PREVENTING DIGITAL IDENTITY THEFT USING FUNDAMENTAL CHARACTERISTICS

A Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Tanmay Thakur

August 2014

PREVENTING DIGITAL IDENTITY THEFT USING FUNDAMENTAL CHARACTERISTICS

Tanmay Thakur

APPROVED:

Dr. Rakesh Verma, Committee Chairperson
Dept. of Computer Science, University of Houston

Dr. Weidong Shi (Larry)
Dept. of Computer Science, University of Houston

Dr. Dan Wallach
Dept. of Computer Science, Rice University

Dean, College of Natural Sciences and Mathematics

This material is based upon work supported by the National Science Foundation under Grant No. CNS1319212. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Acknowledgments

It gives me immense pleasure in expressing my regards to the people who supported me and helped me in making my thesis possible. First and foremost, I would like to thank my thesis advisor Professor Rakesh Verma. Every part of this thesis is a result of his continued guidance and encouragement, without which it would not have been possible. I would like to thank my thesis committee members Dr. Weidong Shi and Dr. Dan Wallach (Rice University) for keeping me on the right track thought out the duration of my thesis.

I would like to thank Earl Lee at Yale University for working closely with me on different modules and helping me integrate those modules in the project. Special thanks to Deevakar Rogith, University of Texas, for critiquing my work and correcting me at each step and giving me access to a good server to run my scripts. The thesis would have been incomplete without timely suggestions and comments from my colleagues, Keith Dyer, Vasanthi Vuppuluri, and Luis Felipe at ReMiND (Reasoning, Mining, Natural Language Processing and Defense) lab. I would like to thank Pranav Koundinya for accompanying me during food and exercise breaks. I am thankful to Jackie Baum for assisting me in all the administrative work.

I am glad that organizations like PhishTank, Alexa Internet, Anti-Phishing Working Group, and DMOZ have made their resources publicly available. It was very helpful to retrieve dataset and trends related to phishing domain.

Finally, my sincere regards to my parents and the Almighty for encouraging and supporting me through each milestone of my academic journey.

PREVENTING DIGITAL IDENTITY THEFT USING FUNDAMENTAL CHARACTERISTICS

An Abstract of a Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Tanmay Thakur

August 2014

Abstract

The social engineering strategy, used by cyber criminals, to get confidential information from Internet users is called *Digital Identity Theft*. It continues to trick Internet users into losing time, money and productivity. A common way to steal digital identity is through phishing. The trends and patterns in such attacks keep on changing over time and hence the detection algorithm needs to be robust and adaptive. Although, many attacks work by luring Internet users to a webs site designed to trick them into revealing sensitive information, recently some attacks have been found that work by either installing malware on a computer or by hijacking a good web site. This thesis presents effective and comprehensive classifiers for both kinds of attacks, classical or hijack-based, with a focus on the latter. According to the literature study, this seems to be the first to consider hijack-based phishing attacks. This thesis focuses on the fundamental characteristics of target websites, attacked websites and introduces new features and techniques for detection. Some of the techniques are equally effective for zero-hour phishing web site detection. It presents results of these classifiers and combination schemes on datasets extracted from several sources. It is shown that the content-based classifier achieves good performance despite the difficulty of the problem and the small size of white list. One of the combination schemes achieved detection rate of over 92% for phishing web sites with false positive rate of less than 0.7% (without Internet search) and 0% false positive rate is also possible with reasonable detection rate of over 74% (with Internet search). Moreover, the classifiers presented are also language independent.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Definition of Digital Identity Theft

Digital identity is information that a WWW user has or knows, e.g., credit card numbers, passwords, etc., that is used for authentication purposes. Digital identity theft is a threat aimed at gleaning digital identities of unsuspecting victims.

According to FBI [12], *identity theft occurs when someone assumes victim's identity to perform a fraud or other criminal act. Criminals can get the information they need to assume victim's identity from a variety of sources, including by stealing the wallet, rifling through their trash, or by compromising their credit or bank information. They may approach victim in person, by telephone, or on the Internet and ask for the information.*

A common way to steal digital identity is through phishing. Attackers typically

lure Internet users to a web site designed to trick them into revealing their identity. The identity is then sold or used to make purchases without the knowledge of the victim. The victim faces not only monitory loss but also time is wasted to prove the identity theft.

Many phishing web pages are copies of some version of a legitimate site such as PayPal or eBay. Some offer money or prizes as incentives. Users are typically attracted to phishing pages by sending them warning or enticing emails, or by posting URL links in forums, social networking sites, chat and bulletin boards.

1.2 Phishing Market and Attacks

Gartner is a renowned information technology research and advisory company. According to a survey by them, $3.2 billion was lost by phishing victims from United States alone in 2007. In the same year, 3.6 million web users fell victim of such attacks [18]. The number of such victims kept on increasing to 2.3 million in 2006, 3.6 million in 2007 and 5 million in 2008. Gartner surveyed 3985 phishing victims from US in 2008 and found that the average monetary loss per incident was $351 [17]. Typically, victims could recover just 56% of their loss. Many anti-phishing solutions were proposed and used but the cyber criminals keeps adapting to hide from such techniques and still in 2014 the problem persists.

Recently, new phishing attacks have surfaced that involve either installing malware on a computer [40] or hijacking a legitimate web site. For example, netcraft reported that the website of the Agency for the Safety of Aerial Navigation in Africa

