        Get number of links on web page
        For each link on web page
                For each domain and URL in blacklist
                        IF URL of Link is found in blacklist
                                SET Flag = 1
                                Stop all processes
                        EndIF
                EndFor
        EndFor
EndIF
```

Figure 4.6: Pseudocode for level 2 checks.

4.4.2.2.3 Level 3 Checks

The PhishProof toolbar uses web page content analysis to counter phishing attacks in level 3. Different labels are evaluated and assigned scores individually in level 3. These labels are then combined together with pre-defined weights to compute risk rating percentage. The checks described in section 3.4.2.1.4 are implemented as follows:

Referrer Label

Since most phishing attacks are initiated from email message, PhishProof checks whether referrer to a current page is an email provider or not. Referrer label check is executed only if web page in browser has a referrer. This is checked using length method of referrer object as explained earlier. URL and domain name of referrer website are also extracted using same commands as described in previous section. The domain name is then compared against a mail list. Mail list contains domain names of all leading email service providers. This list is loaded into browser memory at start of each browser session. If referrer domain exists in mail list, referrer label score is set to 1. Figure 4.7 shows the pseudocode for referrer label.

```
//Referrer Label
IF Referrer for web page exists AND Flag is 0

        Get referrer URL of web page in browser
        Get domain name of referrer from referrer URL

        For each domain in Mail List
                IF referrer domain is found in Mail List
                        SET Referrer Label Score
                EndIF
        EndFor
EndIF
```

Figure 4.7: Pseudocode for Referrer Label.

Password & Encryption Label

Password label checks whether a web page has any input field of type password or not. Since all phishing attacks are aimed at getting users confidential information, most phishing pages have at least one input field of type password. PhishProof toolbar gets all input fields of a web page by using the `getElementsByTagName ()` method. This method takes tag name as argument and returns all elements with that tag name. The following command returns an object that contains all elements of tag name “input”:

`document.getElementsByTagName ("input")`

The number of input fields available on a web page can be determined by using the `object.length` method. The `object[i].type` command is used to get type of each element, where `"i"` is index number from 0 to length of object. If any element of type password is found, password label score is set to 1.

Encryption label checks whether a web page uses encryption to transfer sensitive user information or not. This label is only evaluated if password label is present. This is done because most legitimate website also avoid authentication if no sensitive data is involved. Encryption label checks whether a web page has SSL certificate installed or not using the `document.location.protocol` command. This command returns the protocol of current page. Encryption score is set to 1 if "Https" is returned. Figure 4.8 shows the pseudocode for password and encryption labels.

```
//Password Label
IF Flag is 0

        Get All elements with tag name = "input"
        For each element with tag name = "input"
                IF element type is "password"
                        SET Password Label Score
                EndIF
        EndFor
EndIF

//Encryption Label
IF Flag is 0

        Get Protocol of current Page
        IF protocol of current page is "Https:"
                SET Encryption Label Score
        EndIF
EndIF
```

Figure 4.8: Pseudocode for password & encryption label.

Age, Country & Hosting Label

Age label is evaluated on the basis of age of domain, whereas country and hosting labels are computed on the basis of country and company hosting the web page. The `WHOIS` query is used to compute these three labels. It is a protocol used for querying

databases that contain information about domain names and IP addresses registered. Table 4.3 lists some of the information provided by the WHOIS.

4.3: Information retrieved using WHOIS.

WHOIS information
Domain Name
Details of Organization through which domain was registered
Details of user who has registered the domain
Date on which domain expires
Date on which domain was activated
Date on which domain was last updated

The PhishProof toolbar implements the WHOIS using an open source PHP script [51] hosted on the PhishProof server. The script is accessed using an XMLHttpRequest API. This API is responsible for sending request to server and receiving the result generated. The script on PhishProof server computes date on which domain was activated, number of days since domain was activated, country hosting the domain and company providing hosting. The details computed on PhishProof server and returned to PhishProof toolbar as an object.

XMLHttpRequest object has an `open()` method which is used to assign method and destination URL to pending requests. The destination URL provided has two parts; location of the WHOIS script on server and domain name. The following string concatenation command is used to join both parts and create the destination URL.

Location of script: http://socialightning.com/whois/whois_domain.php

Domain name: `document.domain.replace ("www.",'')`

Destination URL: `location of script + "?domain=" + domain name`

The PhishProof toolbar uses `replace ()` method to remove **"www."** from the domain name because the WHOIS script used requires domain name only. The **"+"** sign in destination URL is used to join domain name and location of script. **"?domain="** part in destination URL informs the WHOIS script that value after **"="** is the domain name. The XMLHttpRequest object then sends a request using `send ()` method. The script on server extracts domain name from destination URL and performs the WHOIS operations. The script gets domain activation date; which is used to calculate the age of domain, and other details listed in table 4.3. The `gethostname ()` method is a PHP

method used to get IP address of server hosting domain. Country and hosting company details are retrieved by using the `WHOIS` script again but this time with IP address of server instead of domain name. This returns details of server from which country and company details are extracted.

The values returned by the `WHOIS` script are used to compute age, country and hosting company labels. Age label is computed by getting the number of days between current date and domain activation date. If number of days is less than threshold value, Age label score is set to 1.

Country label is computed by comparing hosting country against country list. This list includes all countries considered as potential threat by PhishProof (see section 3.4.2.1.4). If the hosting country is present in country list, country label score is set to 1.

Hosting label is computed by comparing hosting company against hosting company list. This includes all hosting companies considered as potential threat by PhishProof (see section 3.4.2.1.4). If the company hosting domain is present in hosting company list, hosting label score is set to 1.

Figure 4.9 shows the pseudocode for connecting to the PhishProof server and getting the response for `WHOIS`. Whereas, figure 4.10 shows the pseudocode for Age, Country and Hosting Labels.

```
//Connecting to WHOIS script on server
IF XMLHttpRequest object is Supported by browser
        create new XMLHttpRequest object
EndIF

SET Destination URL = (script location + domain name)

BeginFunction when XMLHttpRequest object changes state

        IF object readyState is 4 AND object status is 200
                Get response of WHOIS from server
                CONVERT WHOIS response from server into Array
        EndIF
EndFunction

ASSIGN method and destination URL to pending requests
SEND Http request
```

Figure 4.9: Pseudocode for connecting to server and getting WHOIS response.

```
//Age Label
IF Flag is 0
        Get Domain activation date
        Get Current date
        SET Domain Age  = number of days between current date and domain activation date
        IF Domain Age is less than Threshold value
                SET Age Label Score
        EndIF
EndIF

//Country Label
IF Flag is 0
        Get Country hosting the domain
        For each country in country list
                IF country hosting the domain is found in country list
                        SET Country Label Score
                EndIF
        EndFor
EndIF

