# CONTENT SIMILARITY BASED ANTI-PHISHING

**A DISSERTATION**

*Submitted in partial fulfillment of the requirements for the award of the degree of*

**INTEGRATED DUAL DEGREE**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**(With Specialization in Information Technology)**

**By**

**KONCHADA SWAGAT**

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)
JUNE, 2011

## CANDIDATE'S DECLARATION

I hereby declare that the work being presented in the dissertation work entitled "**Content Similarity Based Anti-Phishing**" towards the partial fulfillment of the requirement for the award of the degree of **Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology)** submitted to the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India is an authentic record of my own work carried out during the period from May, 2010 to May, 2011 under the guidance and provision of **Dr. Manoj Misra,** Professor, Department of Electronics and Computer Engineering, IIT Roorkee.

I have not submitted the matter embodied in this dissertation work for the award of any other degree and diploma.

Date: May 2011
Place: Roorkee

K. Swagat.
**(Konchada Swagat)**

## CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: May,2011

Place: Roorkee

Manoj Misra

**Dr. Manoj Misra**
**Professor**
**E&CE Department**
**IIT Roorkee, India**

# ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. Manoj Misra**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his mentorship, kind guidance, regular source of encouragement and assistance throughout the work of this dissertation. I would state that this work would not have been in the present shape without his inspirational support and I consider myself fortunate to have done my dissertation under him.

I also extend my sincere thanks to **Dr. S.N. Sinha**, Professor and Head of the Department of Electronics and Computer Engineering for providing facilities for the work.

I would like to thank all my friends who supported and encouraged me to finish this work.

Finally, I would like to say that I am indebted to my parents for everything that they have given to me. I thank them for the sacrifices they made so that I could grow up in a learning environment. They have always stood by me in everything I have done, providing constant support, encouragement, and love.

**Konchada Swagat**

# LIST OF FIGURES

# ABSTRACT

Phishing is a critical problem traditionally involving a deceiving mail and a duplicate website. Phishers endeavor to lure users into submitting their crucial information and use the information for illegal purposes. The menace of phishing is rampant on the Internet. A recent report of Gartner puts the losses due to phishing at more than $3.2 billion annually [1], banking and other financial institutions and their users being the most affected. But in recent times phishing has spread into new frontiers like social networking, cyber warfare, etc., Innovative phishing strategies are being developed from time to time which go unnoticed. In the midst of all the chaos, it's the user who bears the brunt of phishing attacks.

Numerous anti-phishing tools have been developed with various strategies. Attempts have been done to stop phishing when the phish is in the form of a mail using mail filters[2]. Organizations which deal with highly financial transactions have chosen two-way authentication to avoid risks of phishing, but such methods could not revolutionize the web because of the additional costs and effort involved. Some anti-phishing browser tools use web related heuristics like URL usability, ip-address usage, etc., but lead to unimpressive performance.

We present here, a Content Similarity based Anti-Phishing system, a ubiquitous, robust and scalable method to detect phishing. Instead of depending on phishing related heuristics, we directly address the source of phishing, the content of pages and the similarity of pages involved. A sophisticated algorithm has been presented with proof of concepts wherever required, to detect similar pages and further establish the authority of those pages based on their URL, content and newly introduced factors like site-population. The performance achieved by our system has been impressive with a true-positive rate of more than 99% and better compared to previously developed systems which were discussed.

# CONTENTS

# 1. Introduction

## 1.1 Background

The phrase “information security” has had multiple definitions with subtle variations from sources ranging from research papers, white papers, Government laws, etc., in all; the one given by U.S. Code stands an authoritative one:

Information Security is defined as the means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide—

A. Integrity, which means guarding against improper information modification or destruction, and includes ensuring information nonrepudiation and authenticity;
B. Confidentiality, which means preserving authorized restrictions on access and disclosure, including means for protecting personal privacy and proprietary information;
C. Availability, which means ensuring timely and reliable access to and use of information.” [3]

The above definition purports to all information systems before and after the Information Technology revolution, which shifted the perception towards computing as a rare and luxury entity to a ubiquitous, necessary entity which is a part of almost every important transaction of common man.

Information security gathered enhanced attention and importance after the foray of information technology into banking and money related services. In order to understand the possible reach of an information security breach, it definitely helps to have the knowledge that the reach and inevitability of Information Technology in sectors like banking, etc., is pivotal.

The sectors using Information Technology form the domain for hackers to attack and exploit and extract as much false advantage as they can. The strategy for a hacker to attack a financial organization would be as follows.

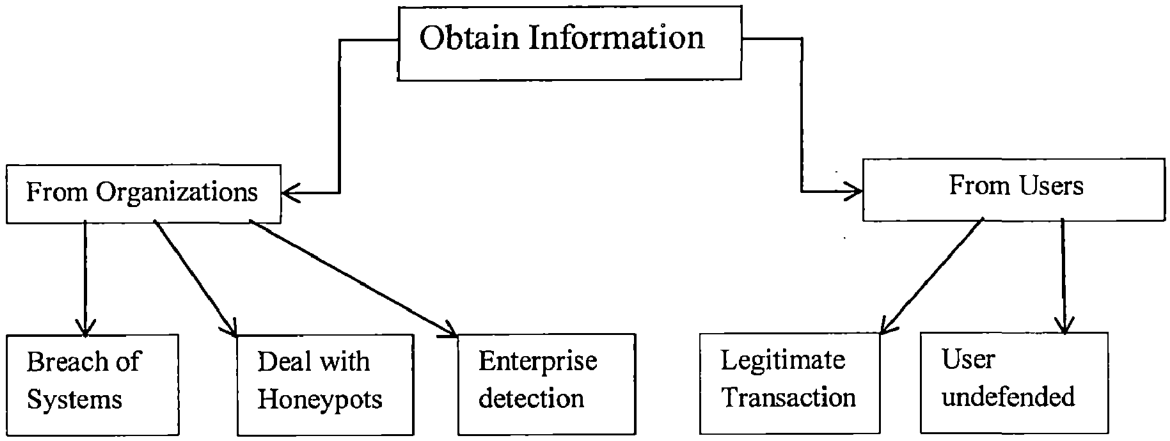


FIGURE 1.1 : HACKER STRATEGY

It is comparatively easy for a hacker to extract information from a user than from an organization, thus violating the key provision mentioned in the definition of information security, "Confidentiality". Confidentiality is defined not just as the prevention of disclosure of information, but as the prevention of disclosure to unauthorized individuals. Let us look into the following statistics for the year 2010, to have a better understanding of the impact of phishing.

| | April | May | June |
|---|---|---|---|
| Unique Phishing Sites | 33,253 | 31,856 | 32,279 |
| Unique Domains Pairs | 12,268 | 12,162 | 9,199 |
| Unique BrandDomain | 14,945 | 14,848 | 11,470 |
| Unique Brands | 270 | 276 | 258 |
| Avg. URLs per Brand | 123.16 | 115.40 | 125.11 |

FIGURE 1.2: EXTENT OF PHISHING [4]

## 1.2 Phishing

An authentic and modern definition of phishing is given by the Anti-Phishing Working Group (APWG), as follows:

- Phishing is a criminal mechanism employing both social engineering and technical subterfuge to steal consumers' personal identity data and financial information [4].

The word "phishing" had its origin from the metaphorical sense of *fishing* i.e., trapping users into submitting their critical information (unlike their own life as in fishing). As with most security attacks, the word "phishing" was coined not by the research community but by the hackers who first deployed an information stealing attack against users of America-Online by scamming them into unsuspectingly submit their account passwords. In fact "ph" is a common replacement for "f' in hacker circles, starting with coining of the word "phreaking", by John Draper, considered to be the first hacker who created the Blue Box, a device which could be used in hacking telephone systems in early 1970 [4].

## 1.3 Broad Classification of Attacks

In effect, any information security attacks can be broadly classified into two types.

**Systemic attacks**

Systemic attacks are those which exploit the deficiencies in systems which compose the information system, to gain access to, or disrupt confidential information. Some attacks which fall into the category are the following:

- Launching a DDoS against a website to breach it's security infrastructure
- Hacking into someone's account using information gathered by packet filters
- Cracking an encryption by guessing or brute-forcing for finding a private-key to read encrypted data

**Semantic attacks**

Semantic attacks are those which exploit the deficiencies in humans who complement the information system, to gain access to critical and confidential information.

Phishing is a form of semantic attack specifically. Phishing involves a communication with a user using false authentication and luring him/her into submitting the relevant information like passwords for their social networking accounts, Personal Identification Numbers (PINs) of their respective bank accounts, etc.,

## 1.4 How Phishing Works

Phishing started as a simple method of making a user believe in the authenticity of the fake website with images and text similar/identical to that being used in the original website. Even in the present scenario, the crux of phishing remains the same. But the part of phishing which evolved through the times is the medium through which ignorant users are made to come across the links, click on them and continue with compromising important information only to find out later that they have been conned.

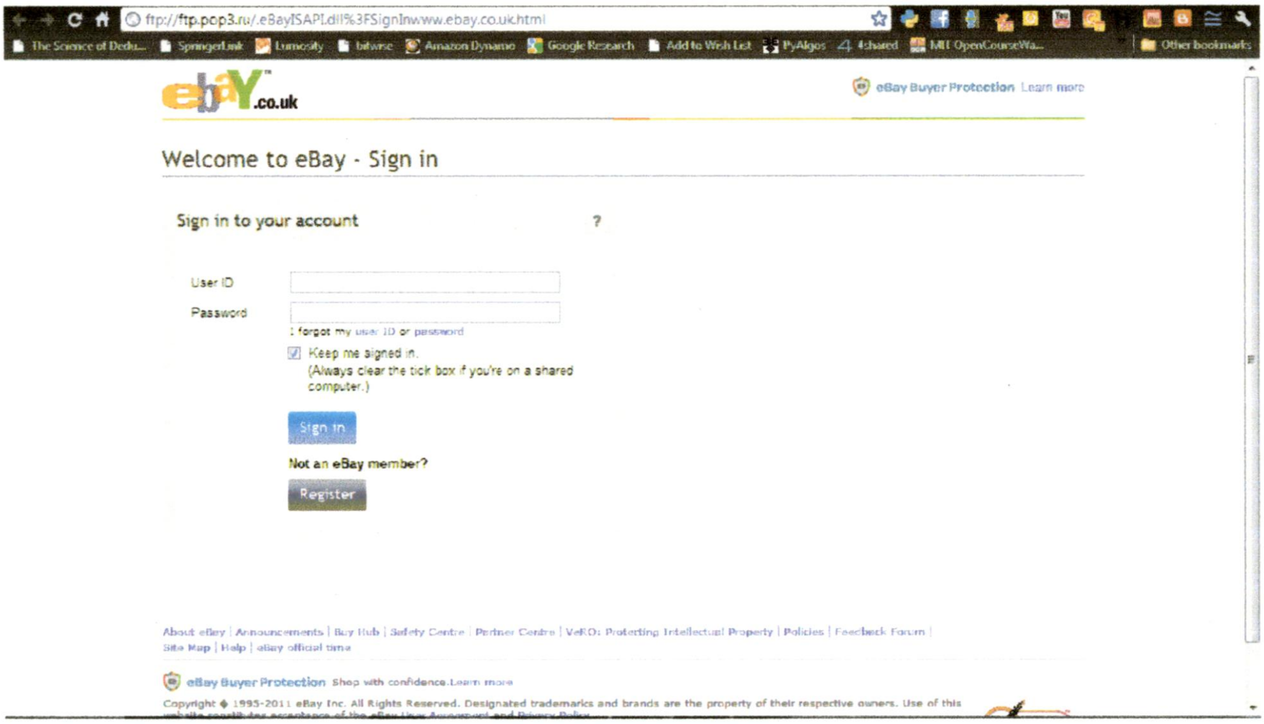
The "Phishing Archive" maintained by the Anti-Phishing Working Group describes phishing attacks dating back to September 2003. A cognitive walkthrough on sample attacks within this archive was performed by [5] with a goal to gather information about which strategies, attackers exploit or use, which could be listed as follows:

- Lack of Knowledge of end user.
  - Lack of computer system knowledge
  - Lack of knowledge of security and security indicators
- Visual Deception
  - Visually deceptive text
  - Images masking underlying text
  - Images mimicking windows
  - Windows masking underlying windows
  - Deceptive look and feel
- Bounded Attention
  - Lack of attention to security indicators
  - Lack of attention to the absence of security indicators

In Figure 1.3, you can find a screenshot of a phishing site with URL ftp://jacvk11:nevada88@ftp.pop3.ru/.eBayISAPI.dll%3FSignInwww.ebay.co.uk.html [6]

Followed by a screenshot of original eBay site,

https://signin.ebay.com/


ftp://ftp.pop3.ru/.eBayISAPI.dll%3FSignInwww.ebay.co.uk.html
The Science of Dedu...
SpringerLink
Lumosity
bitwise
Amazon Dynamo
Google Research
Add to Wish List
PyAlgos
4shared
MIT OpenCourseWa...
Other bookmarks
ebY.co.uk
eBay Buyer Protection Learn more

Welcome to eBay - Sign in

Sign in to your account ?

User ID

Password

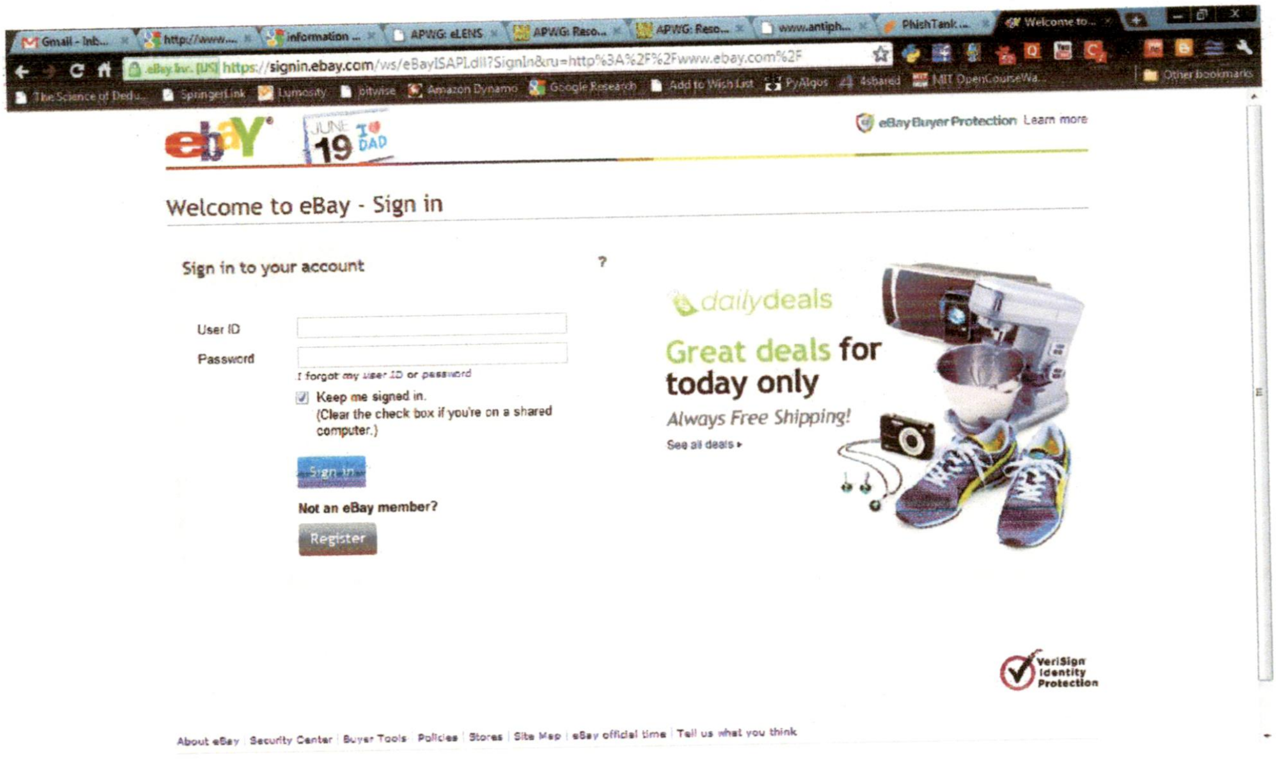
I forgot my user ID or password

☑ Keep me signed in.
(Always clear the tick box if you're on a shared computer.)

Sign in
Not an eBay member?
Register
About eBay | Announcements | Buy Hub | Safety Centre | Partner Centre | VeRO: Protecting Intellectual Property | Policies | Feedback Forum |
Site Map | Help | eBay official time
eBay Buyer Protection Shop with confidence.Learn more

Copyright © 1995-2011 eBay Inc. All Rights Reserved. Designated trademarks and brands are the property of their respective owners. Use of this website constitutes acceptance of the eBay User Agreement and Privacy Policy.

FIGURE 1.3: SCREEN SHOT OF PHISHING SITE



https://signin.ebay.com/ws/eBayISAPI.dll?SignIn&ru=http%3A%2F%2Fwww.ebay.com%2F

Welcome to eBay - Sign in

Sign in to your account

User ID

Password

I forgot my user ID or password

☑ Keep me signed in.
(Clear the check box if you're on a shared computer.)

Not an eBay member?

About eBay | Security Center | Buyer Tools | Policies | Stores | Site Map | eBay official time | Tell us what you think

FIGURE 1.4: SCREENSHOT OF ORIGINAL EBAY LOGIN SITE

In the figures shown, one can observe the phisher using every trick at his disposal to give an authentic impression to the phishing site. Visually Deceptive Text has been used by having imitative phrases like

"Welcome To eBay – Sign In", "Sign in to your account"

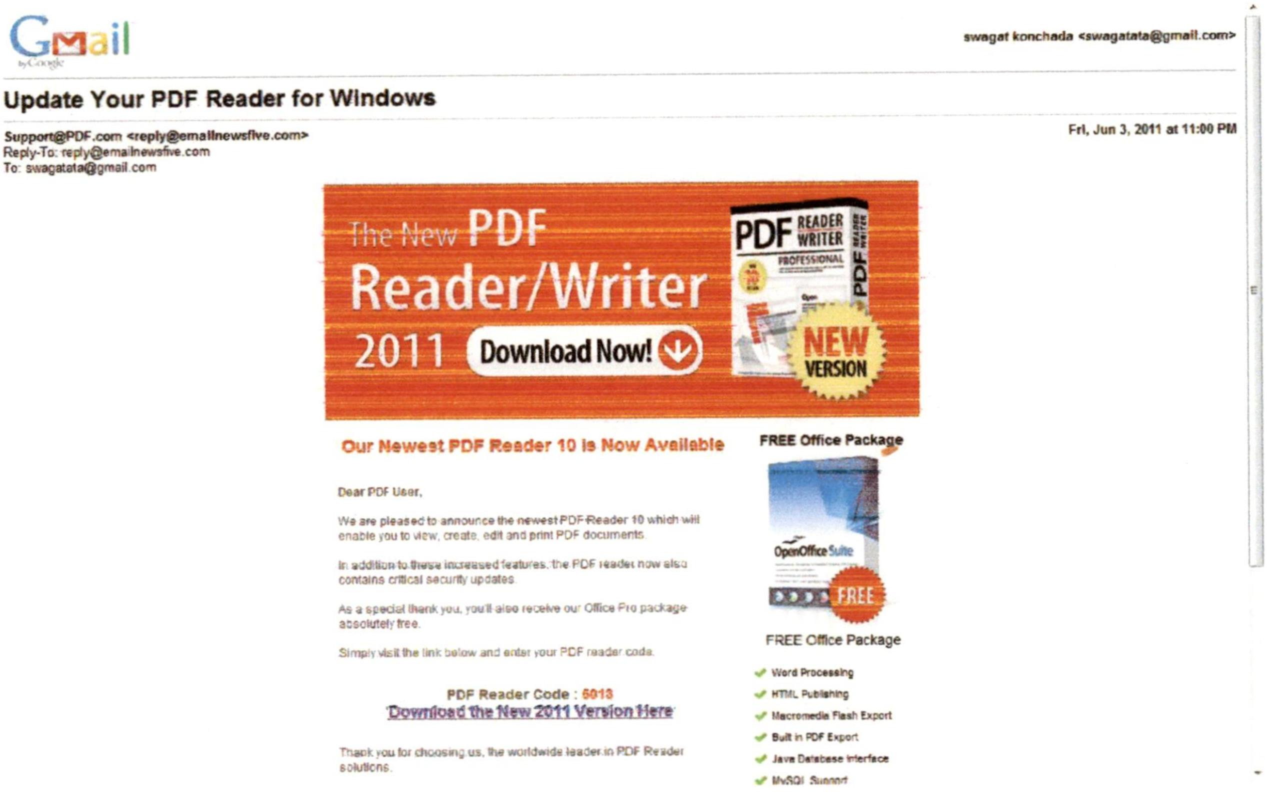
Also, images mimicking the original site have been used with exact positioning. The identical eBay logo used in the phishing site could be observed. In addition, images which are used for form submission, i.e., the "sign in" and "cancel" buttons are also imitated along with the dimensions of all elements used in the form.