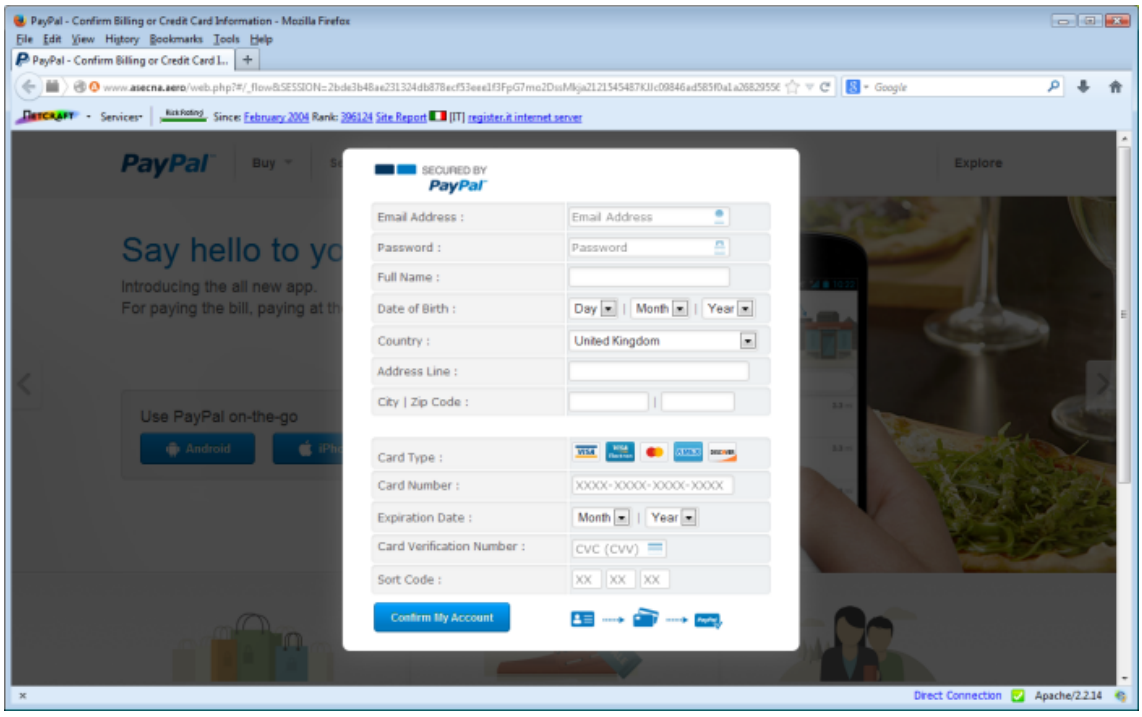

Figure 1.1: Hijack based Attack

and Madagascar (ASECNA) was hacked in an April 2014 report.[33]. When a user went to the homepage, it asked for PayPal account details. After entering it, victims were redirected to actual PayPal website. Figure 1.1 shows that the URL looks legitimate and hence the URL classifiers will fail. According to the in-depth literature study in phishing domain, no work is found in detection of such attacks and hence the thesis work can be said as first detection system for such attacks. The thesis focuses on such **Hijack-based attacks** along with the classical phishing patterns.

1.3 Need of Detection and Motivation

The urgency for efficient and reliable detection schemes becomes clear upon considering the phishing activity trends in the second quarter of 2013 (April-June 2013, Q2 2013) as published by the Anti-Phishing Working Group (APWG) Reports [3]. The APWG definition of phishing encompasses both social engineering and technical subterfuge to steal consumers' digital identities. Social-engineering schemes include spoofed e-mails that point consumers to counterfeit websites and technical schemes include malware planted onto PCs to steal credentials directly. Given this definition, the APWG reports that the number of unique phishing websites detected in Q2-2013 reached a high of 44,511 in May 2013. Payment Services accounted for a large percent of attacks with the number of phished brands reaching a high of 639 in Q2-2013. United States continued its position as the top country for hosting phishing websites during this time. The percentage of computers infected with banking trojans and password thiefs also rose.

Besides the loss of time and productivity, estimates of money lost every year in phishing attacks run from several hundred million to billions of dollars. Therefore, detection is an important challenge for researchers. The trends in patterns of the phishing sites keep changing and hence, it creates a need of a robust algorithm that is not just specific to the given dataset. Also, the lifetime of phishing sites is very short. *On an average a phishing domain lasts 3 days 31 minutes and 8 seconds* [29]. An algorithm should also be fast enough to catch such pages before they do their job. Finally, the algorithm should rely on fundamental characteristics of phishing

web sites.

1.4 Analysis of Latest Phishing Activity

Anti-Phishing Working Group, Inc. (APWG) is the worldwide coalition unifying the global response to cyber-crime across industry, government and law-enforcement sectors. [3]. They produce quarterly reports on the phishing activity trends.

Following are some of the statistics of latest trends in the Q1- first quarter (January to March) of 2014 [4]. Q1 had 125215 total number of phish, which is a 10.3% increase over Q4-2013 (Figure 1.2). Total 171792 unique phishing reports were submitted to APWG during Q1 (Figure 1.3). Around 557 brand names were targeted with payment services as a main focus (Figure 1.4). United States continued to be the top most country hosting phishing sites (Table 1.1). About 3% phishing sites used IP address and 46% used *.com* as TLD (Figure- 1.5. Table 1.2 summarizes the highlights for the first quarter of 2014.

1.5 Contribution of the Thesis

As pointed out in [48], *phishing patterns evolve constantly, and it is usually hard for a detection method to achieve a high true-positive rate while maintaining a low false-positive rate*. Existing phishing site detection methods fall into one of the following categories: URL matching against human-verified blacklists, heuristics used with machine learning, password based, or some combination of information extraction