//Hosting Label
IF Flag is 0
        Get Company Hosting  the domain
        For each company in hosting company list
                IF company hosting the domain is found in hosting company list
                        SET Hosting Label Score
                EndIF
        EndFor
EndIF
```

Figure 4.10: Pseudocode for Age, Country and Hosting Label.

4.5 Graphical User Interface

This section describes the user interface for all 3 components of the PhishProof system explained in section 3.4.

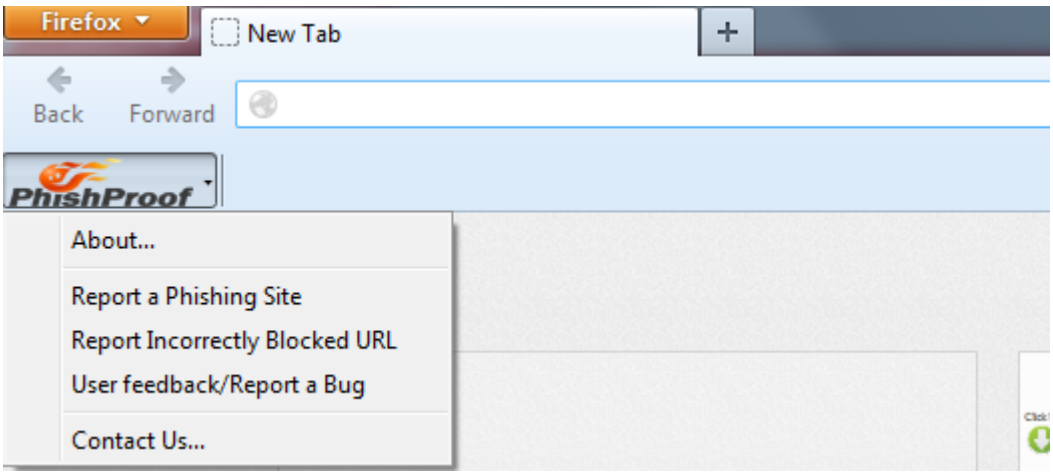
4.5.1 PhishProof Toolbar

After the PhishProof toolbar component is installed on Firefox, a toolbar appears on the main browser window. The toolbar has two states, idle and active.

Idle State

When a browser or new tab is opened and no page is loaded in browser window, the PhishProof toolbar is in its idle state. In idle state, the toolbar only contains a menu

button with PhishProof logo (see figure 4.1). The menu button has 5 menu items which redirect users to different sections in the PhishProof website (see section 3.4.2.2).



4.11: PhishProof toolbar idle state.

Active State

The PhishProof toolbar becomes active when a page is loaded in the browser window. PhishProof's active state has 4 components (see figure 4.2); PhishProof menu button, Risk Rating bar, since label and country label.

PhishProof's menu button contains menu items that redirect users to the PhishProof website (see section 3.4.2.2).

The risk rating bar shows the risk rating percentage for current page calculated by the PhishProof toolbar. The toolbar uses red colour to show risk percentage, whereas green colour is used to display rest of the bar. A combination of these two colours is used because red colour is good to display alerts & warning signs [43], and the contrast of red and green is very effective in getting users' attention as compared to other colours. The risk rating bar also displays the value of risk rating percentage calculated.

The "since label" displays the registration date of current page in the browser window. This is used to compute age of the website (see section 3.4.2.1.4.1).

The "country label" displays country initials and the flag of country hosting webpage in the browser window.

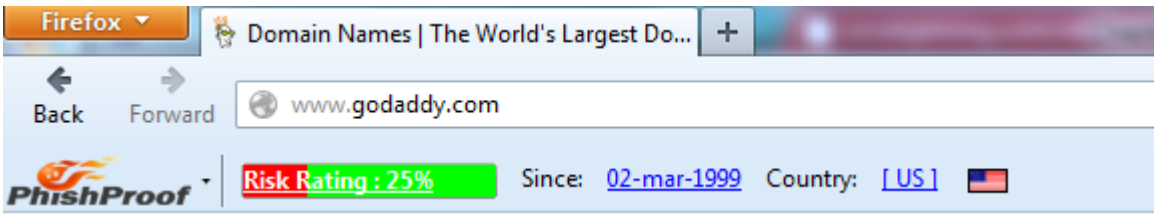


Figure 4.12: PhishProof Toolbar active state.

4.5.2 PhishProof website

The PhishProof website has 4 main tabs as described in section 3.4.2.2. The PhishProof website has been designed in a way that it can be easily used by any both expert and naïve users. The website has been implemented using HTML and PHP, and can be easily accessed on any browser. Figure 4.13 to 4.18 show different tabs of the PhishProof website.

Figure 4.13: Home tab of PhishProof website.

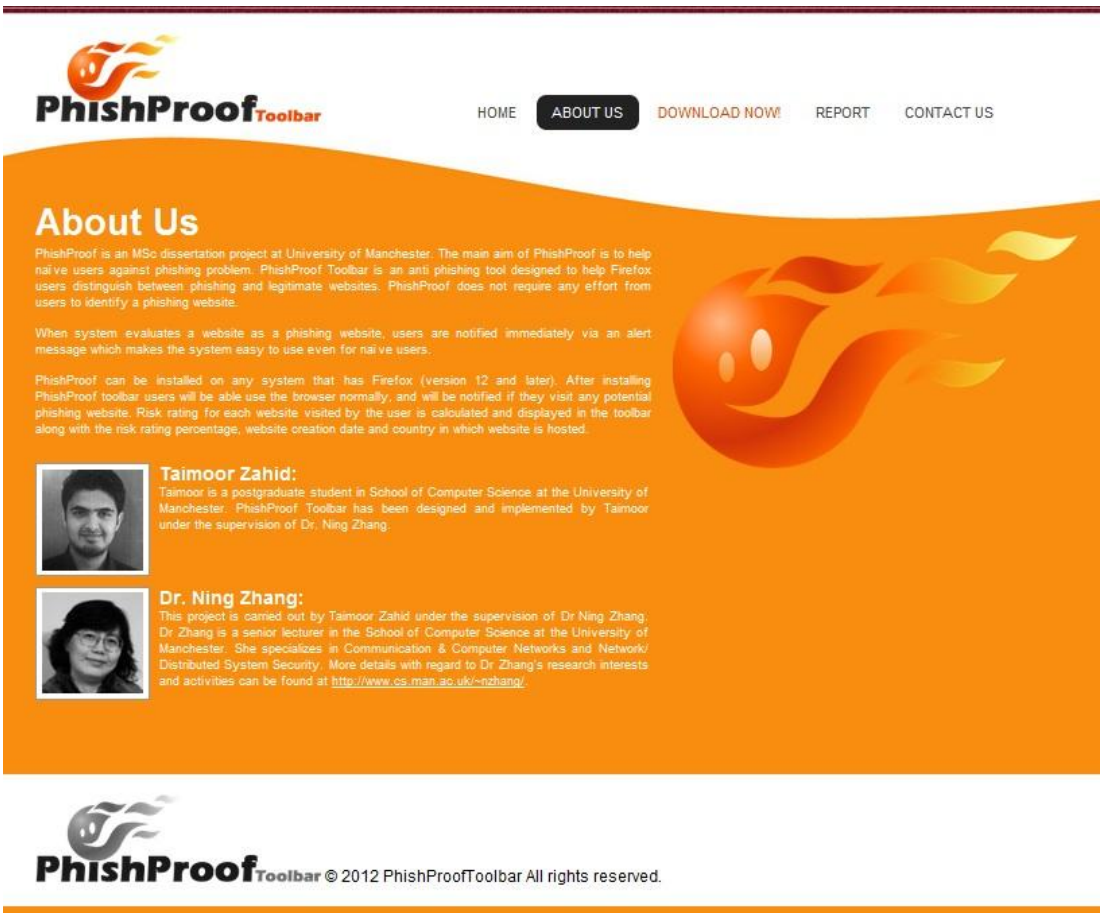


Figure 4.14: About us tab of PhishProof website.

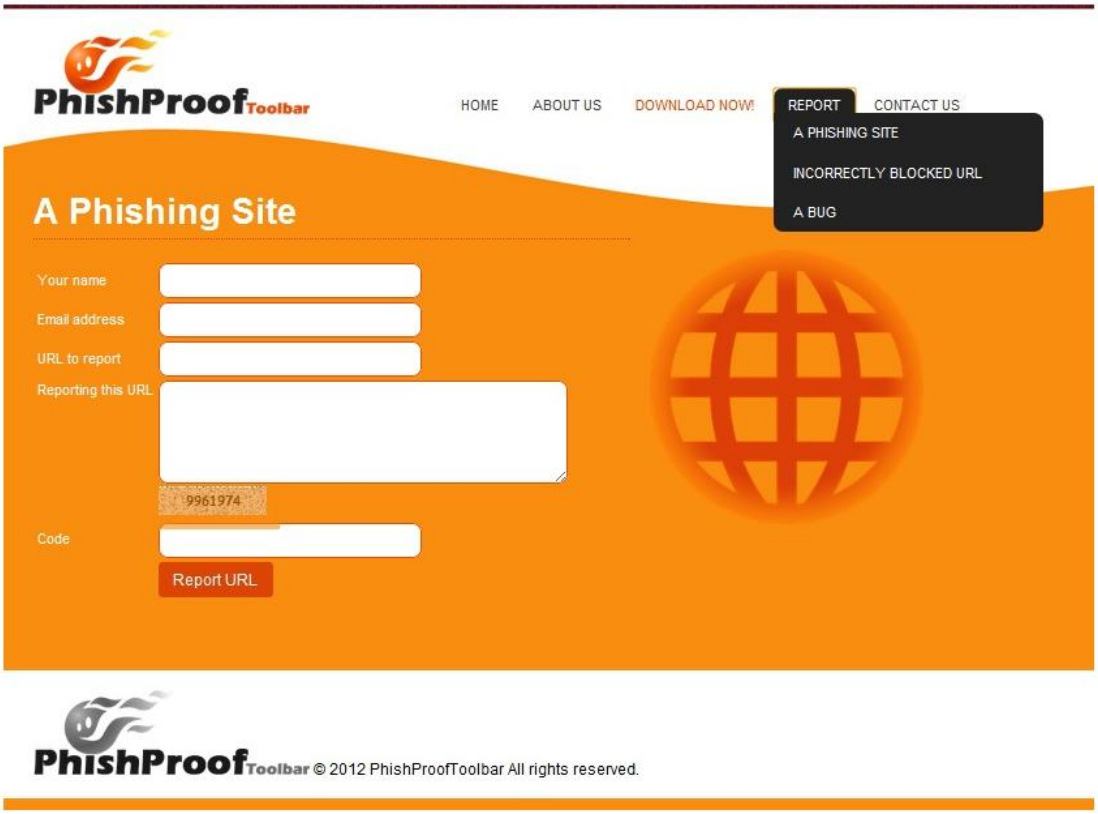


Figure 4.15: Report a phishing site page of PhishProof website.

Figure 4.16: Report an incorrectly blocked URL page of PhishProof website.

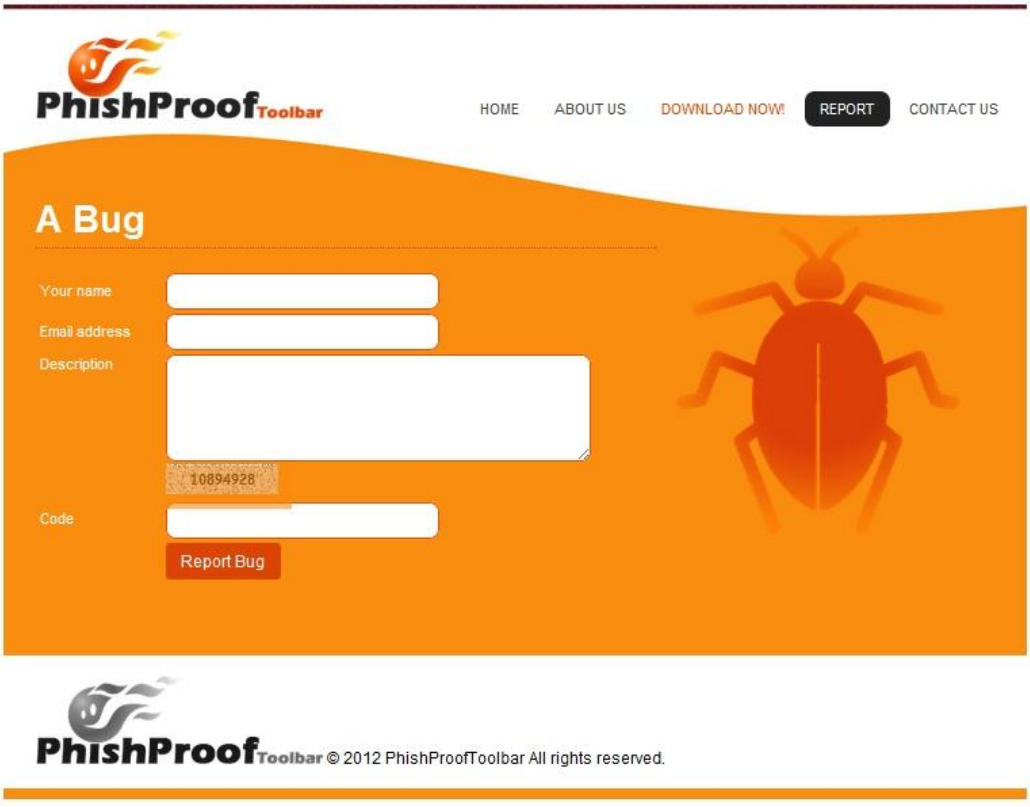


Figure 4.17: Report a Bug page of PhishProof website.

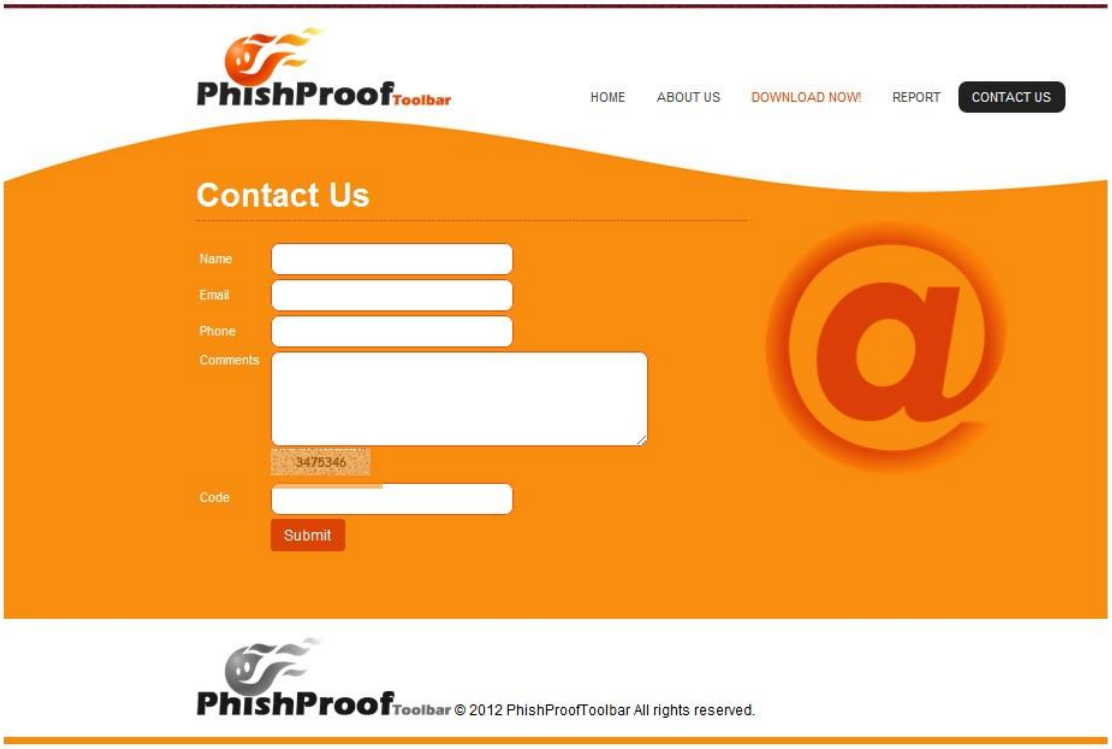


Figure 4.18: Contact us page of PhishProof website.

4.5.3 PhishProof Backend Admin Panel

The PhishProof backend admin panel is used to maintain blacklist and whitelist, and help improve the toolbar by inspecting reports made by PhishProof users (as explained in section 3.4.2.3). The PhishProof admin panel has been designed in a way that it can be easily used by the PhishProof administrator. Admin panel has been implemented using HTML and PHP, and can be easily accessed using any browser. Figure 4.19 to 4.26 show different screens of the PhishProof admin panel.

Figure 4.19: Login Page of PhishProof backend Admin Panel.

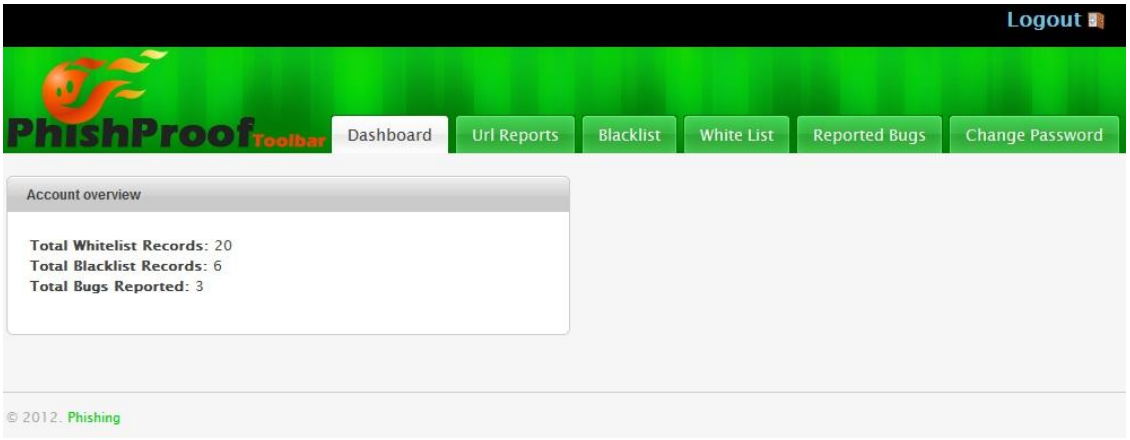


Figure 4.20: Admin panel dashboard.

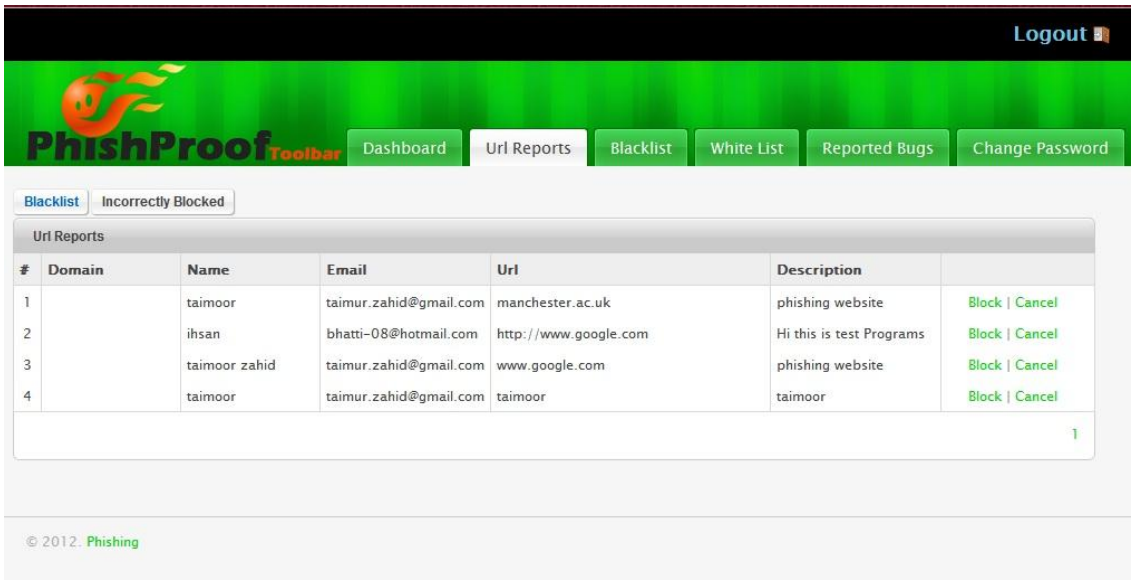


Figure 4.21: Potential phishing URLs reported by users.

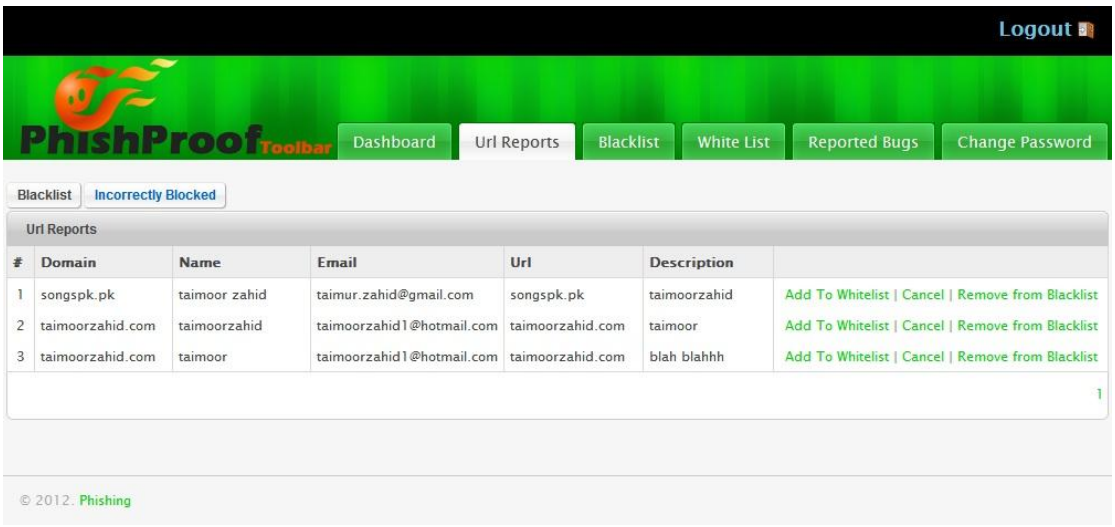


Figure 4.22: Incorrectly blocked URLs reported by users.

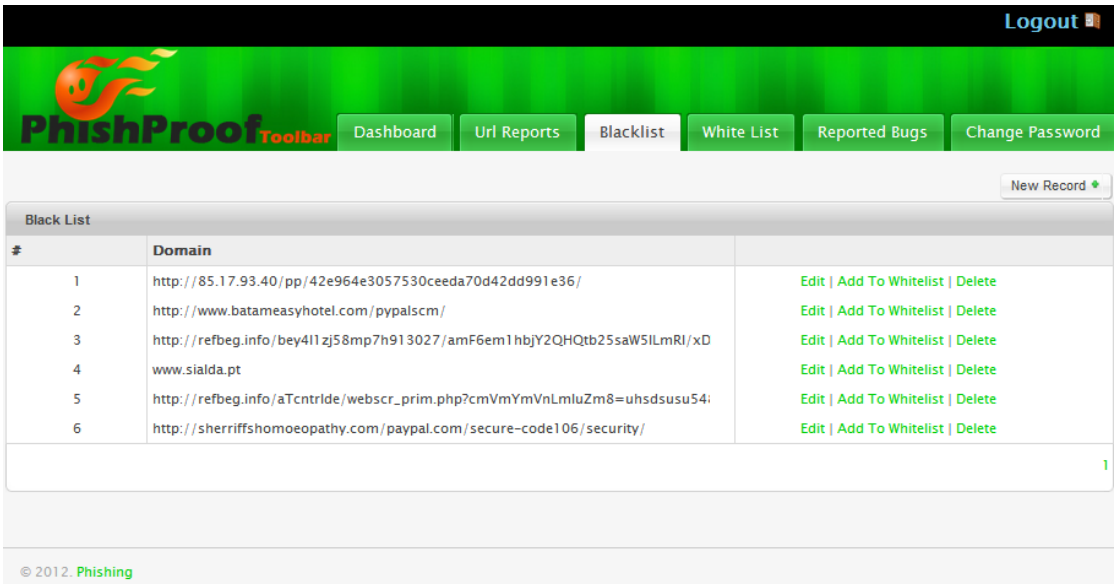


Figure 4.23: Manage Blacklist page of PhishProof backend admin panel.

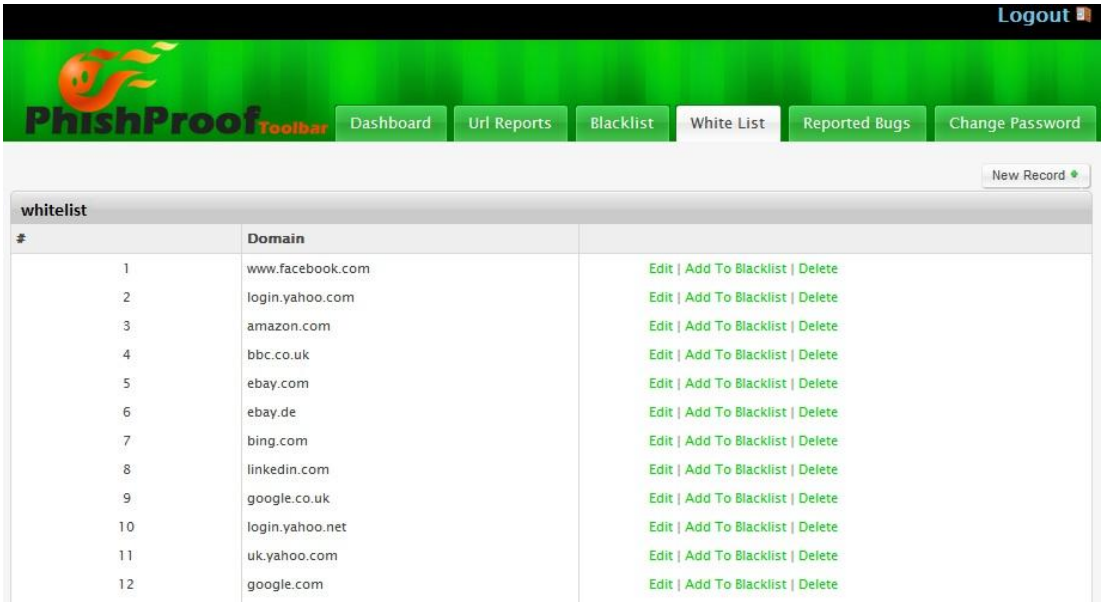


Figure 4.24: Manage Whitelist page of PhishProof backend admin panel.

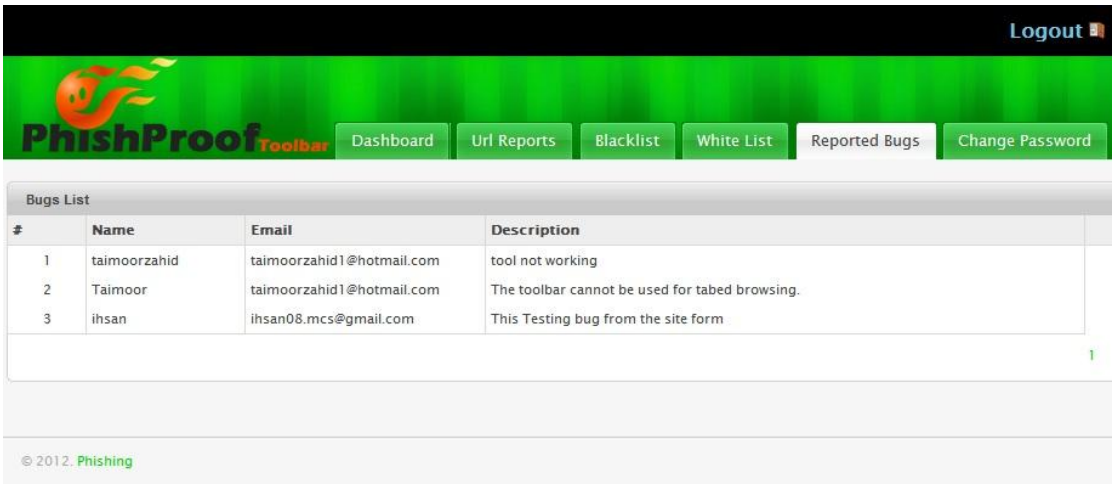


Figure 4.25: Reported Bugs page of PhishProof backend admin panel.

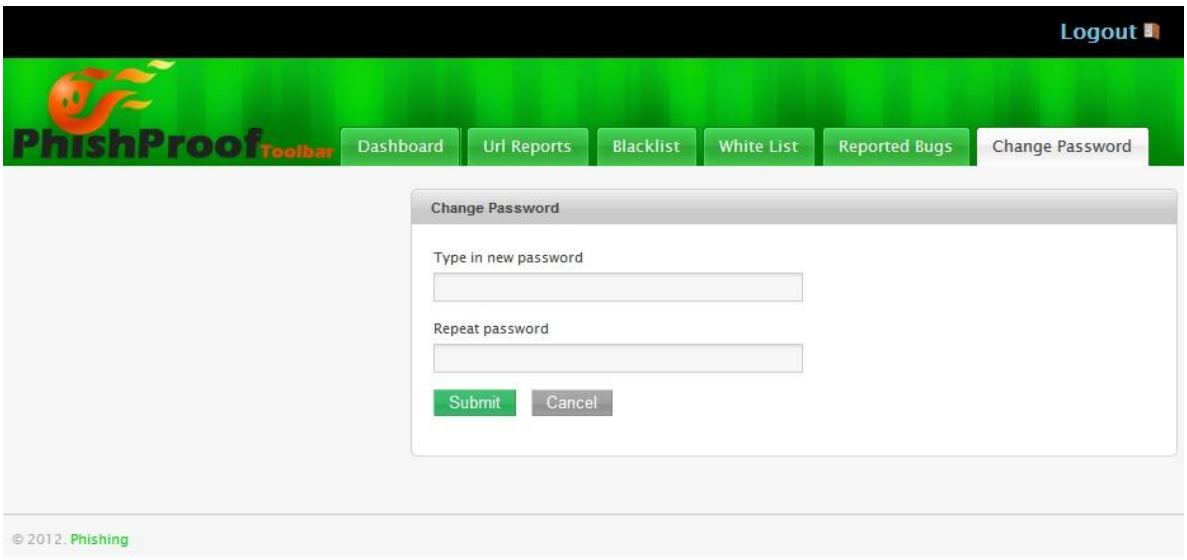


Figure 4.26: Change Password page of PhishProof backend admin panel.

4.6 Difficulties Faced During Implementation

Development of PhishProof involves overcoming some hurdles and solving some challenges. Issues faced during the implementing PhishProof are: limited phishing pages, large size of blacklist and query processing time.