Complementing the merits of the phisher are the deficiencies of the user. Lack of knowledge of computer systems or the mechanism of URLs or lack of attention to security indicators if any installed could be, in an unpresumptuous manner, attributed to a user who is deceived by the aforementioned bait.

Here, the URLs of both the sites are visibly nonmatching. Users with basic knowledge of URLs and domain names could avoid such attacks with less effort. But hackers have a workaround for this defense too. Phishers generally register domain names which have an extent of visible resemblance to the original site and host their phishing sites on the respective domain names. For example, a phishing site of 'www.phishtank.com' can have any of the following domain-names:

- www.phshingtank.com
- www.phishtank1.com
- www.phishtank.net
- www.phish.tank.com

The challenge left for the phisher is to take care that a user clicks on the link of the phishing site. The advent of web based mail clients came as a boon to such phishers. A recent finding has proved that 1 in every 242 emails in the Internet are phishing emails [7]. Mails are sent to innocent users from entities pretending to be institutions like banks, Government Finance regulators, etc., and mostly comprise a demand of key information related to identification or security corresponding to the institution, from the user. Even in the present day mail users, mails like the following are a daily occurrence.



swagat konchada <swagatata@gmail.com>

**Update Your PDF Reader for Windows**

Support@PDF.com <reply@emailnewsfive.com>
Reply-To: reply@emailnewsfive.com
To: swagatata@gmail.com

Fri, Jun 3, 2011 at 11:00 PM

Our Newest PDF Reader 10 is Now Available

Dear PDF User,

We are pleased to announce the newest PDF Reader 10 which will enable you to view, create, edit and print PDF documents.

In addition to these increased features, the PDF reader now also contains critical security updates.

As a special thank you, you'll also receive our Office Pro package absolutely free.

Simply visit the link below and enter your PDF reader code.

PDF Reader Code : 6013
Download the New 2011 Version Here

Thank you for choosing us, the worldwide leader in PDF Reader solutions.

FREE Office Package

FREE Office Package

- ✓ Word Processing
- ✓ HTML Publishing
- ✓ Macromedia Flash Export
- ✓ Built in PDF Export
- ✓ Java Database Interface
- ✓ MySQL Support

FIGURE 1.5: PHISHING MAIL WITH VISUAL DECEPTION

The mail shown in figure 1.5 claims to users that their installed software (in this case, PDF Reader), is outdated and asks them to download their new released version by clicking on the provided link, which only leads them to a phishing site to gather more personal information from users. Every phishing email is a spam. But not every spam could be considered phish. Spam filters which generally use the text that of the mail to classify the mail as spam or non-spam fail when the mail contains images like above.

The recent revolution of social networking is another medium through which phishing URLs could be propagated [8]. Bots are being created which automatically share apps with links to phishing sites. Since a social-networking site like facebook contains almost the entire "friends" of a user, it becomes a faster medium to propagate unwanted links at a faster pace. In addition to these micro blogging sites like twitter advocate the use of short-URLs which hide the original URL under the face of a smaller and more concise one, to enable reduced memory usage by them.

## 1.5 Motivation

Phishing has caused huge losses to the financial industry not just with denial of access to e-mail but substantial financial loss due to theft of identity, key information and illegitimate transaction. Around US $2bn of losses has been reported to be suffered by businesses whose clients are victimized by phishing attacks [9]. An escalation of phishing attacks has been reported in 2007. 3.6 million Adults lost US$3.2 billion in the 12 months ending in August 2007 [1]. An April 2004 survey of 650 U.S. banking customers by software vendor Cyota shows that phishing is diminishing customer's trust in online interactions with their banks. In the study, 65 per-cents of account holders were less likely to use their bank's online services due to phishing, and 75 percent were less likely to respond to email from their bank because of phishing [9].

The damage caused by phishing highlights not only the importance of an effective and most efficient anti-phishing system with minimum costs, but also the ineffectiveness of already existing techniques. Many techniques are introduced and discussed in Chapter 2, along with their deficiencies.

## 1.6 Problem Statement

The goal of this dissertation was

"To design, implement and optimize an accurate, responsive, scalable and deployable anti-phishing system."

- Accuracy: The most important goal of this dissertation is to achieve high accuracy levels in detecting phishing sites in comparision with tools previously developed.
- Responsiveness: The second most important feature which was tried to achieve is usability. Since we aim to provide a great anti-phishing tool to the average user, responsiveness and in turn efficiency is of highest significance.
- Scalability: The World Wide Web is huge. So should any system which attempts to work with pages on WWW.
- Deployability: At the end, the system which is developed should be able to easily deployable with minimum additional costs.

## 1.7 Organisation of Dissertation

The remaining of this report is organized as follows

*Chapter 2*: In this chapter, we present a thorough review of pre-existing literature in the field of anti-phishing in a clear and classified manner. The classification is done on the basis of anti-phishing techniques' point of usability, i.e., whether a technique is designed to be used at the user-side of to be administered by an Organisation.

*Chapter 3*: In this chapter, we propose a new and innovative anti-phishing method based on content similarity of web pages. We emphasize upon why this particular approach is important and gets expected results. In addition the key concepts behind the method like similarity and site population defined and explained with the help of apt examples and demonstrations.

*Chapter 4*: In this chapter, we explain to the reader the intricacies of input data and extensively enumerate the steps involved in implementing the proposed solution. We also attempt to warn the interested reader who may attempt to build a similar system of the challenges that could be faced during the implementation.

*Chapter 5*: In this chapter, we specify the environment of the system. The nature of the test database, the extra factors which influence the results are also demonstrated. Finally, a comparision of various dimensions of accuracy is done with respect to previously established anti-phishing techniques.

# 2. Literature Review

## 2.1 Anti-Phishing Techniques

A solution to phishing cannot simply rely on millions of users being trained to check the details of email routing headers and to scrutinize the minutia of Internet URL web links to ensure that email communications are genuine, and not from a phisher. The basic building blocks of an effective anti-phishing effort include detection, prevention and education. Both organizations and individuals experience a loss due to phishing. Thus also the ways of tackling phishing must be a two-thronged approach where both the organizations and users are stake holders.

### 2.1.1 Anti-Phishing at User Side

A broad classification of Anti-Phishing strategies based on the philosophy behind the strategies could be described as shown [5].

*Educating People about phishing attacks*: Educating users of Anti-phishing has focused on online tutoring materials, testing, and situated learning. The tutoring materials explain what phishing is and provide tips to prevent users from falling for phishing attacks.

*Testing:* is used to account and demonstrate how susceptible people are to phishing attacks. For example, SonicWall has a web site containing screenshots of potential phishing emails [10]. Users are scored based on how well they can identify which emails are legitimate and which are not.

*Situated Learning*: This approach uses situated learning, where users are sent phishing emails to test their vulnerability of falling for attacks. At the end of this, users are given materials that inform them about phishing attacks.

*Anti-Phishing user interfaces*: Several software have been developed to alert users of phishing and to avoid phishing in a preventive manner using unique strategies. Some are quoted below.

- PwdHash transparently converts a user's password into a domain-specific password by sending only a one-way hash of the password and domain-name. Thus, even if a user falls for a phishing site, the phishers would not see the correct password[5].
- PassPet and Norton Safe Web are browser extensions that make it easier to login to known web sites, imply by pressing a single button. PassPet requires people to memorize only one password, and like PwdHash, generates a unique password for each site. They not only help in memorizing the passwords but identify legitimate websites by themselves, enormously reducing the expected background knowledge of the user [11].

*Automated detection of phishing*: Anti-phishing services are now provided by Internet service providers built into mail servers and clients, built into web browsers, and available as web browser toolbars.

Anti-Phishing software developed not only uses these strategies independently but they are at their zenith of efficiency and effectiveness when implementing a combination of above strategies.

## 2.2 Preventive Methods - Organisation's Perspective

### 2.2.1 Strong Website Authentication

This approach would require all users of legitimate e-commerce and e-banking sites to strongly authenticate themselves to the site using a physical token such as a smart card.

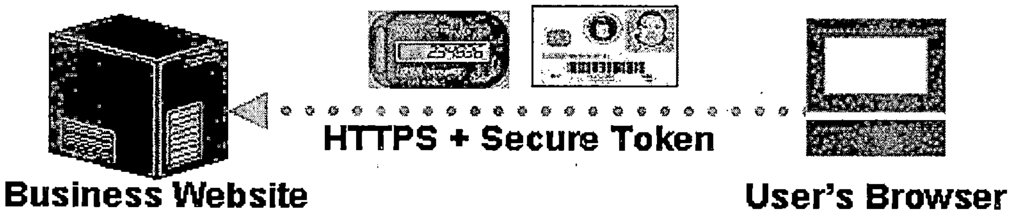


FIGURE2.1: TWO FACTOR AUTHENTICATION[12]

The positive aspects of this approach are

- Even if a user falls for a phishing attack, a phisher can't log into real site without the right physical token.
- Users have higher trust levels compared to systems of one-factor authentication.

The downsides of this approach are

- Establishment of infrastructure required for the system
- Extra costs involved and their burden on the users.
- Ineffective when the number of users involved is enormous(like that in a social networking site).
- Extra layer of authentication required in the case of loss or damage of software/hardware involved.

A prime example of two-factor authentication is RSA's SecurID. The SecurID mechanism works as a normal login mechanism with a user-confidential password, but in addition, consists of a token, either hardware or software (for devices like smartphones, PDAs, etc.,) which generates an authentication code at regular time intervals, generally 30 to 60 seconds. In order for a user to securely login to a website, he needs to provide the username, password along with the authentication code. Thus the phisher even if gets hold of a username, password and an authentication code, the lifetime of the information is no more than the token's time period, which is too short to launch an attack.

This approach is feasible for e-commerce and e-banking applications that do not have a large number of users, and where the risk of a phisher gaining access to a user's account are high. Examples would include corporate banking websites and high-value brokerage trading websites.

### 2.2.2 Mail Authentication

This method aims to stop phishing even before the phishing URL reaches the user through a phishing email, by authenticating either the server which sends the mail or the mail itself.

A straightforward method for phishers to broadcast phishing links is to flood mails containing those links to random users. The phisher only needs to have a list of legitimate email-addresses as targets for phishing. Delivery of mails is done through SMTP or Simple Mail Transfer Protocol. Here is a simple demonstration of how SMTP works.