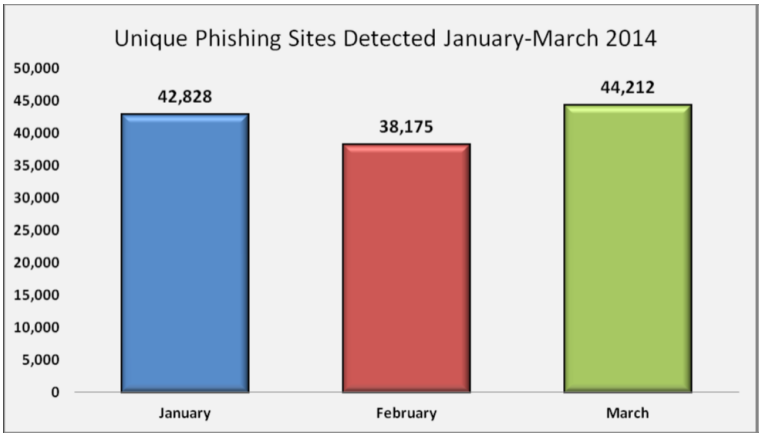

Figure 1.2: Unique Phishing Sites Detected in Quarter-1: 2014

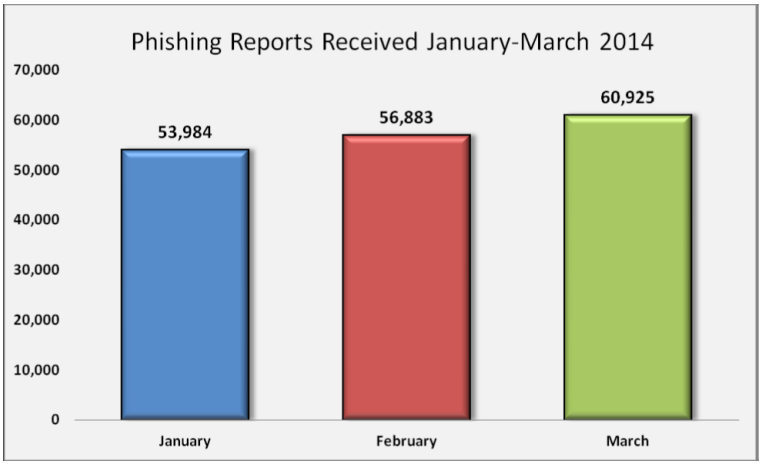

Figure 1.3: Phishing Reports Quarter-1: 2014

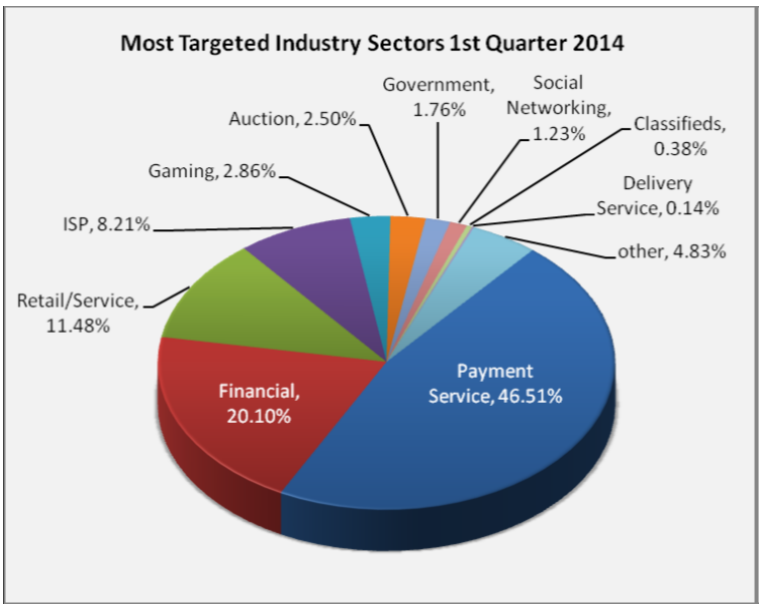

Figure 1.4: Targeted Industry Sectors

Table 1.1: Countries Hosting Phishing Sites 1st Quarter 2014

January		February		March	
United States	56.30%	United States	46.29%	United States	40.21%
United Kingdom	5.17%	France	5.88%	Turkey	4.40%
Hong Kong	3.86%	Turkey	4.11%	Hong Kong	4.13%
Germany	3.52%	Germany	4.04%	Russian Federation	4.00%
France	3.40%	Netherlands	3.39%	Germany	3.87%
Russian Federation	2.62%	United Kingdom	3.37%	Netherlands	3.69%
Netherlands	2.49%	Russian Federation	2.87%	France	3.28%
Canada	2.12%	Canada	2.40%	Japan	2.88%
Turkey	1.95%	Japan	1.96%	United Kingdom	2.86%
Brazil	1.36%	Poland	1.89%	Poland	2.76%

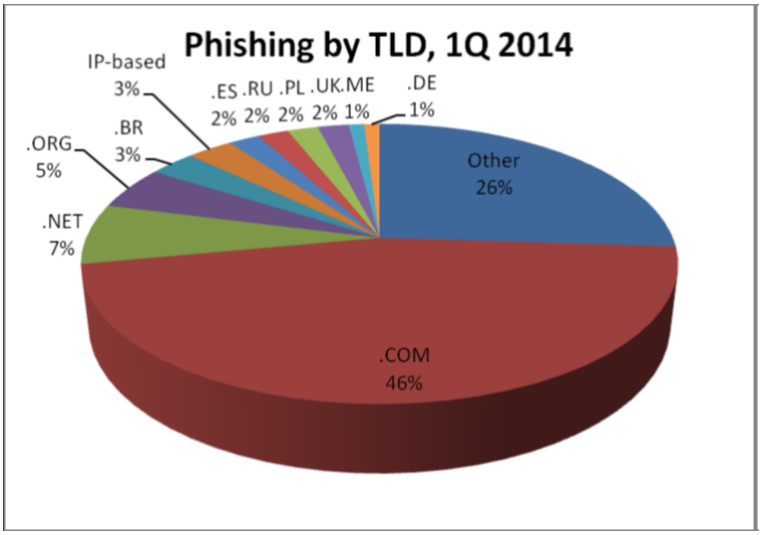

Figure 1.5: Phishing by TLD

Table 1.2: Highlights of Quarter-1: 2014

Statistical Highlights for 1st Quarter 2014

	January	February	March
Number of unique phishing websites detected	42,828	38,175	44,212
Number of unique phishing e-mail reports (campaigns) received by APWG from consumers	53,984	56,883	60,925
Number of brands targeted by phishing campaigns	384	355	362
Country hosting the most phishing websites	USA	USA	USA
Contain some form of target name in URL	56.76%	54.31%	64.47%
Percentage of sites not using port 80	0.85%	0.42%	0.56%

with information retrieval. The first category of methods has very low false-positive rate, but such methods are not robust against future cases. Sheng et al. [39] show that zero-hour protection of major blacklist-based tool-bars has true positive rates in the 15-40 % range. Updating these blacklists typically requires much human effort and is a slow process. For example, January 2012 statistics from Phishtank show that the median time to verify that a URL is a phish was 2 hours. Of course, human-verification can be easily overwhelmed by automatically generated URLs. Heuristics used with machine learning could suffer from the need for a clean, labeled training corpus, over-fitting to the training corpus and the need for retraining because of model drift over time. Password-based schemes lack robustness for phishing detection [16]. Information extraction based schemes such as named-entity extraction [48] suffer from the lack of parsable sentences on web sites and also the limits of automatic natural language processing techniques. Moreover, named-entity extraction techniques are also particularly prone to lower-casing problems such as failure to recognize PayPal (note the lower case) as a named entity.

Proposed schemes in the thesis are built on the following observations. First, the fundamental difference between a phishing and a legitimate site lies in its objective. While a legitimate site typically conveys some information to the user or elicits some basic information from a user to provide a service, a phishing site is designed to *steal* the victim's information. Second, there are basically two kinds of phishing sites in classical phishing attacks. The first type of sites copy the content of a legal site such as a bank, or a credit card company, or a payment site such as PayPal, etc. The second type consists of sites that do not copy any legal site, but instead entice the

user through either an advertisement or a financial lure such as the promise of a prize, gift, etc. These observations play a crucial role in the proposed method.

Although the proposed methods are suitable for classical phishing attacks also, they can also solve the problem of hijack-based attacks. A key feature of *hijack-based phishing attacks* is that the browser address bar continues to show a legitimate URL, e.g., see [40] and the Netcraft news item link given above. Hence, URL analysis cannot catch such phishing attacks. Another disadvantage of URL analysis is that it must deal with URL shortening services and it is relatively easier for phishers to defeat it than the fundamental features and behaviors. Therefore, to thwart hijack-based attacks, a content-based classifier is presented in addition to URL classifiers that combines structural elements of a site together with certain intentional information extracted from the page itself. A key advantage of the content-based classifiers is that they are language independent. They assume no knowledge of phishing signatures or specific implementations. Hence, they could be suitable as both a zero-hour, stand-alone phishing site detection scheme, and also in combination with other existing methods such as blacklists and whitelists.