Limited Phishing Pages

To provide a solution for phishing, it is very important to analyse existing phishing pages. However, phishing pages have a very short lifetime and are either taken down by anti-phishing organizations or removed by attackers themselves [3]. To overcome this hurdle, a detailed literature review of academic articles and analysis of existing security solutions has been conducted. Security blogs have also been referred to perceive how user failed to recognize a phishing website.

Large Size of Blacklist

The first two levels of protection offered by PhishProof use blacklist, which is stored on a shared server and is loaded in the browser memory for each browser session (as explained in section 3.4.2.1). Having a large blacklist affects Firefox's performance because the browser memory is used. To counter this issue, the size of blacklist is kept small and the admin panel is used to keep the blacklist up-to-date by removing phishing pages that have been taken down by anti-phishing organizations.

Query Processing Time

The `WHOIS` query is used to compute age, country and hosting labels. The results of this query are obtained from the PhishProof server using an XMLHttpRequest API (as explained in section 4.4.2.2.3). This API requires some time to connect to the server, process query and return result. Since JavaScript does not wait for results of an XMLHttpRequest object, the final risk rating displayed error because age, country and hosting scores were missing. To deal with this issue, the `setTimeout` method is used to delay processing of next command by 2.5 seconds. This gives the API enough time to connect to server and retrieve the results for `WHOIS`.

4.7 Chapter Summary

In this chapter the platform and languages used for implementing the PhishProof system have been explained. The Firefox browser has been chosen as the platform for proposed anti-phishing solution. XUL, CSS and JavaScript are used to implement the PhishProof toolbar. The pseudocode for all phishing checks has been explained in detail which includes all three levels of protection. The graphical user interface for the PhishProof toolbar, the PhishProof website and the PhishProof admin panel are also shown. Difficulties faced during implementation are also explained which include: limited phishing pages, query processing time and large size of blacklist.

Chapter 5 PhishProof Testing and Evaluation

5.1 Chapter Introduction

In this chapter, the PhishProof toolbar and its phishing checks are tested and evaluated. A dataset consisting of phishing and legitimate URLs is used to evaluate its performance, where phishing URLs are obtained from PhishTank. PhishProof is tested using two methods. First method tests the functionality of each phishing check individually. Second method involves two groups of users, where each group is assigned a task to perform using PhishProof. Users are then asked to fill a questionnaire and give their feedback. Limitations of the PhishProof system are also discussed in this chapter.

This chapter is structured as follows: section 5.2 describes two methods used for testing the proposed anti-phishing solution. Section 5.3 discusses limitations of PhishProof. Testing and evaluation chapter is concluded by summarising the chapter in section 5.4.

5.2 PhishProof Testing

Once the design is implemented, it is important to test the functionality of PhishProof and check if it fulfils the system requirements identified in section 3.3. PhishProof's functionality is tested using two different methods. First method involves testing phishing checks individually, whereas second method involves testing and evaluating the system as a whole using two groups of users. In level 1 checks (section 3.4.2.1.2), whitelist functionality is not tested and instead only blacklist functionality is tested because there is no alert message to notify users if whitelist check has been performed or not. In addition, they both are applied using the same technique and if one works, other will work automatically.

5.2.1 Testing Phishing Checks Individually

The main aim of PhishProof is to protect users against phishing attacks; therefore, it is tested using phishing websites. Phishing websites are not available easily as mentioned in section 4.6, therefore, some phishing websites have been created for testing purpose. These websites are hosted on www.uphero.com as sub domains. Blacklist, referrer and links check are tested using these websites, whereas websites available on PhishTank are used to test risk rating functionality.

Blacklist Check Testing and Result