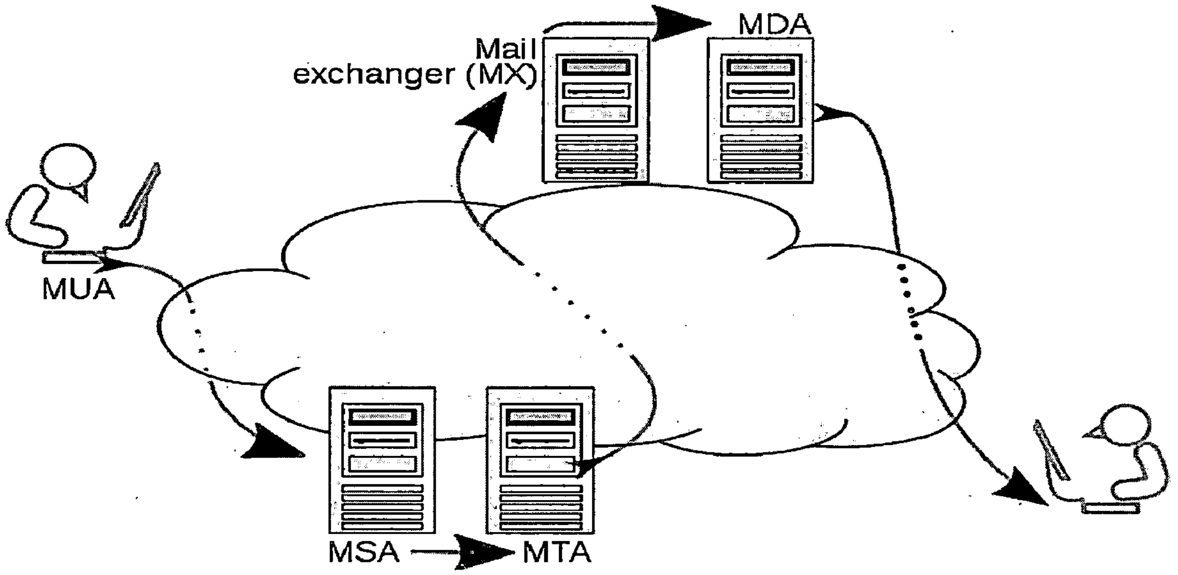


**Figure 2.2: SMTP working[13]**

| | |
|---|---|
| MUA | Mail User Agent |
| MSA | Mail Submission |
| MTA | Mail Transfer Agent |

SMTP in itself doesn't comprise of a security mechanism to maintain authentication. A phisher who owns an SMTP MTA or MSA can send mails claiming to be from any email address he wishes. The next MTA which receives the mails has no way of authenticating them. i.e., anyone who owns an MTA can send a mail to any email address in the world with the from-address corresponding to say 'developer@google.com'. In order to curb such violations, an extra layer of authentication against mail-servers could be used. One way of achieving it would be to match the IP address of the server sending the mail to the local DNS entries of the domain name corresponding to the from-address of the received mails.

A specific example of using DNS checking involves the use of DKIM, or Domain Key Identified Mail. It's an application layer analogy to transport layer's SSL. It tries to associate a mail message to a domain name using public-key cryptography. DKIM allows an organization to claim responsibility of the message while in transit [13]. The technique involves maintaining of public keys of organizations by DNS servers as a TXT resource. The DKIM header consists of the domain name of the acclaimed organization. Any MTA which receives the message authenticates this mail by decoding it by the public key which could be attained by doing a simple DNS query for TXT resource of the corresponding domain name.

The positives of this approach are

- Easy to configure at senders mail servers
- Makes it harder for phishers to be anonymous
- Legitimate business email can be better identified – lower spam false positives

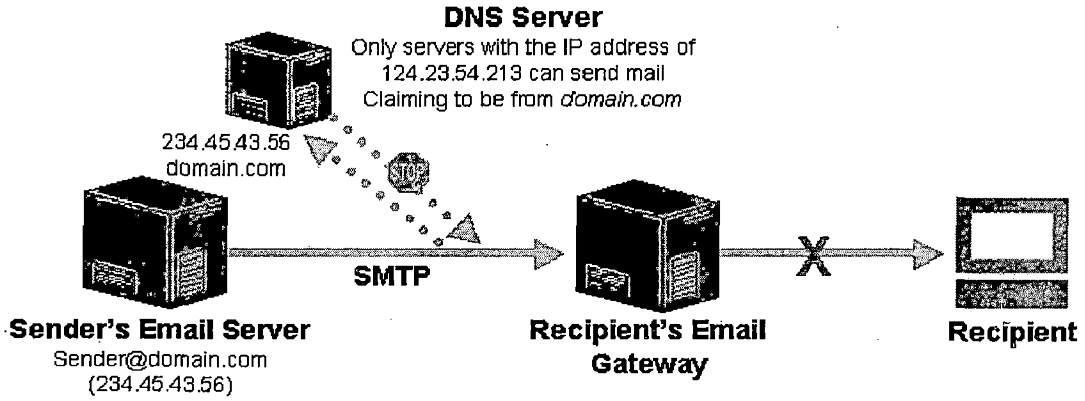


FIGURE 2.3: MAIL SERVER AUTHENTICATION[12]

The downsides of this approach are

- Requires sender and recipient gateways to both use these methods
- SMTP sender is not visible to recipient
- From: address still can be spoofed and users can be fooled
- Will be a problem for anyone using a 3rd party emailing service
- Doesn't accommodate email forwarding

Mail server authentication is a necessary but not sufficient approach in the battle against spam and email scams. This approach is appealing to the large web email providers and ISPs, as it can allow them to cut down on a great volume of spam.

## 2.3. Novel Anti-Phishing Techniques

### 2.3.1 Anti-Phishing Using Honeypots

A honeypot is a security resource whose value lies in being probed, attacked, or compromised. All honeypots work on the same concept: no legitimate user should be using or interacting with them, therefore any transactions or interactions with a honeypot are, by definition, unauthorized.

According to the definition given in [14], "a honeypot is a closely monitored computing resource that we want to be probed, attacked, or compromised". A honeypot is not necessarily a physical machine, but can be any computing resource which is open for phishers or other attackers to attack. Examples may consist of a Local Area Network (LAN), a computer or a web service.

In the case of phishing, the honeypot used is information, which only seems vital to phishers. Such a honeypot is termed honeytoken [15]. The fact remains that honeytokens which are themselves honeypots exist to lure and possibly track attackers. In fact, one major way for early detection of phishing sites is to use spam traps (i.e., honeypots against spams), which are usually email-addresses maintained especially to keep track of spam and its origin, to collect phishing emails. Another of implementing honeytokens is "phoneytokens"(or phishing honeytokens), which are usually provided to phishing sites as information of a legitimate user and which might later help in detecting phishers in case of the use of phoneytokens.

As a powerful anti-phishing tool, honeypots have been widely used by security service providers and financial institutes to collect phishing mails, so that new phishing sites can be earlier detected and quickly shut down. Another popular use of honeypots is to collect useful information about phishers' activities, which is used to make various kinds of statistics for the purposes of research and forensics [15].

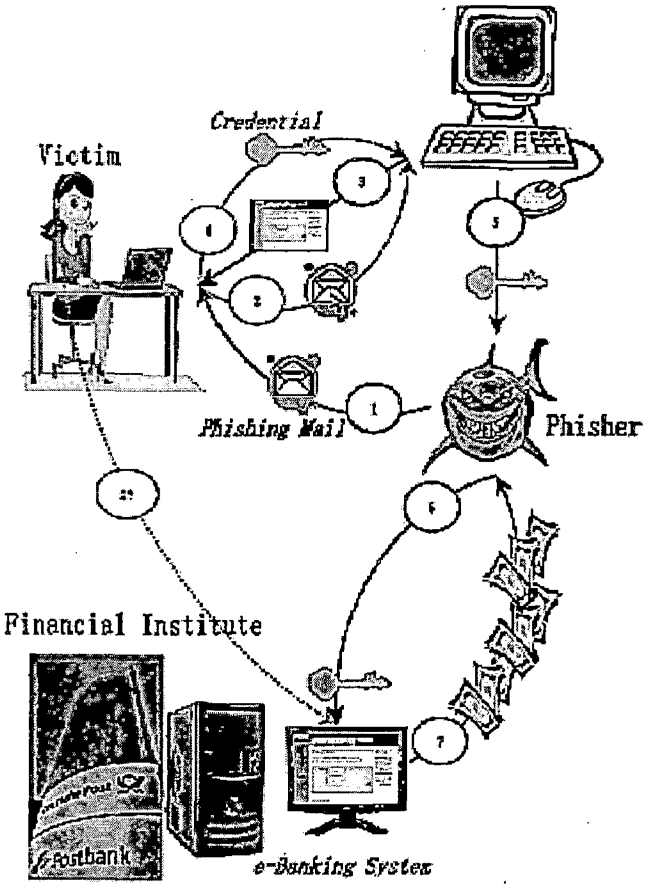


FIGURE 2.4: E-BANKING WORKING PROCESS[14]

Figure 12 shows various steps involved in a phishing process and thus various opportunities to avoid phishing. Many conventional countermeasures already discussed in these steps depend on end users to make a final decision. Because users are not very dependable to properly respond to security indicators/alerts about phishing, these counter-measures may not work as effectively as expected.

An example of using honeypots to counter phishing can be in e-banking, which can be explained as follows.

A duplicate e-banking system could be set up as a phoneypot (phishing honeypot) to trace the way in which phishers use phoneytokens. In this case, these phoneytokens are submitted by the organisation itself to known phishing sites. Such phishers are bound to visit the duplicate e-banking system which is a honeypot, and also draw along with them, other phishers who are looking to attack the organisation. The collected information of the identity of phishers and the nature of phishing could be traced by the phoneypot. The problem with this method might be phishers easily detecting the honeypot nature of the fake e-banking system. To avoid such problems, the honeypot should be available at the same domain name as the original e-banking system and should lead to the phoneypot only on detection of phoneytokens being submitted to the system.

There are also some commercial anti-phishing solutions based on honeypots. The security service "FraudAction", developed by RSA security Inc. exploits a proprietary technology called "Randomized Credentials Technology" and feeds random phishing sites with fake information as baits to track profiles of phisher. But there are known deficiencies in CAPTCHAs which are out of the scope of this thesis discussion.

### 2.3.2 Heuristics and Phishing Blacklists

Detecting phishing at web-site level falls into two categories,

- Heuristic approach
- Phishing Blacklists

Heuristic approaches use HTML or content signatures to identify phish in addition to various rules, and blacklist-based methods use human-verified phishing URLs.

The heuristics approach have both advantages and disadvantages. Heuristics can detect attacks as soon as they are launched, without waiting for blacklists to be updated. However, attacking phishers who come have the knowledge the nature of heuristics might be capable of designing their attacks so as to avoid heuristic detection. In addition, the probability of heuristic methods generating false positives, identifying a non-phishing site as a phishing site, has been known to be high compared to any other existing method.