The presented content-based classifiers analyze the phishing web-site's behavior using a copy-detection algorithm and a real-time bot that injects random input data of the correct type for the form input fields on the web site. Thus, it can also be used to flood the phishing website with junk information and carry out a denial of service attack on confirmed phishing sites.

1.6 Organization of the Thesis

Chapter 2 does a survey and literature study of proposed solutions, approaches and techniques for phishing URL / page detection. Chapter 3 gives the details of the datasets used in the experiments. Chapters 4 and 5 are the crux of the work. They describe the phishers' strategy and hence the proposed detection system in details. They have all the preprocessing and sanitization steps, classifiers required for the various models (discussed in subsequent chapters), and filtering technique. Chapter 6 is a description of a model using real-time bots and fundamental characteristics. Chapter 7 discussed a model using combination of URL and content based classifiers. Chapter 8 proposes a model using just content based classifiers. Security analysis, overall result and comparisons of systems is done in Chapter 9. Chapter 10 concludes with some description about the challenges faces and planned future work.

Chapter 2

Existing Phishing Detection Techniques

Digital identity theft through phishing is primarily a social-engineering attack and has attracted a lot of research interest in this context. Different research groups have studied this problem from various perspectives: server-side and browser-side strategies, education/training, and evaluation of anti-phishing tools, detection schemes and finally studies that analyze the reasons behind the success of phishing attacks [27, 39, 13, 8, 7, 5, 6, 15, 1, 50]. In this section, a survey has been done with prior research directly related to this work on detecting phishing sites and especially those techniques that do not rely on URL analysis since as mentioned earlier, the latest attacks manage to leave the URL unchanged from that of a legal site, which are the focus of this thesis. The following paragraphs outline the prior work on phishing categorized by the research objectives.

2.1 Using Blacklist Approach

The reporting of a page as phishing, then its manual verification is a basic building block of generating blacklists. This could not prove effective alone and gave detection rate of just 15-40% [39]. Updating such blacklists over time is a very time consuming process and requires tedious human effort for manual verification. Also, this strategy will not help in zero-hour phish detection and hence no blacklists is used in any of the algorithm.

Smart use of phishing blacklist is proposed by Prakash et al. [37]. Based on five heuristics, they combine pieces of known phishing URLs from a blacklist to generate new URLs and create predictive blacklisting technique- PhishNet. TLDs, IP address equivalence, directory structure similarity, query string substitution, and brand name equivalence are the heuristics used for creating new URLs. And the approximate matching is done by matching IP address, non web-hosting services, directory structure, and brand names. They claim that PhishNet is efficient and faster than Google Safe Browsing, however the FP rate is 3% to 5%.

2.2 Using Content of Page and Information Retrieval

One approach to detect phishing using web-page content is analyzing the structure of the URLs and validating the authenticity of the content of these target web pages. CANTINA [51] is one such scheme: a content-based approach to detecting phishing

websites, based on TF-IDF (term frequency/ inverse document frequency) information retrieval and text mining algorithms. Calculating TF-IDF of the given page, taking 5 terms with highest TF-IDF, searching these terms in Google and matching the domain in specified search results is their architecture. Simple heuristics like domain age, known images, suspicious URL, suspicious links, IP address, dots in URL and form with <input> tag are also combined with TF-IDF to reduce false positives. They tested their methods on a small dataset (100 phishing and 100 legitimate) because analyzing content takes time and showed that the pure TF-IDF approach can catch about 97% phishing sites with about 6% false positives and 90 % detection rate with 1 % false positive rate if combined with some heuristics. Their results exhibit a trade-off between detection of phishing web sites and the false positive rate for legitimate web sites. The methods do not produce good results with East Asian languages and are mostly restricted to English. Search querying adds delays in the processing. CANTINA+: A Feature-rich Machine Learning Framework [47] is an advanced version of CANTINA. This is explained in details in machine learning section below.

There are many other schemes that use some subset of URL features, IP-based features, and content-based features [16, 28]. Garera et al [16] showed an average TP of 95.8 % and FP of 1.2% with a smaller dataset of 2508 URLs. Capturing phishing patterns, fine-tuned features and few other features from four set of phishing URLs, they apply a logistic regression model and to detect phish. Although the results are exciting, the system produces unstable results.

2.3 Zero-hour Phishing Detection Using Text Analysis

Xiang et al. [49] proposed a scheme for zero-hour phishing site detection, which uses whitelists, text comparison of the web-page against the text content of existing phish sites, and additional verification using a search engine (as in CANTINA) if a page is flagged as a potential phish. They also use a sliding window in the back-end to incrementally build a *machine learning* model as new phishing signatures are built. They achieve a 0% false positive rate with a true positive rate of 67.74% using the search-oriented filtering and a 0.03% false positive rate and 73.53% true positive rate without it.

2.4 Using Machine Learning

In CANTINA+ [47], the CANTINA and zero-hour phishing detection researchers enriched their techniques with machine learning and did bigger experiments with unique and near-duplicate websites and were able to achieve a detection rate of 92% for phishing web sites with false positives ranging from 0.4% to 1.4% depending on the testing scenario: randomized (10% of phishing websites as training data - the sites could be future or historical) versus timed testing (20% of phishing websites as training dataset - the sites came from only historical data). The architecture of CANTINA+ can be divided into 3 major modules. First one uses hashing to filter highly similar phishing pages, second one checks for the utilization of login forms, and

third module 15 features with machine learning algorithms for classification. 8 out of the 15 features are claimed to be novel. Embedded domain (seeking dot-separated string segments), Out-of-position TLD, bad forms, bad action fields, non-matching URLs, out of position brand names, and searching for copyright names etc. are some of the novel features. Although having such a complex structure, the system will fail for cross side scripting attacks, hijack based attacks, and for pages with just images (no text).

A research team from Google has presented a machine learning classification technique with hundreds of features to accomplish a large scale automatic classification of phishing web pages [45] by analyzing both the URL and the content of the page and claims to achieve 90% accuracy in classifying web pages with false positives below 0.1%. This classifier is being used to maintain Google's phishing blacklist automatically.

The Google classifier is rebuilt *every day* and if a website is off-line during the training phase it will not be in the model. This work directly compares results by testing same URLs using Google's detection and the thesis. Although they have better false positives rate but their detection rate is significantly lower than the classifier (detailed comparisons are in the Results section). Even if the phishing contents are taken down from the website, their classifier marks the URL as phishing. Hence, their analysis is not real time.

2.5 Other Detection Techniques

There are many other strategies used in phishing detection. Some of them are based on URL analysis, information extraction, text analysis, etc. For more details on phishing and detection schemes, readers are encouraged to refer the books by [24, 25] and [35] and the paper [22] for additional references. Although there is considerable research on phishing detection but there are very few schemes for zero-hour phishing detection or for hijack-based attacks and the existing schemes are not very effective in these scenarios. Hence, this thesis proposes new schemes for hijack-based attacks in particular that could also be used for zero-hour phishing and evaluate them rigorously.