This check examines the ability of PhishProof to restrict access to web pages that are already identified as phishing pages. To test blacklist functionality, a fake page has been created and hosted on the domain name www.gmailhelp.uphero.com. The domain hosting fake page is then added to PhishProof server using backend admin panel. When this URL is entered into Firefox address bar, PhishProof toolbar recognises the page as blacklisted and shows a warning message explaining the problem. Figure 5.1 shows the screenshot of warning message generated during blacklist check.

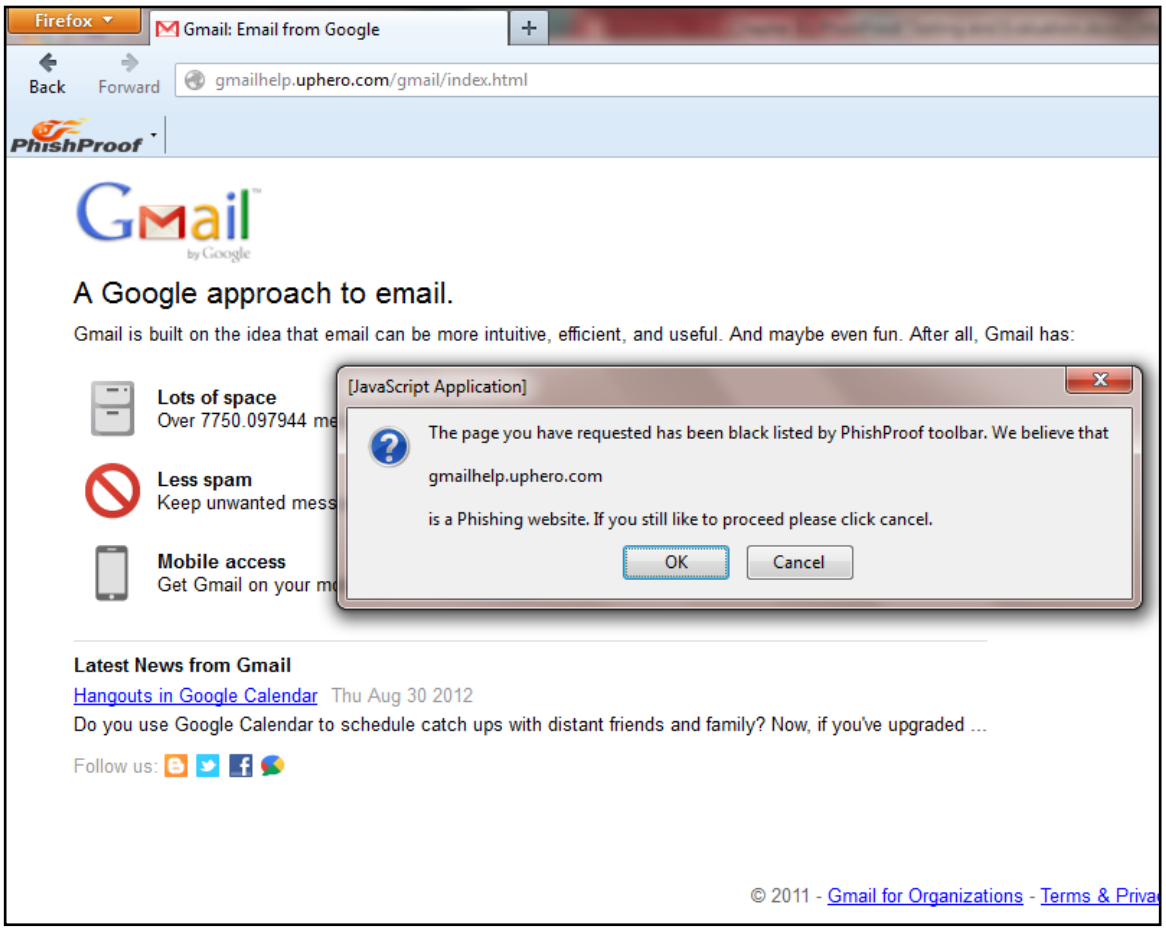


Figure 5.1: Blacklist domain check warning message.

If users click “OK”, they are redirected to a page which shows that page requested is blocked by PhishProof (see figure 5.2). If users ignore first warning and click “Cancel”, they are allowed to access the blacklisted page and second warning is initiated. The risk rating percentage is set to maximum and displayed in toolbar along with a message stating that current website is blacklisted (see figure 5.3). An audio alert is also played which informs the users that website is not trusted.

Figure 5.2: website blocked page.

Figure 5.3: Risk rating and alert message in toolbar when users ignore initial warning.

Referrer Check Testing and Result

This check examines the referrer of current page in browser window and checks whether the user is redirected from a blacklisted page or not (see section 3.4.2.1.3.1). To test referrer check functionality, a page is created and hosted on domain www.gmailhelp.uphero.com. This domain is then added to PhishProof blacklist. When this page is accessed through browser, PhishProof prompts about the blacklisted link through an alert message and toolbar notification as shown in figure 5.1 and 5.3. If both of these warning messages are ignored and any link from the blacklisted page is visited, an alert message is generated by referrer check as shown in figure 5.4.

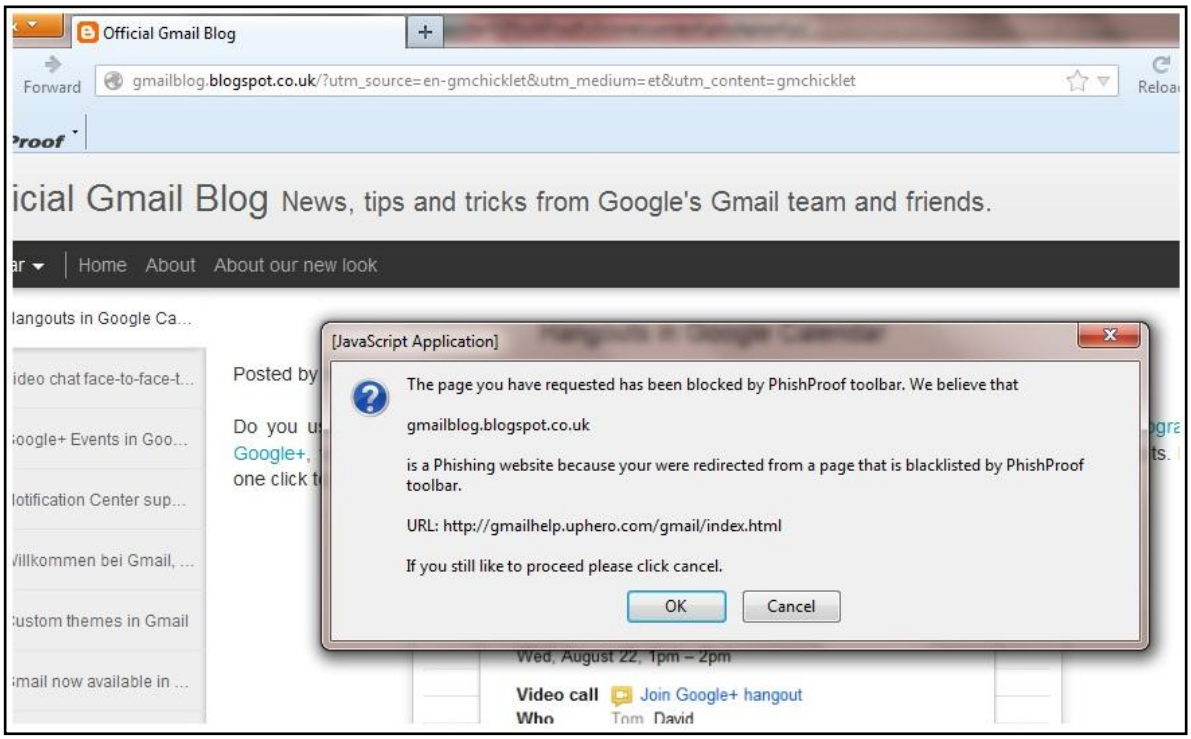


Figure 5.4: Referrer Check warning message.

Links Check Testing and Result

Links check scans all the links on a web page and compares them against PhishProof blacklist as explained in see section 3.4.2.1.3.2. To test the functionality of links check, a page is hosted on domain www.gmailhelp.uphero.com and some links are added on the page. The page contains one blacklisted link while all other links are legitimate. When the page is accessed using Firefox browser, PhishProof toolbar link check recognises the blacklisted link and displays a warning message. Figure 5.5 shows the warning message generated by PhishProof toolbar links check.

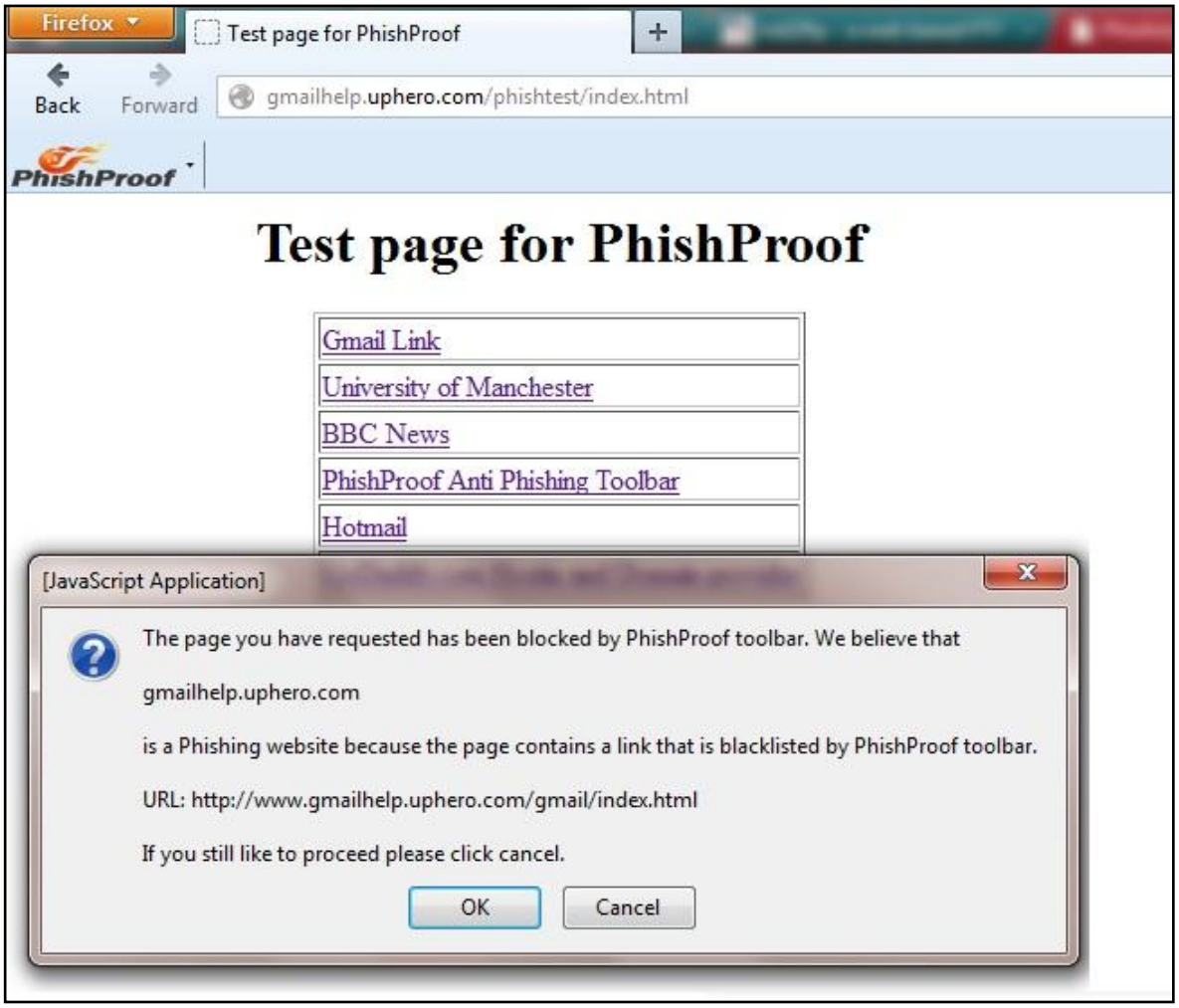
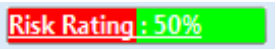
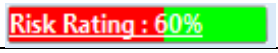
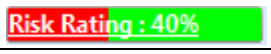
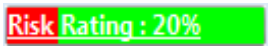


Figure 5.5: Links check warning message.

Risk Rating Functionality Testing and Result

If a web page survives individual tests without being flagged as a phishing page, it is evaluated by analysing its contents as described in section 3.4.2.1.4. To test risk rating functionality, a dataset consisting of 7 phishing and 3 legitimate websites is used. Where phishing websites are taken from www.phishtank.com. Each URL is accessed using a Firefox browser with PhishProof toolbar. None of the URLs is blacklisted to make sure that risk rating functionality is used to evaluate it. Table 5.1 below shows the result of risk rating functionality test.

Table 5.1: Results of risk rating functionality test.

URL	PhishProof Risk Rating %	PhishProof's Classification	Original Value
http://gmailhelp.uphero.com/gmail/index.html		Phishing	Phishing
http://gmailhelp.uphero.com/phishtest/index.html		Legitimate	Legitimate
http://www.h19.ru/denied.shtml?signeby024.h19.ru		Phishing	Phishing
http://www.h19.ru/denied.shtml?signeby026.h19.ru		Phishing	Phishing
http://sugarcrm101.com/wp-includes/js/tinymce/plugins/inlinepopups/skins/clearlooks2/img/Mini/6cd5e730687a9b58a2fd41b9868c9b15/		Phishing	Phishing
http://www.w3schools.com/default.asp		Legitimate	Legitimate
http://www.godaddy.com/?ci=72737		Phishing	Legitimate
http://simplycool.fr/images/Nets/Sikker%20nettbetaling.htm		Phishing	Phishing
http://www.smvworld.com/del/		Legitimate	Phishing
http://www.baymakservisi-tr.com/wp-content/index.html		Phishing	Phishing

The accuracy of risk rating functionality is computed using True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), as shown in equation 5.1 below.

$$Accuracy = \frac{TP + TN}{P + N}$$ **(Equation 5.1: Accuracy Calculation of Risk Rating functionality)**

Where:

TP = when a web page is a phishing page and PhishProof classifies it as a phishing page

TN = when a web page is not a phishing page (legitimate) and PhishProof classifies it as not a phishing page