Another method web browsers use to identify phish is to check URLs against a blacklist of known phish. Blacklisting methods have been in use for a long time in areas other than phishing also. Blacklists of known spammers have been one of the established spam filtering techniques.

There are more than 20 widely used spam blacklists in use today. One possible attribute of blacklisting is IP address. IP addresses of detected spam senders could be blacklisted. Another type of black-listing could be of domain names. The problem with such black listing might be over-caution where the IP address could be that of an ISP (Internet Service Provider) and the spam might only be a diminutive part of data which pass through it. Another deficiency of this method is that it overlooks the possibility of hacked domains. Hackers might gain control of a foreign domain using DDoS attacks and then host spam and phish on the hacked domain.

It is therefore not possible to block the whole domain because of a single phish on that domain. So a blacklist of specific URLs is a better solution in the phishing scenario. The working of phishing URL blacklists should be authentic and open.

A multi-thronged approach needs to be given to building and propagating a phishing blacklist. First, a blacklist vendor enters into contracts with various data sources for suspicious phishing emails and URLs to be reviewed. These data sources may include emails that are gathered from spam traps or detected by spam filters, user reports (e.g. Phishtank or APWG), or verified phish compiled by other parties such as takedown vendors or financial institutions. Depending on the quality of these sources, additional verification steps may be needed. Verification often relies on human reviewers. The reviewers can be a dedicated team of experts or volunteers, as in the case of Phishtank. To further reduce false positives, multiple reviewers may need to agree on a phish before it is added to the blacklist. For example, Phishtank requires votes from four users in order to classify a URL in question as a phish.

Once the phish is confirmed, it is added to the central blacklist. In some instances, the blacklist is downloaded to local computers. For example, in Firefox 3, blacklists of phish are downloaded to browsers every 30 minutes thus enabling the browser to detect and warn users against phishing.

### 2.3.3 Cantina

Cantina is a combination of heuristics and content based guessing.

The heuristics which are often used in evaluating a given URL in the context of phishing re as follows [16]:

*Age of Domain* – This heuristic checks the age of the domain name. Many phishing sites have domains that are registered only a few days before phishing emails are sent out. We use a WHOIS search to implement this heuristic.

*Known Images* – This heuristic checks whether a page contains inconsistent well-known logos. For example, if a page contains eBay logos but is not on an eBay domain, then this heuristic labels the site as a probable phishing page.

*Suspicious URL* – This heuristic checks if a page's URL contains an "at" (@) or a dash (-) in the domain name. A @ symbol in a URL causes the string to the left to be disregarded, with the string on the right treated as the actual URL for retrieving the page.

*Suspicious Links* – This heuristic applies the URL check above to all the links on the page. If any link on a page fails this URL check, then the page is labeled as a possible phishing scam

*IP Address* – This heuristic checks if a page's domain name is an IP address. This heuristic is also used in PILFER.

*Dots in URL* – This heuristic checks the number of dots in a page's URL. We found that phishing pages tend to use many dots in their URLs but legitimate sites usually do not. Currently, this heuristic labels a page as phish if there are 5 or more dots. This heuristic is also used in PILFER.

*Forms* – This heuristic checks if a page contains any HTML text entry form asking for personal data from people, such as password and credit card number. We scan the HTML for <input> tags that accept text and are accompanied by labels such as "credit card" and "password". Most phishing pages contain such forms asking for personal data, otherwise the criminals risk not getting the personal information they want.

In addition Cantina also does some content based conjectures in detecting phishing nature of pages.

## 2.4 Research Gaps

Each anti-phishing technique discussed above suffers from one of the following deficiencies. A careful study of them has led to the below enumeration.

- Two-factor authentication: In this method, the organization managing the website requires a second authentication procedure additional to the one already provided on the website. For examples: authenticating using phone call, mail, etc., the disadvantage of this method is that it requires extra time and effort from the customer and becomes a hindrance for most websites of non-financial operations. Mostly used websites like social networking sites, mail providers, gaming websites cannot afford to adopt 2-factor authentication because of the delay involved.
- Email Filters: Neural networks are generally deployed to detect terms and phrases generally used in phishing emails. This is more of a guessing game. And phishers who intelligently follow the development of these methods always find loopholes in the methods. For example using images instead of examples to show the company Logo, slight change in the spelling of words would evade the email filters. The most important problem with email filters though is false positives.
- Blacklists: Looking up a given URL in a standardized blacklist. A lot of human intervention is needed before a URL goes onto the blacklist. This is not completely reliable.
- CANTINA : The issues with Cantina are as follows –
- The first step of CANTINA is heuristic validation. So the problem of false positives haunts CANTINA like any other heuristic approach.
- Concept of lexical signature: Cantina tries to assign a lexical signature to each web-page and tries to detect the page with that signature. The lexical signature is 5 terms from a page. As I would show below, the method adopted to determine this lexical signature is not very convincing.

- TF-IDF: The IDF which was used in CANTINA is not clear. For using TF-IDF, we need to calculate TF (term-frequency) and IDF (Inverse Document Frequency) of each term in the documents. The concept of IDF comes from a collection (or corpus in Data Mining lexicon) of documents. But CANTINA hadn't really used any such corpus. They used the British National Library's collection of most frequent English words used. This corpus need not heed any resemblance to the frequency distribution of the terms on the web.
- Using URL/domain-name of the page in the lexical signature. In addition to the five terms, CANTINA used the URL of the page as a part of the lexical signature. But even URLs of phishing sites are tricky. A phish of **login.ebay.com** may lookalike **ebay-login.secure.com,** which would lead to errors in the results.
- Some very basic cases are overlooked in CANTINA. For example, let us consider the url, google.co.in. If this page is evaluated by CANTINA, it takes five terms and submits to google.com search engine. These terms are the same in all pages like google.com, google.co.in, google.co.uk, google.co.hk, etc., The results of the submission would include domain names mostly from google.com. If the starting N results are considered, CANTINA cannot find google.co.in and marks google.co.in as a phishing site, which would be disastrous.

# 3.Proposed Solution

Instead of approaching the problem of phishing in so many different ways, a simple and straightforward way of attacking the origin of phishing itself could be developed. One thing could be agreed upon about all kinds of phishing attacks. Phishers use some means to lead users to phishing websites, where the actual phishing occurs. If we could deter the phishers' attempts by detecting a phishing site before the user submits his information, then the problem is easily solved. The solution sounds easy, but the immediate question arises as in "why wasn't this already implemented?" Many tools have been implemented to detect phishing and they have been introduced in the Literature Review section. But none of the tools came to prominence for reasons ranging from low detection-rate to high false-positives. As discussed in section 2.4, until now phishing-site detection systems have been largely using heuristic methods to warn users of phishing-sites. Heuristics are effective, but not sufficient. The method which we are going to propose uniquely identifies the exploitation of phishers and directly attempts to break the loophole in preexisting anti-phishing methods. Before diving into the method, let us look into the concepts on which the proposed method is based on.

## 3.1 Content, URL and Authority

What makes users believe that a phishing site is a legitimate site? It is the content of the site. The content of any website, phishing or legitimate, comprises of html content. But in the view of a user, a website is a rendering of text, images, forms,. presentations, etc. The phisher is aware of this fact and architects the phishing site in such a way that the content (text, images & forms) of it perfectly matches the content of the original page. The only difference between the two pages is the authority of the page. The organization which owns the page also owns the intellectual rights on the content it provides on its page. However the phisher illegally copies the content onto his page. One obvious difference between the phishing and original site is the URLs they carry. So, if we can authorize a given page with the help of its content and URL to a necessary accuracy, we can identify whether or not the page is a phish. Nevertheless, the content of a phishing site and that of an original site are the same. The method being proposed here exploits this particular "similarity of pages" to identify phishing sites.

## 3.2 Similarity of Web Pages

We define similarity as follows:

- Given two web pages p1 and p2, we say p1 and p2 are similar to each other if and only if an average user finds it improbable to differentiate between the two pages when he has zero access to the URLs of the webpages.

The concept of similarity seems simple. Nevertheless in order to avoid future confusions we need to have an unambiguous definition of similarity of web pages. If the definition is carefully observed, one can notice that similarity here is a subjective concept. It depends upon the preconceptions of the user under consideration. Factors like knowledge of security systems, experience in web browsing and attention span of the user might affect the decision of whether two pages could be called similar. In addition different concepts exist in the name of similarity. A recent application developed by Google Inc., for their chrome web browser involves detection of similar websites given a website. Here similar websites mean websites where one can perform actions similar to those one can perform in the original website.

The concept of similarity is crucial in identifying fake websites. Let us say two websites w1 and w2 are completely similar. The explanation to such a situation could be one of the three following ways:

- Both belong to the same authority, and each is closely related to the other.
- One is an original site and the other a phishing site attacking the original.
- The similarity is coincidental.

Of the three cases above, the last is the least probable and can be neglected.

## 3.3 Site Population

Site-population (SP) is a property we attribute to each domain name. For example, the domain of the website of State Bank of India is www.statebankofindia.com. In finding the SP score of a domain name, we represent the structure of a website as a graph (a tree with additional back links) with the domain name as the root of the graph. Each edge emanating from a node of the graph represents a hyperlink out of the page to a different page belonging to the same domain.

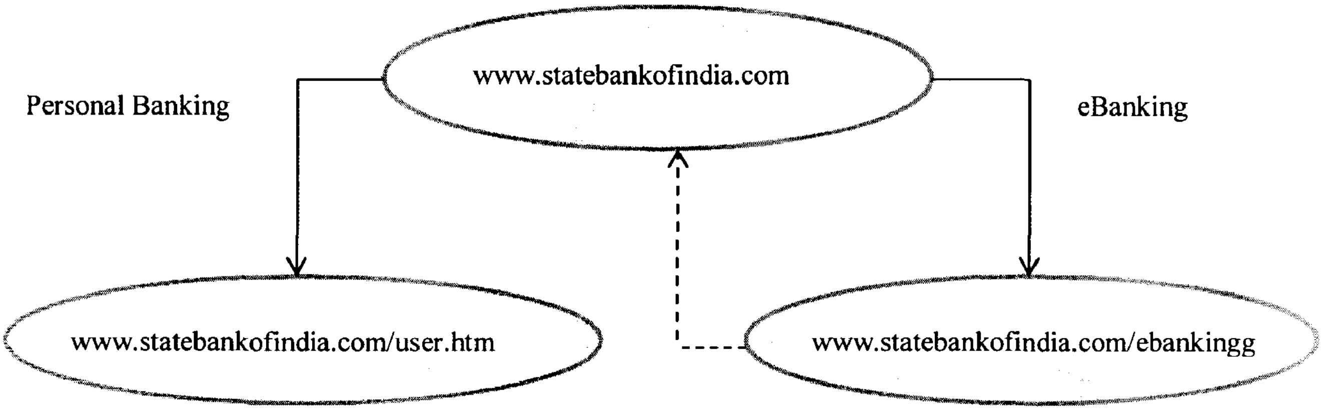

www.statebankofindia.com
Personal Banking
eBanking
www.statebankofindia.com/user.htm
www.statebankofindia.com/ebankingg

FIGURE 3.1: HYPERLINK GRAPH EXAMPLE

We say the original page corresponding to the domain name www.statebankofindia.com is at depth 0. And the two other pages shown are ate depth 1. In addition sites in the World Wide Web also have back links, which link to the root page from lower pages.