2.5.1 Password Based Techniques

Monitoring information flow is one of the ways to safeguard users from getting phished. AntiPhish [24] is one of the system that will keep track of domains and password combination. The system looks at the password and tries to match it with the earlier visited login domains. If it sees the familiar password being entered for another domain, it will warn the user. There are some shortcomings with this approach. The FP rate will be more if the user has same password across different domains. The tool requires some manual intervention. No formal experiments have been done to compare the system with others.

PwdHash [38] studied password hashing strategy. PwdHash is a browser extension created to improve the password authentication They send hash values rather

than plain text to the server. However, most of the password based techniques (including PwdHash) will suffer from DNS attacks, focus stealing, spywares etc.

2.5.2 Toolbars

There are many toolbars/ plugins/ extensions and small softwares available to guard users from phishing attacks. All recent browsers come with black-list verification. McAfee site advisor, Norton, etc are some of the products. WOT (Web of Trust), NetCraft are famous browser plugins. Although they hide the implementation and technical details, they have been used and appreciated widely for regular use. Sheng et al [39] measures the efficiency of such tools. They conducted two tests on 191 fresh phish and 15,345 legitimate URLs to study the effectiveness of blacklists. They found that the blacklists for the extensions are updated at varied speeds and the detection rate is less than 20% at zero hour and around 47%-83% in 12 hours with 0% false positives.

2.5.3 Using DNS records

Another interesting technique in detection is to monitor DNS records on particular time intervals. Fast flux is a DNS trick in constantly changing network of compromised hosts which will act as proxies. Hence, botnets widely use this trick to hide phishing sites. McGrath et al. [30] proposed that it is possible to identify and protect from it. They used IP address, geo-location, border gateway protocol prefix and anutonomous system number (ASN). This information is made into features

like 'Number of IP addresses', 'Number of associated ASNs ', 'Number of associated prefixes', 'Number of associated countries', 'Number of DNS servers corresponding to Web servers', 'Short time to live.' Using SVM, the model classifies URL as either 'flux' or 'non flux'. Although they claim that such system achieves zero FP rates, but retraining SVMs, detection time and implementation are some of the issues.

2.5.4 Image Processing

Using similar images as of target web page to trick the user is very old and successful strategy by phishers. Most of the text based systems will fail if only images are used to give the similar looks to the phishing pages. Liu et al. [44], Dhamija et al. [10], Chou et al. [9] etc. showed interesting results on small corpus of URLs. Guang-Gang Geng et al. [19] introduced novel features that work on the favicon and presented impressive results. But, all such techniques are more expensive and slower as they require much processing. And hence, implementation of such systems is an issue.

2.5.5 Phishing Email Detection

As [23] observed, detecting phishing email messages automatically is a non-trivial task since phishing emails are designed cleverly to look legitimate. Besides attachments, an email can be decomposed into three main components: a header, a text body, and links. While the header and links have been well studied by phishing detection methods previously, unsupervised natural language processing (NLP) techniques for text analysis of phishing emails have been tried by only a few researchers.

In [50], rudimentary analysis of the anchor text was used to enhance detection. In [41], hand-crafted patterns and some scoring functions for verbs in the email were designed using trial and error. In [42], a semantic t-test was introduced for feature selection that improved the text analysis significantly.

As a final note, even after referring to almost all famous detection techniques, none of the system dealing with hijack based attack [40] can be found. In this thesis, in addition to classical and fundamental features, two novel classifiers are introduced that can tackle this issue.

Chapter 3

Datasets

The final experiments on each model/ combination is conducted on various diverse datasets. The main motif is to make the decision on real-time behavior of the page. Hence, gathering freshly reported phishing URLs is a must. PhishTank is a collaborative clearing house for data and information about phishing on the Internet. Also, PhishTank provides an open API for developers and researchers to integrate anti-phishing data into their applications at no charge [36]. XML, CSV, serialized PHP and json are different formats provided for phishing URLs on Phishtank.

The json structure is as follows:

array(

array(

“‘phish_detail_url’ = ‘ `http://www.phishtank.com/phish_detail.php?phish_id=123456` ’,”

```
        'submission_time' = '2009-06-19T15:15:47+00:00',
        'verified' = 'yes',
        'verification_time' = '2009-06-19T15:37:31+00:00',
        'online' = 'yes',
        'target' = '1st National Example Bank',
       'details' = array(
         array(
          'ip_address' = '1.2.3.4',
          'cidr_block' = '1.2.3.0/24 ',
         'announcing_network' = '1234',
          'rir' = 'arin',
         'detail_time' = '2006-10-01T02:30:54+00:00'
          )
        )
    )
)
```

The main attributes are defined as follows:

- **phish_id:** The ID number by which Phishtank refers to a phish submission.

All data in PhishTank is tied to this ID. This will always be a positive integer.

- **phish_detail_url:** PhishTank detail url for the phish, where one can view data about the phish, including a screenshot and the community votes.
- **url:** The phish URL. This is always a string, and in the XML feeds may be a CDATA (character data) block.
- **submission_time:** The date and time at which this phish was reported to Phishtank. This is an ISO 8601 formatted date.
- **verified:** Whether or not this phish has been verified by phishtank community. In the data files, this will always be the string 'yes' since they only supply verified phishes in the files.
- **verification_time:** The date and time at which the phish was verified as valid by phishtank community. This is an ISO 8601 formatted date.
- **online:** Whether or not the phish is online and operational. In the data files, this will always be the string 'yes' since they only supply online phishes in the files.
- **target:** The name of the company or brand the phish is impersonating, if it's known.

Responsible web users report phishing URLs in phishtank. The by crowd voting by the registered user, a phishing URL is verified. Figure 3.1 and 3.2 represent the daily submission and verification count of the phishing URLs in phishtank.