FP = when a web page is not a phishing page (legitimate) and PhishProof classifies it as a phishing page

FN = when a web page is a phishing page and PhishProof classifies it as not a phishing page

$$P = TP + FP$$

$$N = TN + FN$$

PhishProof correctly classified 8 web pages and failed to classify 2 web pages, 1 phishing and 1 legitimate, as shown in figure 5.6. The values of TP, TN, FP and FN for risk rating evaluation are shown in table 5.2. These values compute accuracy of risk rating functionality to be 80%.

Figure 5.6: PhishProof risk rating results.

Table 5.2: Values for calculating accuracy of risk rating functionality.

Label	Value
TP	6
TN	2
FP	1
FN	1

5.2.2 User Acceptance Testing (UAT)

This is the second test performed to evaluate PhishProof. UAT is carried out to check if requirements of a system are met. Following are the specifications of UAT for PhishProof.

Users

Participants involved in testing of PhishProof are all students of University of Manchester. All participants are in the age group 20 to 32 years old and regularly use Firefox and other browsers to perform operations that involve providing confidential information on a web page. These participants are divided into two groups: expert users and naïve users, based on their knowledge of phishing. 3 students from School of Computer Science make up for expert users and 3 students from Manchester Business School (MBS) make up for naïve users.

Testing Environment & System Requirements

UAT was conducted on Wednesday 29th August 2012 in John Rylands Library. All participants used laptops with windows 7 and Firefox version 12 or above. They were connected to internet via Wi-Fi.

Task

Each user in both the groups was given the following set of tasks to perform:

1. Open www.phishproof.com and download PhishProof installation package.
2. Follow the toolbar tutorial on website and install PhishProof toolbar on Firefox.
3. Visit each of the following URLs one by one and check the appropriate checkbox.

Table 5.3: Task sheet for naive and expert users.

Web page URL	Phishing	Legitimate
http://sugarcrm101.com/wp-includes/js/tinymce/plugins/inlinepopups/skins/clearlooks2/img/Mini/6cd5e730687a9b58a2fd41b9868c9b15/	☐	☐
http://gmailhelp.uphero.com/gmail/index.html	☐	☐
http://www.godaddy.com/?ci=72737	☐	☐

4. Report (if any) URLs identified as phishing to PhishProof using "Report a Phishing Site" option.
5. Report (if any) URLs you think are added to blacklist as an error, to PhishProof using "Report Incorrectly blocked URL" option.
6. Report (if any) bugs or issues faced while performing task, to PhishProof using "User Feedback/Report a Bug" option.

All participants were requested to fill a questionnaire after performing the tasks:

Table 5.4: UAT questionnaire.

PhishProof Toolbar Questionnaire						
	Strongly Agree	**Agree**	**Some What Agree**	**Some What Disagree**	**Disagree**	**Strongly Disagree**
It is easy to report a suspicious URL using PhishProof toolbar	1	2	3	4	5	6
It is easy to report an incorrectly blocked URL using PhishProof toolbar	1	2	3	4	5	6
It is easy to report any bug in PhishProof using PhishProof toolbar	1	2	3	4	5	6
It is easy to differentiate between a phishing and legitimate website using PhishProof toolbar	1	2	3	4	5	6
warning alert messages are attention grabbing	1	2	3	4	5	6
audio alert is more effective as compared to alert message	1	2	3	4	5	6
PhishProof toolbar is user friendly and easy to use	1	2	3	4	5	6

Results

The results obtained for both the groups are surprisingly similar. Both groups responded well to the alert messages. The alert messages and audio alerts used to notify users about phishing website were also a success as can be depicted from the results shown in figure 5.7. Participants were not able to ignore them and were able to recognise the phishing threat. All participants were able to download and install the toolbar easily. The reporting features also had 100% success rate and no help was required by any participant during the activity.

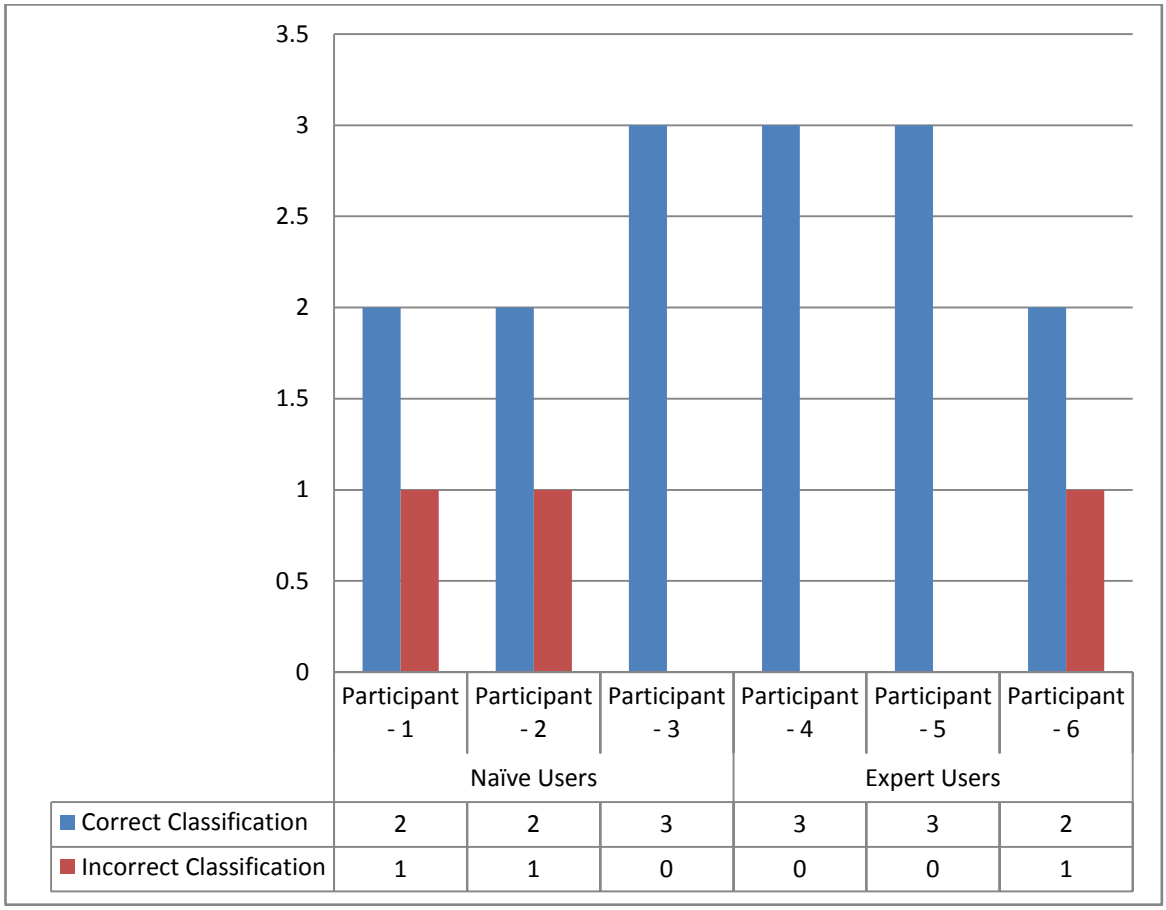


Figure 5.7: Test results.

5.2.3 PhishProof Performance Evaluation

Evaluation of PhishProof's performance is carried out to check how its operations affect Firefox. To test performance of the toolbar, two timers have been used. First timer is initiated when a page is requested by the user, whereas the second one is initiated when document is loaded completely. The difference between these two timers is the time required to load a page in Firefox. This experiment was carried out for 10 different websites with and without PhishProof toolbar using 2.10 GHz Core 2 Duo with 4 GB of