The site-population of the domain name www.statebankofindia.com is the total number of unique nodes in the hyperlink graph of www.statebankofindia.com.

### 3.3.1 K-Site-Population

K-Site-Population or kSP of a domain name is defined as the total number of unique nodes in the hyperlink graph of the domain name not considering the nodes with depth greater than or equal to k.

Now let us compare the SP scores of legitimate sites to that of phishing sites. A general trend in phishing sites is to create one page like that shown in Figure 4 similar to an original site like that in Figure 3 but to lead every link that goes out of the phishing page to the corresponding links of the original page [5]. i.e., in a phishing site of eBay.com, which has a different domain name, all other links like careers, products, forgot password, etc., lead to the corresponding links of the original eBay page with domain name ebay.com. This in turn results in much lesser SP scores of phishing sites compared to those of original sites. In general it is observed that more than 90% of phishing sites have their SP score less than 2, i.e., they only have the page corresponding to the root node in the hyperlinks graph.

## 3.4 Proposed Theory

The whole method of detecting phishing pages could be spelled with the below single sentence:

- Given a large database of webpages and a given page of the database P and a set of pages similar to P say S, if the domain name of P has the lowest SP score of all the domain names corresponding to pages in S, then P can be unambiguously labeled a phishing site.

The proposed method leaves two problems to be faced.

- How to calculate the SP scores?
- How to find similar pages in a database of pages?

SP scores of websites could be calculated by crawling the given site. But much effort is needed in identifying similar websites. In this method, we consider similarity of websites by establishing similarity of content of the websites. Furthermore, we take into consideration only the text part of the webpages considered. Now the problem of identifying phishing sites decomposed itself into identifying websites with similar text content. We approach this problem in a way which search engines approach the problem of searching in a given corpus of documents. The core of the method involves the famous text mining algorithm, TF-IDF.

We have seen the definitions of Term Frequency and Inverse Document Frequency in section 2.4. Here, we shall extrapolate these concepts to establish a quantitative measurement of the similarity between two webpages.

# 4. Implementation

A sophisticated and structured system has been developed to implement, test and obtain the results of the proposed solution for a content-based Anti-Phishing system.

## 4.1 Input Data

An authentic corpus of websites has been given as the input to the system.

In a system similar to the web, web-pages could be classified into three types.

- Legitimate sites, for which no corresponding phishing sites exist.
- Legitimate sites, which are duped and suffer phishing attacks.
- Illegitimate sites which are dups of legitimate sites.

Care is taken so as to include all the above types of websites in the input considered. A continuously updated list of detected phishing sites is provided by www.phishtank.com. In addition to the URLs of phishing site, the JSON (JavaScript Object Notation) objects provided also contain the information about the entity that has been attacked by the mentioned phishing URL. The input database consists of an impartial combination of phishing sites and also the corresponding original sites. In order to include sites which are original but have no phishing pages, an unsupervised list of websites was chosen from those of the top commercial banks of India.

Challenges faced:

In general, crawlers tend to crawl through any hyperlink in a given page. In reality every hyperlink need not be an html page. Many extraneous files like pdf documents, images, audio and video files could be hyperlinked in an html page. So before jumping to parse a given document for html, care should be taken that erroneous files are not downloaded, otherwise which majority of bandwidth and time could be wasted. Also since the SP score of some websites could be huge, a hyperlink-graph-depth of 5 was chosen in downloading the pages, which is a good approximation of the complete hyperlink-graph.

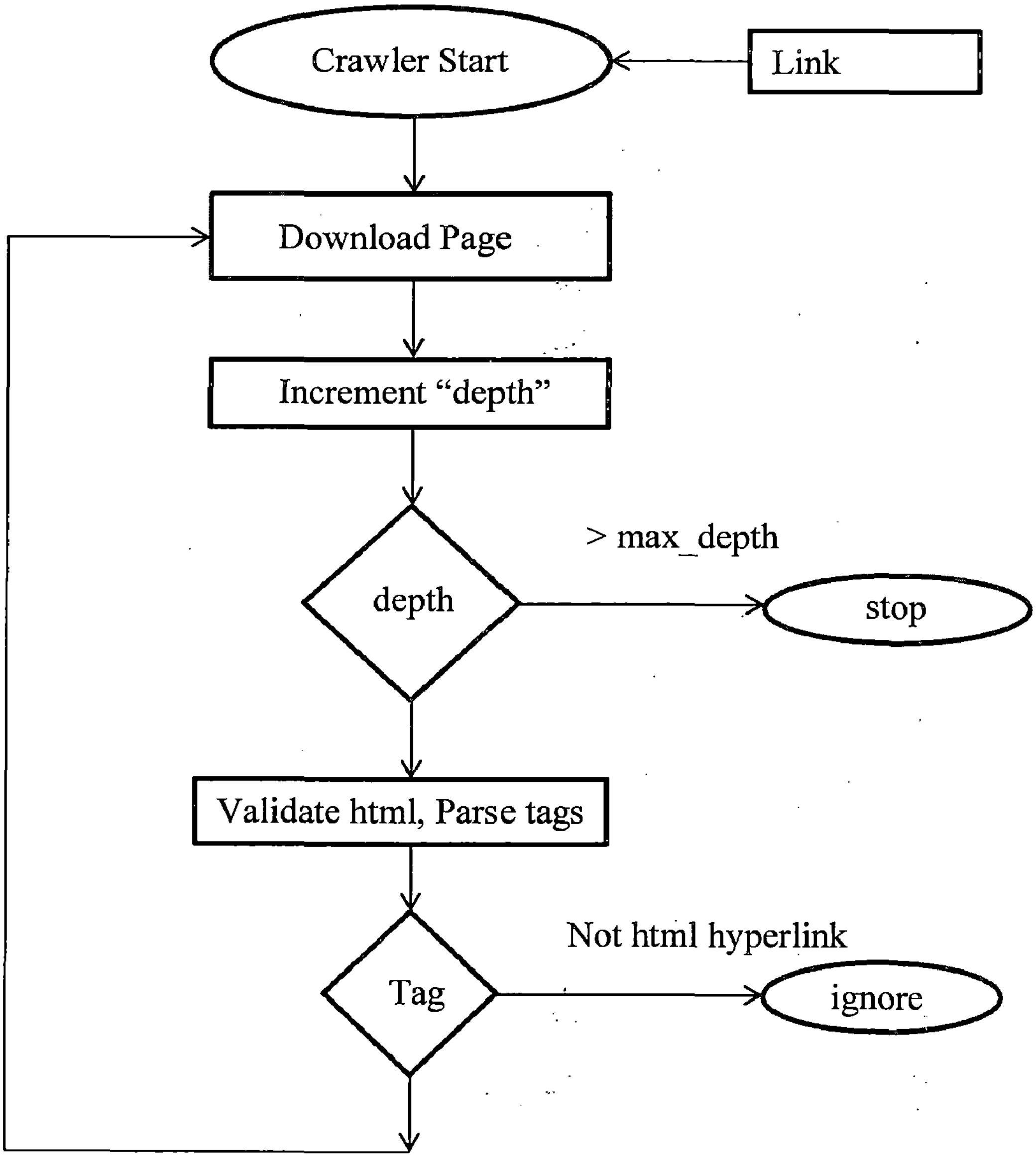


FIGURE3.2: WEB CRAWLER FLOWCHART

## 4.2 Parsing and Preprocessing

The UNIX tool w3m was designed so as to render the text as viewed by the user. Thus this extraction becomes a perfect tool for our purpose.

A preprocessor is defined as a program that processes its input data to produce output that is an input to another main program. Before deploying a text mining technique, preprocessing becomes essential to the input text data. Careful tokenization is done on the text.

### 4.2.1 Stemming

Stemming or lemmatization is the technique for the reduction of words into their root[17]. Many words in the English language can be reduced to their base form or stem e.g. agreed, agreeing, disagree, agreement and disagreement belong to agree. The s from the end of most of the words in plural in English language could be removed to get the singular form of the word, which is also the root. The variation "Peter's" in a sentence is reduced to "Peter" during the stemming process. The result of the removal may lead to an incorrect root. In our case, the preprocessed text doesn't interact with the user however. When we search for similar files of a given file, we preprocess the queried file also. The stem is still useful, because all other inflections of the root are transformed into the same stem. Case sensitive systems could have problems when making a comparison between a word in capital letters and another with the same meaning in lower case. Following a selection of suffixes and prefixes for removal during stemming:

suffixes: ly, ness, ion, ize, ant, ent , ic, al , ical, able, ance, ary, ate, ce, y, dom , ed, ee, eer, ence, ency, ery, ess, ful, hood, ible, icity, ify, ing, ish, ism, ist, istic, ity, ive, less, let, like, ment, ory, ty, ship, some, ure

prefixes: anti, bi, co, contra, counter, de, di, dis, en, extra, in, inter, intra, micro, mid, mini, multi, non, over, para, poly, post, pre, pro, re, semi, sub, super, supra, sur, trans, tri, ultra, un

However, most stemming algorithms do not remove the prefix of a term. The reason for this is the huge impact for the meaning of a sentence. For instance, stemming the word nonhazardous to hazardous is an unwanted change. There are different types of stemming methods. The simplest one is the brute force method. This method requires a dictionary which contains the inflections of a word. The dictionary is used as a lookup table. This approach has some serious disadvantages. Firstly the speed for the word search is very low and the whole stemming process requires many resources in storage. This is the result of a missing algorithm which could increase the transformation speed. The other disadvantage is the problem that the look-up table usually does not contain all inflections for each root. The need for a comprehensive dictionary is fundamental for acceptable results. The quality of the result entirely depends upon it.

In the current implementation a modern stemming algorithm for stemming, Porter's Stemming Algorithm is used.

In our implementation, we remove capitalization of text in our database. We only consider terms in their raw form. Also any rare characters are not taken into the context of our search. Only word characters, which are alphabets and digits, are considered.