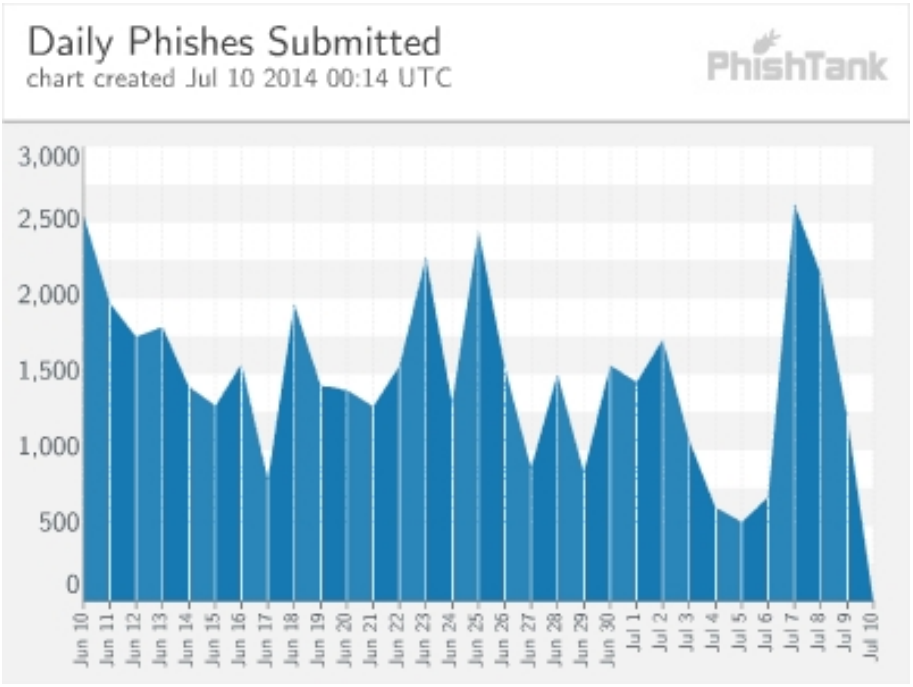

Figure 3.1: Daily Phishes Submitted

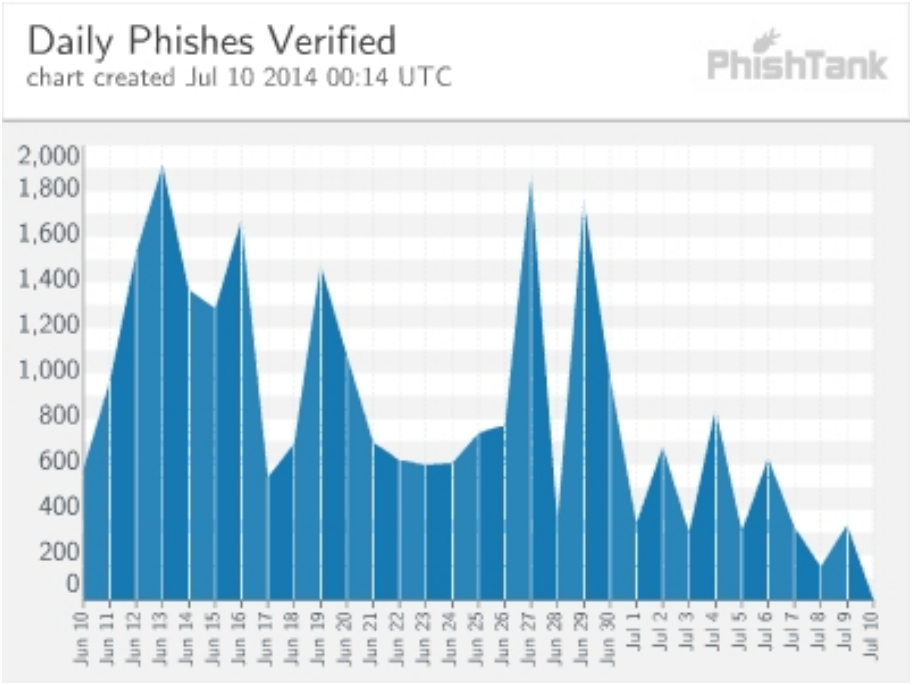

Figure 3.2: Daily Phishes Verified

Verified and online phishing URLs are taken from Phishtank just before the experiments so that the websites would still have phishing content. The availability of the content on the URL is confirmed by opening it in sandbox and asking for the response. If response is received (withing 10 seconds of the request), then it can be generally said that the page is still alive and can be added to the phishing dataset. 10,000 such fresh online phishing URLs are parsed, randomized and 4000 URLs are added in phishing dataset for testing. The availability of the legitimate URLs can be trusted and hence 4000 random URLs from legitimate URL set are directly added. Legitimate URLs are from top 10,000 Alexa [1] domains and random URLs from DMOZ [34]. Some of the classifiers need training and hence different sets are created confirming that there is not a single overlapping URL between any URL from training set and testing set. The detail of the datasets used for particular model is explained in the respective chapters.

[1] as of mid November of 2013

Chapter 4

Preprocessing and Sanitization

4.1 Preprocessing

The very first and simple pre-processing step is to remove duplicate URLs and analyze only unique URLs. This step checks if the original URL or the URL resulting from opening the page (redirected URL) is already processed. This removes the redundancy in the input data and ensures unbiased results.

To make the system completely real-time and zero hour, a model tries to open the URL in sandbox and gather the content. The source code of the web-page, title, the redirected URL and the domain of the input and redirected URL is collected. There are many cases where the URL was malformed or no longer available. Such URLs give *time-out* errors. Some of the URLs give 404 in HTTP standard response code. Some URLs show a *Page Not Found* or *Suspended Page* error page without giving

any exception. It is also necessary to check for the pages that returned no source code. Almost all HTTP Errors (including *HTTP Error 400: Bad Request,HTTP Error 500: Internal Server Error*) are checked. All such invalid URLs are reported and removed, and remaining are passed for further processing.

4.2 Highly Reputed and Genuine Domains

This step checks whether the domain of the input URL or redirected URL is in the whitelist, and then bypasses it as a legitimate URL. This function avoids further complex processing steps for reputed legitimate URLs and hence the classification of the web pages can be done faster.

The administrators of reputed domains follow professional security practices and ensure the sanctity of the content hosted on the domain. They typically will not allow anybody to put malicious content that will cause their domain reputation to go down. A set of such genuine domains is gathered for the whitelist.

Over time, the whitelist of some highly visited and reputed domains is built using top 5,000 domains as ranked by Alexa and some of the targets domains that were not present in Alexa top-5000. Many phishers use free hosting domains to launch their phishing attacks. There can be instances in which an administrator takes some time to remove such pages from the hosting domain. To avoid this scenario, domains that offer free hosting and free blogging are removed from the whitelist. It is also noticed that the list contained shady domains like illegal downloads, porn and pirated entertainment. It might contain adware and end up in phishing sites. Also,

URL shortening services are used to disguise the URLs. Hence, all such possible domains from whitelist are removed from the list by manual inspection. A separate program is run to see the distribution of common targets from the json file provided by Phishtank. It is discovered that some of the common banking targets and on-line shopping links were missing from the top rated domains. They are added to the whitelist. Finally, the whitelist of just 5005 domains is ready for use.

Google is a top rated domain and it has google document service. But, phishers also managed to create phishing pages and host it on the Google document site [20]. The URL for phishing page can be composed as `https://docs.google.com/my-demo-PayPal-phishing-page`. So, all the Google domains except for docs.google.[Top Level Domains (TLDs)][1] are compared with whitelist and necessary action is taken. This is one time offline training for whitelists.