RAM. The average time for loading a page in Firefox without PhishProof is 2.5 seconds. Whereas average time to load a page with PhishProof is 3.7 seconds. Although PhishProof takes some time to calculate risk rating of a page, it does not affect the time required to load a web page because its operations run in the background. Hence, it can be concluded from this evaluation that the performance of Firefox is not affected by PhishProof toolbar and its phishing checks.

5.3 PhishProof Limitations

A lot of anti-phishing tools already exist and many anti-phishing tools will be developed in future as well but no phishing tool can protect users against all kinds of attacks. All phishing tools available have some limitations and aim at protecting users against some of the available and recognised phishing threats. New techniques to get users' confidential data are being developed everyday by attackers. Like all existing solutions, PhishProof also has some limitations and cannot protect users against all known phishing attacks. Following are some limitations of proposed PhishProof system:

1. It is only developed for Firefox browser and will not work for other browsers
2. It cannot protect users against JavaScript based attacks
3. It cannot protect users against malware attacks and there is no way to detect whether an attacker has installed anything on user's system or not
4. It has problems displaying toolbar in tabbed browsing
5. It only analyses 6 features of a webpage, where attackers can bypass these features if they know how these toolbar features work. Therefore, more features are required for the risk rating functionality

5.4 Chapter Summary

This chapter described how PhishProof system can be tested and evaluated. All phishing checks are tested individually and then the system is tested as a whole using two groups of users. Results of each test show that system checks cannot be bypassed easily and warning messages are hard to ignore. PhishProof risk rating functionality is also tested and evaluated. Results show that it has an accuracy of 80%. Furthermore UAT conducted shows that the system developed is user friendly and can be used effectively by both expert and naïve users. It is also shown in this chapter that the Performance of Firefox is not affected by PhishProof and its phishing checks. They system has some limitations which include malware and JavaScript based attacks.

Chapter 6 Conclusion and Future Work

This chapter summarises the proposed PhishProof system and also mentions some suggestions for future work that can be done to improve the system.

6.1 Conclusion

The main aim of helping users differentiate between legitimate and phishing web pages with minimal input has been achieved. This has been achieved by analysing existing anti-phishing solutions.

In this project, a combination of blacklist and web page content analysis method is used to design and develop a 3-level protection system against phishing. Level 1 uses blacklist method to protect users against phishing attacks. Level 2 uses a combination of blacklist and web page content analysis to implement referrer and links check. Level 3 uses web page content analysis to analyse 6 features of a web page in two rounds. In first round, all individual labels are assigned scores, whereas in second round all individual scores are combined with preset weights to compute risk rating. If risk rating percentage is above a threshold value, the webpage is classified as a phishing page. The 6 features analysed in first round of level 3 are: referrer, password, encryption, age of domain, country of server hosting the domain and company hosting the domain. PhishProof uses alert messages and audio alerts to notify users about potential phishing websites.

The proposed solution allows users to report web pages they consider as phishing through PhishProof website. They can also report any URL they think has been incorrectly blocked by PhishProof. All these reports can be seen using PhishProof backend admin panel which also allows system administrator to manage the blacklist and whitelist.

6.2 Future Work

Only 6 web features are analysed for risk rating because of time restrictions. Although these features are important in determining the nature of a page, still a lot more needs to be done in order to provide users with a phishing free environment. This can be done by adding more features like analysing number of JavaScript in a page, image checks, and keyword analysis to risk rating functionality. The toolbar should also be able to detect

malware attacks. In addition, it should work for tabbed browsing and should be made available for other popular browsers as well.