### 4.2.3 Stop Word Removal

Stop words are the extremely common words which occur in a language and thus in documents which contain text of the language. Search engines neglect the effect of stop words because, otherwise if the stop words are taken into consideration, the data considered would be more populated with stop words than the relevant search terms. Since we attempt to implement search capability and thus implement similarity, we need to remove stop words from the files in our database. The stop words list used here is an authentic list of most frequent English words used on the web provided by the British National Corpus.

## 4.3 TF-IDF

The following data structure has been used to maintain the term-frequencies and inverse-document-frequencies of terms in a preprocessed database of text files. Since implementation of the whole system is done in python, a python dictionary is used to store the term-frequencies.

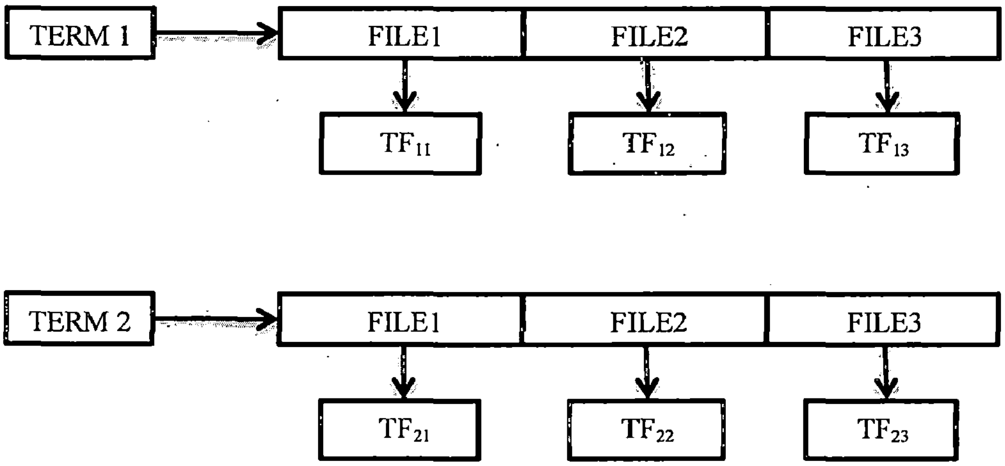


FIGURE 4.1: TERM-FREQUENCY TABLE

$TF_{ab}$ stands for the term-frequency of the term TERMa in file FILEb.

The input to the table generation program is the root directory of the preprocessed pages. The output being the above data structure populated with all the terms and files in the input directory, traversed recursively. The above table is constructed as a python dictionary as follows:

```
def update_table(flist):
  global table
  table = dict()
  for inputfile in flist:
    docf = open(inputfile)
    docterms = docf.read().split()
    for term in docterms:
      if table.has_key(term):
        if table[term].has_key(inputfile):
            table[term][inputfile] += 1
        else:
            table[term][inputfile] = 1
    else:
        table[term] = dict()
        table[term][inputfile] = 1
```

From the above table, deriving Inverse Document Frequencies is straightforward. For each term, calculate the number of files it exists in, say $T_{all.}$ IDF of T is equal to (1 + TotalFIles)/$T_{all.}$

In python, this could be expressed as

```
global idflist
    idflist = dict()
    for term in table.keys():
        idflist[term] = math.log( (float(N) +  1)/ (term.__len__()))
```

Challenges faced:

In a system where webpages are being crawled and updated continuously, we cannot afford to rebuild the term-frequency table after each update of the database. In order to avoid such updates, a persistent data structure is maintained, using python's PICKLE technology. Every update of the preprocessed database calls for the loading of the term-frequency table from the disk to memory and updates to TFs are done to entries corresponding only to the new terms and files being added to the database.

## 4.4 Search and Similarity

The formulation of the theory behind similarity search could be done incrementally as follows:

Given a term T, what can we say about the occurrence of this term in the Database of our preprocessed documents? How can we associate files to this particular term?

If a single term is considered, it has a uniform IDF across all documents. So the only variant property is the TF. Thus, a file with the highest TF associated with the term T is also the file most distinguished by the term T.

Now, let us consider a group of i terms, $T_I$ to $T_i$. Now, how to find out the files which are best characterized by the aforementioned group of terms? We apply a ranking mechanism by which we assign a score corresponding to the set of terms to every file in our database. Here we apply the core concept of TF-IDF. For each term T, the TF-IDF score corresponding to a given file is the product of TF in the file and IDF of T.

$$\textbf{TF-IDF}_{ij} = \textbf{TF}_{ij} * \textbf{IDF}_i$$

The score of a file F corresponding to the set of terms $T_I$ to $T_i$ is

$$\textbf{TF-IDF}_{F(T1-Ti)} = \sum \textbf{TF}_{iF} * \textbf{IDF}_i$$

The problem with the above formula is that, if a particular term $T_j$ has much higher TF compared to the other terms in a file, the TF-IDF score of the file will unanimously be dependent upon the term $T_j$. So we use a normalized form of the term-frequencies

$$\textbf{TF}_{ij} = \textbf{tf}_{ij} / \textbf{harmonic\_mean}(\textbf{tf}_{1j}, \textbf{tf}_{2j}, \textbf{tf}_{3j}, \ldots \textbf{tf}_{ij})$$

Thus, the TF-IDF score is assigned to each file with respect to the given set of terms S. The result of sorting all the files in our database according to the TF-IDF score is the search result for S. Thus given a file F, in order to find other files which are similar to this, we extract the text from the file, preprocess it, query the database and rank the files based on each files TF-IDF score with respect to this text. We say that the files with the TF-IDF score closest to the TF-IDF score of the file whose text has been considered for search is the most similar to it. But this is still an incomplete measure of similarity. Let us consider two files F1 and F2 where F2 is a minor part of F1. After extracting the text from F2, we calculate TF-IDF scores of F1 and F2 with respect to text in F2. But since F2 is present in both the files, both the TF-IDF scores are nearly identical. But F1 and F2 are not similar according our definition of similarity.

In order to establish similarity of two files F1 and F2, we need two conditions,

- Content of F1 should be present in F2
- Content of F2 should be present in F1

### 4.4.1 Similarity Score

$$\mathbf{SS_{11} = 1 / diff\{diff\{TF\text{-}IDF_{12}, TF\text{-}IDF_{11}\}, diff\{TF\text{-}IDF_{21}, TF\text{-}IDF_{22}\}\}}$$

Diff{a, b} stands for the difference of the two entities.

The formula for SS has been thus defined so as to achieve global maximum when the two files have exactly the same text. And minimize when intersection of the two files have no text at all.

Once the search for similar pages is done, we need to establish the authority of the web pages. From section 3.3, we can say that in general phishing websites have much less kSP score or k-site-population compared to the original websites. So we can make a conjecture as in

- Of two similar websites, the websites belonging to the domain name with higher site-population has a higher probability of showing up in search results.
- Of the list of similar pages achieved only those with higher or relevant similarity are to be chosen.
- Of the relevant search results we should find if our present page exists or not.

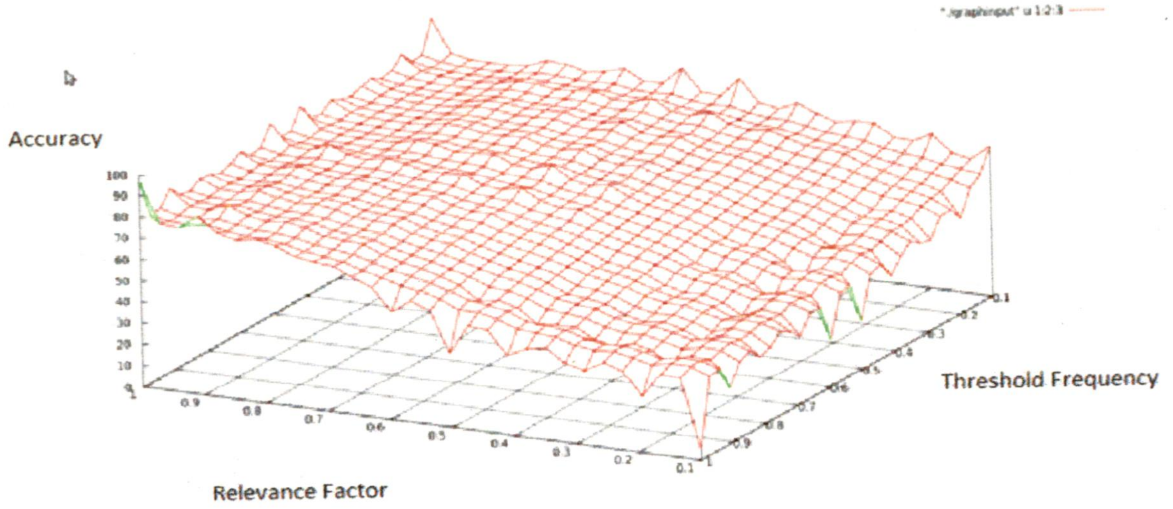
# 5.Results

In this section, we present and compare the performance of our method in various dimensions.

We have tested our algorithm and implemented the system with an authentic database of webpages. As specified in Section 4.1, web pages of all the three kinds are taken into consideration. The phishing sites which were considered were taken from phishtank.com where the websites are multiple times verified by users. A number of original sites were downloaded without ambiguity. Some legitimate sites downloaded are the most attacked websites according to phishtank.com. Other legitimate sites which have no corresponding phish are downloaded as the websites of most popular commercial banks in India. Total number of webpages which are given as input to the system is in excess of 1000.

Two factors were introduced to fine tune the search results: Relevance-Factor and Threshold-Frequency. Given a page the decision of whether this is phishing or not is decided in the following manner,

- Rank all pages in the database in decreasing order of Similarity Score.
- Eliminate the results with similarity-score < (relevance-factor * similarity-score)
- Initialize "count" to number of pages in the result with the domain name same as that of the page in consideration.
- If the ratio of count and total results is more than threshold-frequency, then the page is declared "NOT PHISHING", else "PHISHING".



SFIGURE 5.1: PLOT OF ACCURACY WITH THRESHOLD-FREQUENCY AND RELEVANCE-FACTOR

## 5.1 Comparison

The comparision of various tools is being done in the following manner. Information about the performance of pre-existing tools has been taken from [18]. In addition the performance of our system has been compared with CANTINA [16]. According to the sources, the performance analysis has been done for all the tools using urls from phishtank.com, which is the same as ours. For the tools analysed by [18] and CANTINA exactly 100 phishing URLs were taken as the input. Of the 1000 URLs we used, more than 100 URLs are phishing URLs.