4.3 Sensitive Information Check

The main motive of phishers is to steal confidential information from the victims. This step ensures that the page is really asking for some sensitive information and can be a potential phishing candidate. The HTML source code is inspected (checked for <input> tags) and it has been checked whether the page has a password field (*type =“password”*) anywhere on it. Many phishing pages go off-line soon, and the pre-processing step takes care of the errors and exceptions caused by such behavior. But, many domains show a completely different page with fancy error messages

[1] [TLDs]= set of all TLDs like com, co.in, jp etc.

Table 4.1: Confusion Matrix for Classification

	Classified as Phishing	Classified as Legitimate
Phishing page	True Positive (TP)	False Negative (FN)
Legitimate page	False Positive (FP)	True Negative (TN)

(e.g., `http://commaccounting.co.nz/moodle/blog/a/e84c` showing 'Whoah! You broke something!'). This URL hosted phishing content in past but while processing it, there were no such contents and no errors but a page with this fancy message. This page can no longer be classified as phishing and *Sensitive Information Check* will mark such cases as legitimate. It has been observed that some phishing URLs might have been hosting phishing contents in the past but they did not show any malicious behavior or phishing stuff. As the thesis focuses on real-time behavior of the pages, the model doesn't mark such URLs as phishing. Only pages asking for sensitive information advance after this step.

4.4 Metrics

As the thesis work is interested in detecting phishing pages, this the "positive condition". The detailed confusion matrix is shown in Table 4.1. The results are summarized using the following measures.

True Positive Rate (TPR) = True Positives/Actual Positive= $TP/(TP + FN)$

False Positive Rate (FPR) = False Positive/Actual Negative = $FP/(FP + TN)$

Precision (PR) = $TP/(TP + FP)$

F_1 score (F-score) = $2 * PR * TPR/(PR + TPR)$

Chapter 5

Classifiers

The overall system architecture of each model consists of five major steps. The first one is *Preprocessing and Sanitization* of the input URL. The second step to check for *Highly Reputed and Genuine Domain* in the input URL and whitelist it. The third step *Sensitive Information Check* confirms that the page is asking for some confidential data and can be potentially phish. Fourth step comprises of rigorous testing with various classifiers (described further in the section). Fifth step is the final decision of the model. This step records individual output from various combinations of classifiers and based on some rules, it gives the final verdict. As the rules are different for each model, this step is discussed in corresponding chapters. Sixth step *Search Based Filtering* is optional and takes the advantage of Internet search to attain lower false positive rate. All the classifiers are described in details in this chapter.

5.1 Classifiers

Distinguishing features from legitimate and phishing web-pages provides us with many heuristics. Some of them are classical and well studied for phishing detection. Two completely novel heuristics depending on behavioral and textual analysis are proposed and proved efficient along with other classifiers in the thesis. Each of them leads to some set of rules and thus separate methods are designed for such rules. Each method marks particular URL (hence the web page) as phishing or legitimate. Hence the methods are called as classifiers and the classifiers can broadly categorized as **'URL based'**, web page **'Content Based'** analysis, and **'Machine Learning Based'** classifiers.

5.2 URL Based Classifiers

5.2.1 Targets in URL

Cyber criminals put the target (e.g., PayPal, eBay, BankOfAmerica, etc.) in the URL to disguise the URL to the user (Figure 5.1). According to quarterly reports by APWG (Anti Phishing Working Group), 45-50% phishing URLs contained target names in the URL last year (2013). This strategy is used to fool naive users into believing that they are really visiting the desired page. Many researchers discussed this strategy and use this feature. This method is improved over existing techniques. The URL is taken, and the main domain and TLD is removed from it. It gives either sub-domain and/or the extended URL. Then it has been checked for targets

in such remaining part of the URL and not the complete URL directly. Obviously an accurate and comprehensive list of targets is needed for this purpose. More than 12,000 URLs from Phishtank are analyzed and the top targets (that were targeted more than 0.1% of the time among those URLs) are collected. As expected, PayPal ranked first with more than 13.6%, followed by AOL, which was just 1.1%. Also, phishers try to attack a big population and hence a popular domain. So, all the whitelisted domains are added to the list of targets. This made the list strong and removing the actual domain name with TLD from the input URL helped us to get rid of false positives. Small experiments were done to test this function and the result varied from true positive rate of 40% to 50% with about 0% false positives.

5.2.2 Misplaced TLD

The idea is similar to checking targets in URL, but instead of targets, the interest is in misplaced TLDs (Figure 5.1). A list of all top level domains (TLDs) that are not placed at the actual TLD location are noted. Phishers use this strategy to disguise URLs. The main domain and actual TLD from the URL is removed and checked for the TLDs in the rest of the URL.

5.2.3 Presence of IP address

Only cyber criminals use IP address (Figure 5.2) to advertise their websites. The strategy behind it is that the user is left clueless to which website he/ she is visiting. All genuine websites use meaningful and descriptive words in their URLs. Such