Adding more features will improve risk rating functionality of PhishProof. At present the risk rating functionality relies on 6 features only and if a particular feature is not present, a noticeable change in the risk rating percentage can be seen. This can allow attackers to bypass risk rating check. Adding more features will restrict attackers and lower the dependency of risk rating on each feature.

PhishProof toolbar cannot protect users against malware attacks. If a website tries to install some program on a user's system, PhishProof won't be able to stop this. Key-loggers are a very common way of getting user's personal details. Malware checks should be added to the proposed solution that will notify users when a program is being installed on their system.

PhishProof toolbar is not updated when a user moves from one browser tab to another within a Firefox browser window. The toolbar is only updated when a page is loaded. If a user navigates from one tab to another, the toolbar only displays result for the most recent page that is opened and results for other web pages accessed are not maintained.

A new version of PhishProof can be developed for other commonly used browsers like Internet Explorer, Google Chrome etc. This will increase the number of users protected against phishing attacks.

References

[1] Bergholz,A., Beer J. D., Glahn S., Moens M.-F., G. Paass, and S. Strobel (2009). New filtering approaches for phishing email. *Journal of Computer Security*, pp. 1- 31.

[2] Chandrasekaran, M., Narayanan, K. andUpadhyaya, S. (2006) Phishing email detection based onstructural properties. *In NYS Cyber Security Conference*.

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

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

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

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

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

[8] NETCRAFT INC, (2012) .Netcraft anti-phishing toolbar. [Online] Available at:http://toolbar.netcraft.com/ [Accessed 1 April, 2012]

[9] CLOUDMARK INC, (2012). Cloud mark desktop one. [Online] Available at: http://www.cloudmark.com/desktop/download/ [Accessed 22 March, 2012]

[10] Microsoft (2011), SmartScreen Filter. [Online] Available at:http://windows.microsoft.com/en-US/internet-explorer/products/ie-9/features/smartscreen-filter [Accessed 24 March, 2012]

[11] Microsoft (2009), SmartScreen Filter and Resulting Internet Communication in Windows 7 and Windows Server 2008 R2. [Online] Available at: http://msdn.microsoft.com/en us/library/ee126149(v=ws.10).aspx [Accessed 24 March, 2012]

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

[13] Sheng, S. et al., 2009. *An empirical analysis of phishing blacklists.* Mountain View, CA, s.n.

[14] APPLE INC. (2012). New features in safari. [Online] Available at: http://www.apple.com/safari/features.html#security. [Accessed 8 April, 2012]

[15] Schneider, F., Provos, N., Moll, R., Chew, M., And Rakowski,B. Phishing protection: Design documentation. [Online] Available at: https://wiki.mozilla.org/Phishing_Protection:_Design_Documentation. [Accessed on 10 April, 2012]

[16] Net Applications. Inc. (2008). Browser market share q4. [Online] Available at:http://marketsharehitslink.com/report.aspx?qprid=0&qpmr=15&qpdt=1&qpct=3&qpcal=1&qptimeframe=Q&qpsp=39 [Accessed 10 April, 2012]

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

[18] Anti-Phishing Working Group (2006). Phishing Activity Trends Report.[Online] Available at: http://www.antiphishing.org/reports/apwg_report_june_06.pdf [Accessed 11 April, 2012]

[19] Jendricke, U, D. Gerd tom Markotten, "Usability Meets Security - The Identity-Manager As Your Personal Security Assistant for The Internet," in *Proceedings of The 16th Annual Computer Security Applications Conference (ACSAC'00)*, 2000.

[20] LUDL, C., MCALLISTER, S., KIRDA, E., AND KRUEGEL, C. On the effectiveness of techniques to detect phishing sites. In *DIMVA '07: Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (Berlin,Heidelberg, 2007), Springer-Verlag, pp. 20–39.

[21] Garera, S., Provos, N., Chew, M. and Rubin, A. D. A framework for detection and measurement of phishing attacks. In *WORM '07: Proceedings of the 2007 ACM*

workshop on Recurring malcode, 9New York, NY, USA, 2007). ACM, pp. 1-8.

[22] Chou, N.,Ledesma, R.,Teraguchi, Y., Boneh, D. and Mitchell, J. Client-side defense against web-based identity theft. In 11th *Annual Network and Distributed System Security Symposium (NDSS '04),* San Diego, 2005.

[23] SpoofGuard (2005). Client-side defense against web-based identity theft [Online]Available at: http://crypto.stanford.edu/SpoofGuard/ [Accessed 22 April, 2012]

[24] Baihan, M. (2011) *Anti Spoofing tool – Project background report*, Unpublished Thesis (Msc.), University of Manchester.