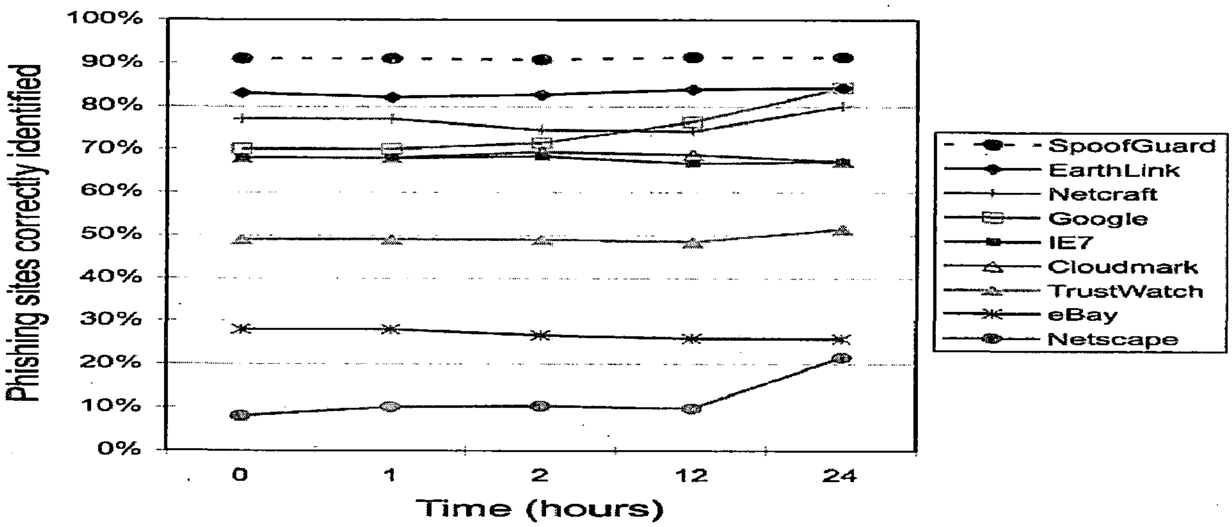


FIGURE 5.2: TRUE-POSITIVE OF PREVIOUSLY DEVELOPED METHODS[18]

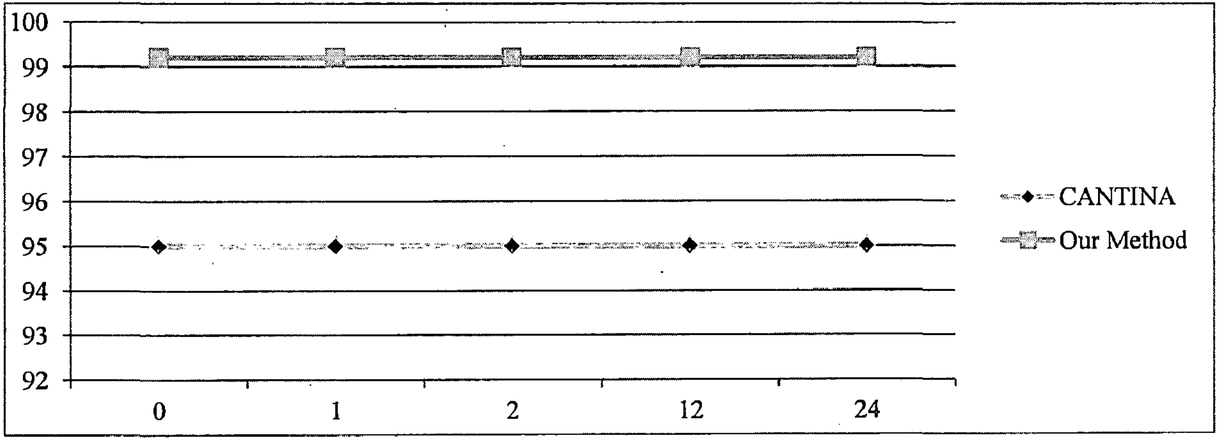


FIGURE 5.3: TRUE-POSITVE RATE WITH RESPECT TO URL ALIVE-TIME

As you can see our method is time invariant since we don't depend on blacklists, which are stronger with time. We achieved almost complete accuracy in detecting a site as a phishing site with 99.26%.

A more important feature of anti-phishing tools is false-positive rate. More than detecting the right phishing sites, tools should take care that legitimate sites are not marked as phishing. Information about false-positive rates of tools were taken from [18].

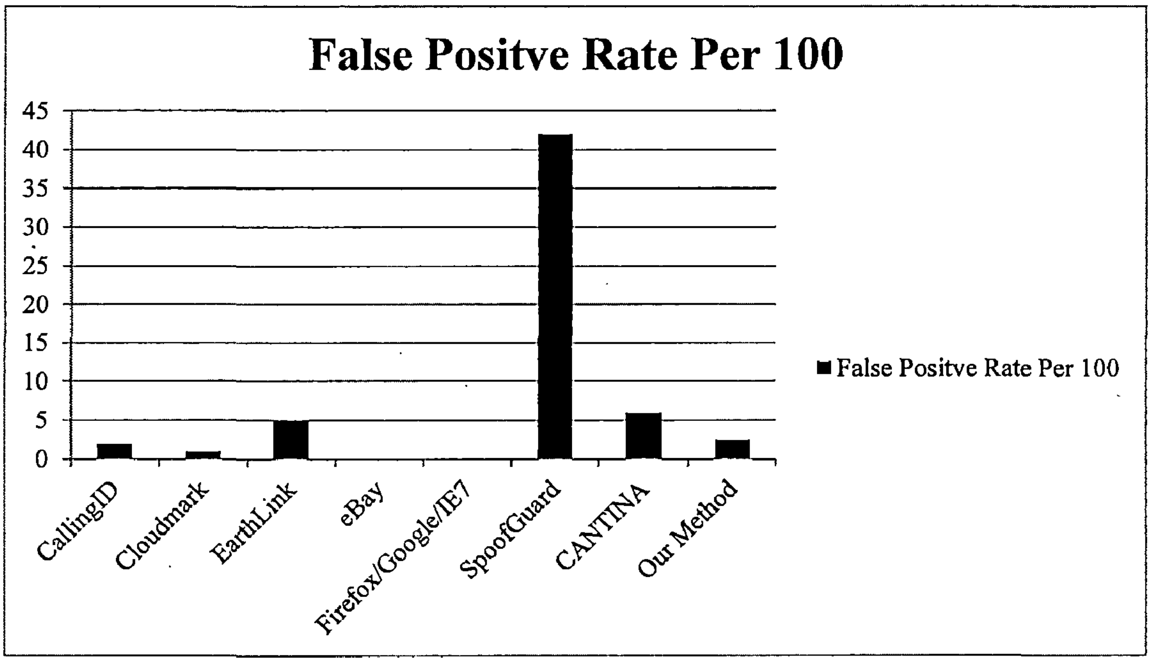


FIGURE 5.4: FALSE-POSITIVE COMPARISION

A false-positive rate of 1.94% was achieved, which is the lowest of non-blacklist type Anti-Phishing methods. We have to remember than blacklists use user-confirmed lists of phishing sites, hence have the lowest false-positive rate.

# 6. Conclusion and Future Work

In this dissertation report, we proposed an efficient solution to the problem of phishing based on content similarity. The best thing about a content based phishing approach is that it is ubiquitous. The future is going to be data-centered with new ways of accessing, sharing and publishing data. Any data on the Internet could be bait for phishing. New ways of propagating phishing URLs are emerging, like XSS, JavaScript injectors, etc., the domain of phishing might itself change. To build a robust and stable anti-phishing system, the basics of phishing, which is existence of a duplicate entity which plagiarizes the content of the true entity, phishing detection systems need to base their results on the content, instead of using heuristics, fuzzy logic, etc., After all heuristic techniques wear away with time. Thus we advocate once again that content based anti-phishing is the solution to be explored more than other approaches.

Proving the above logic, we have shown that our similarity based method gives much better detection of phishing sites with comparatively lower false-positive rate. In the current scenario, organizations which implement web-search-engines like Google, Microsoft have enormous amounts of web data and enough search techniques. The future of phishing lies in the hands of such organizations which should develop content based anti-phishing tools. A browser toolbar which could communicate with such search-engine would be an ideal anti-phishing tool.

## 6.1 Future Work

After implementing the presented system, we could come to only one conclusion. That it is far from complete. Much work is needed to refine and redefine the components of the system. Using python as a programming language, in general leads to slower performance compared to systems built with C++. So we look forward to implement the system in C++ and compare the performance.

With exponentially increasing amounts of data and not-so exponentially (according to Moore's Law) increasing amounts of processing power, we look forward to parallelize our algorithm on fast multi-processors and GPUs.

# REFERENCES

[1]. Gartner, Inc. "Gartner Survey", www.gartner.com, 2007.

[2]. A Inomata, M Rahman, T Okamoto, "A Novel Mail Filtering Method against Phishing", Communications, Computers and Signal Processing, PACRIM, IEEE Pacific Rim Conference, 2005.

[3]. U.S. Department of Justice. "U.S.Code, Chapter 35, Article III", § 3542.

[4]. Anti-Phishing Working Group(APWG), "Phishing Activity Trends Report", 2010.

[5]. Rachna Dhamija, J.D.Tyagar, Mart Hearst, "Why Phishing Work's", Proceedings of the SIGCHI conference on Human Factors in computing systems, 2006.

[6]. www.phishtank.com phish-Id 1208582.

[7]. Fahrenheit Marketing Inc., "Latest Statistics on Email Spam". 2011.

[8]. Adrienne Felt, Davind Evans, "Privacy Protection for Social Networking Platforms", Workshop on Web 2.0 Security and Privacy, Oakland, CA, 2008.

[9]. Wetzel, Rebecca. "Tackling Phishing", Business Communication Review, 2005.

[10]. Sonic Wall. "Phishing And Spam". www.sonicwall.com, 2007.

[11]. http://safeweb.norton.com/

[12]. The Anti-Phishing Working Group, "Proposed Solutions to Address the Threat of Email Spoofing Scams White Paper", 2003.

[13]. http://en.wikipedia.org/wiki/File:SMTP-transfer-model.svg. SMTP.

[14]. Spitzner, L."Honeypots: Tracking Hackers", Addison Wesley, 2002.

[15]. Shujun Li, Roland Schmitz, "A Novel Anti-Phishing Framework Based on Honeypots", Proceedings of 4th Annual APWG eCrime Researchers Summit (eCRS) 2009.

[16]. Yue Zhang, Jason Hong, Lorrie Cranor. "Cantina: A Content Based Approach to Detecting Phishing Websites". Banf, Alberta, Canada : s.n., 2007.

[17]. Keno Buss, "Literature Review on Preprocessing for Text Mining", Software Technology Research Laboratory, http://www.cse.dmu.ac.uk/STRL/, 2007.

[18]. Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong, "Phinding Phish: Evaluating Anti-Phishing Tools", Carnegie Mellon University, USA, 2006.