[25] Tout, H. andHafner, W. (2009*), Phishpin: An identity-based anti-phishing approach.* In International Conference on Computational Science and Engineering.

[26] Gartner Research. Number of Phishing E-Mails Sent to U.S. Adults Nearly Doubles in Just Two Years. Press Release, 2006.

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

[28] Anti-Phishing Working Group.(2012) Phishing Activity Trends Report. Pp. 1-11

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

[30]Jagatic, T., Johnson, N., Jakobsson, M. and Menczer, F. (2005) "Social Phishing" [PDF]. *To appear in the CACM (October 2007)* Available at: http://www.indiana.edu/~phishing/social-network-experiment/phishing-preprint.pdf. [Accessed 28April, 2012]

[31] UPI (2010), Google, Yahoo! join security firm in fight against phishing. [Online] Available at:http://www.upi.com/Business_News/Security-Industry/2010/04/14/Google-Yahoo-join-security-firm-in-fight-against-phishing/UPI-67691271272715/#ixzz25l6BAYfyhttp://www.upi.com/Business_News/Security-Industry/2010/04/14/Google-Yahoo-join-security-firm-in-fight-against-phishing/UPI-67691271272715/[Accessed 6 June, 2012]

[32] Casto, G., Fisher, O., Moll, R., Nazif, M. and Born, D. (2009) Google safe Browsing. Weblog [Online] 24thNovenmber.Availabe at:http://code.google.com/p/google-safe-browsing/wiki/Protocolv2Spec[Accessed 23 June, 2012]

[33] Goldman, D. (2012) *Facebook turns its user into anti- phishing detectives*. [Online] Available at: http://money.cnn.com/2012/08/09/technology/facebook-phishing/ [Accessed 3 July, 2012]

[34]Mcgeehan, R. (2009) Protect yourself against phishing. *The Facebook blog.*Weblog [Online] 2nd May. Available from: http://blog.facebook.com/blog.php?post=81474932130 [Accessed 1 July, 2012]

[35] Alexa (1996) Topsites. [Online] Avalable at: http://www.alexa.com/topsites[Accessed 22 June, 2012]

[36] Anti Phishing Scams (2012), Defending against Phishing Attacks – What Is Phishing. [Online] Available at: http://www.antiphishingscams.com/defending-against-phishing-attacks.html [Accessed 28 June, 2012]

[37] Microsoft patterns & practices (2005), Security Guidelines: ASP.NET 2.0. [Online] Available at: http://msdn.microsoft.com/en-us/library/ff649487.aspx [Accessed 4 July, 2012]

[38] Netcraft (2012), Phishiest Countries. [Online] Available at: http://toolbar.netcraft.com/stats/countries[Accessed 1 July, 2012]

[39]Netcraft (2012), PhishiestHosters. [Online] Available at:http://toolbar.netcraft.com/stats/hosters[Accessed 7 July, 2012]

[40] Hindle, A., Godfrey, W. M. and Holt, C. R. (2008), Reverse engineering CAPTCHAs, *15th working conference on reverse engineering*, pp. 59-68

[41] Troung, D. H., Turner, F.C. and Zou, C. C. (2011), iCAPTCHA: The Next Generation of CAPTCHA Designed to Defend Against 3rd Party Human Attacks. [Online] Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5963009[Accessed 7 July, 2012]

[42] Snijders, A. B. T. (2005), Power and sample size in multilevel modelling, *Encyclopedia of Statistics in Behavioral Science*, Vol. 3, pp. 1570-1573

[43] Wright, P., Wooley, M. D., and Wooley, B. Techniques & tools for using color in computer interface design. [Online] Available at: http://pirun.ku.ac.th/~btun/pdf/colour1.pdf[Accessed 12 July, 2012]

[44] Bishor, J. (2012), Chapter1: Getting started. [Online] Available at: http://www.borngeek.com/firefox/toolbar-tutorial/chapter-1/[Accessed 12 June, 2012]

[45] Chittora, A. (2009), Mozilla firefox extensions development tool. [Online] Available at: http://www.slideshare.net/bijolianabhi/firefox-extension-development[Accessed at: 10 July, 2012]

[46] Flanagan, D. (2002), Variables, Javascript: The definitive guide. Fourth edition.Sebastopol: O'Reilly & Associates

[47] Rakin, A. (2007), Never use a warning when you mean undo. [Online] Available at: http://www.alistapart.com/articles/neveruseawarning[Accessed 5 July, 2012]

[48] W3C (2012), XMLHttpRequest Level 2. [Online] Available at: http://www.w3.org/TR/XMLHttpRequest/[Accessed 12 July, 2012]

[49] W3school.com (2012), The XMLHttpRequest object. [Online] Available at: http://www.w3schools.com/dom/dom_httprequest.asp[Accessed 21 July, 2012]

[50] Badassery, F. (2007), ParseURI 1.2: split URLs in javascripts. [Online] Available at: http://blog.stevenlevithan.com/archives/parseuri [Accessed 21 June, 2012]

[51] Saez, D. (2005), Whois.PHP. [Online] Available at: http://www.phpkode.com/source/s/phpwhois/phpwhois-4.2.0/whois.main.php[Accessed 19 June, 2012]

[52] Rosiello, P. E. A.,Kirda, E.,Kruegel, C. and Ferrandi, F., A Layout-Similarity-Based Approach for Detecting Phishing Pages. [Online] Available at:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4550367&tag=1[Accessed 22 June, 2012]