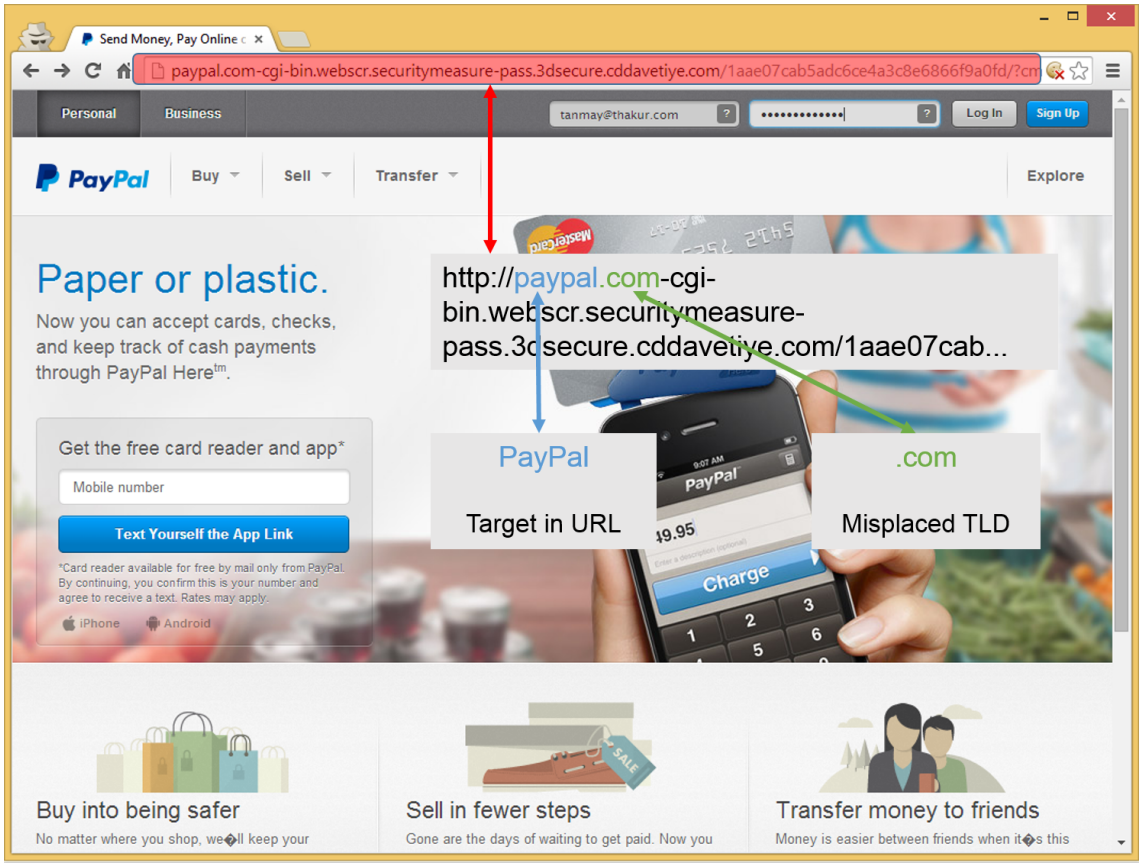

Figure 5.1: Phishing URL Tricks

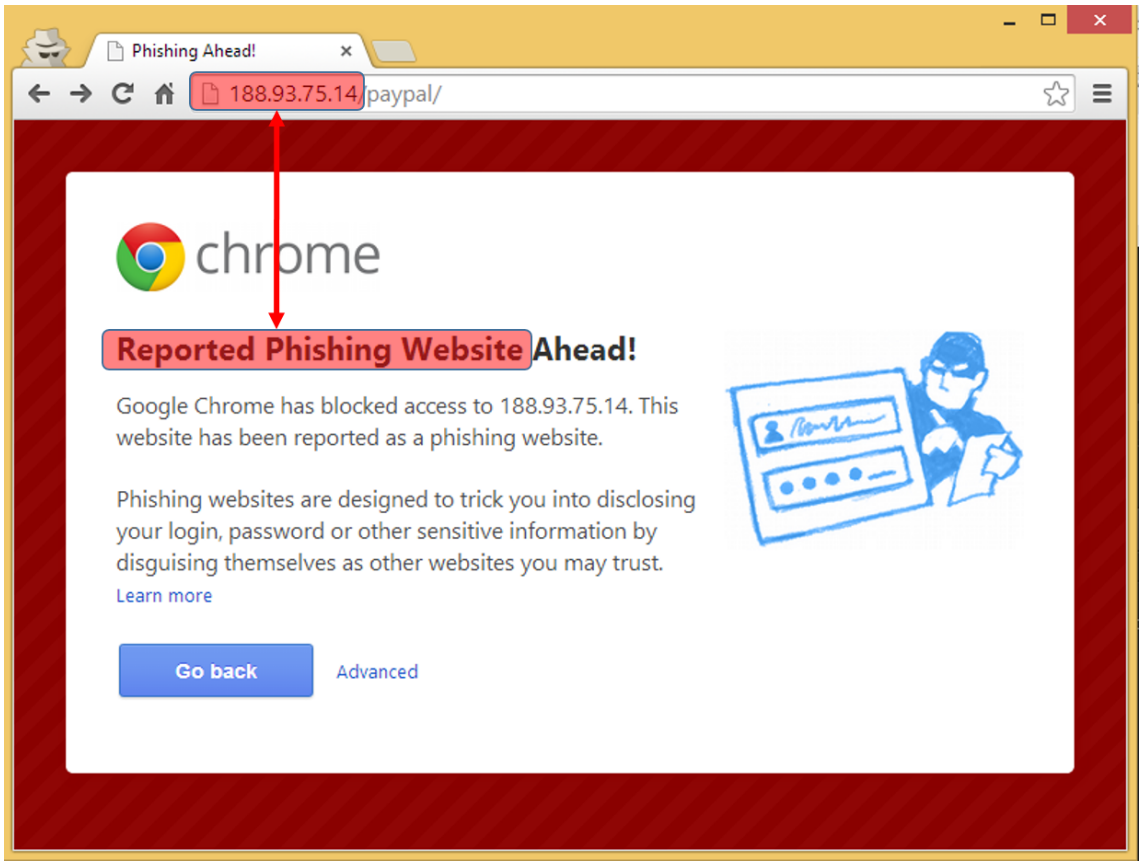

Figure 5.2: Presence of IP address

websites can be represented as IP address but no legitimate website owners use it. Hence, use of IP address seems suspicious.

5.3 Content Based Classifiers

5.3.1 More Redirections

To maximize the profit by phishing attacks, phishers use free hosting sites to launch their attacks. Such free hosting sites want to advertise their domains and so they keep many links redirecting to the main site. They also give limited data space and phishers cannot use big-sized data such as images. So, phishers directly use the image links from the actual target site to display it on the phishing page. All of this leads to more external URLs than internal (also called as *redirections*). Both internal, external URLs are counted and if external URL count is more than internal then it can be said that the page may be a potential phish.

5.3.2 Copy Detection

This aims on calculation of closeness and similarity of the given page with the target pages. This method is developed working closely with Earl Lee et al. [26] and using the fact that phishers try to make the phishing page look almost similar to the target page to disguise. Precision, recall and F-score of the closeness of the given page and potential target page is calculated. A small experiment with about 100 phishing pages and 100 legitimate pages was conducted to determine the threshold to use for the F-score. If the F-score for the similarity of a candidate page with a legitimate page exceeds the empirical threshold and the URL of the page is not in the whitelist, then it is marked as phishing.

The second part in this function is to check the copyright name of the page. Signatures of 513 login pages of common targets are recorded. Big organizations have their own website team and they reserve their own copyrights. Of course, the copyright name partially matches the part of the domain of the URL. Again, this seems a novel idea. CANTINA+ [47] has used copyrights for Internet search and not for domain comparison. For example, PayPal phishing site has copyright as ©1999-2014 PayPal, Facebook has ©Facebook 2014. Phishing pages copy the targets as much as possible and hence, they show the same copyright message. This definitely fails to match with the domain of the URL and hence such cases are marked as Phishing. If no similar candidate of the given page is found in the whitelist, then it is marked as undetermined. This is justified since the whitelist is not necessarily exhaustive. Earl Lee has a great contribution in the designing of the classifier.

5.3.3 Unsecured Password Handling

It is very important for any page to transmit the passwords in encrypted format. SSL is an acronym for an encryption technology called Secure Sockets Layer. SSL certificates are given to the domains or sites to ensure this secured and encrypted flow of passwords. The best case is when the page having SSL is asking for password fields and submitting it to the page with SSL certificate. Some organizations ask for the password fields without SSL but ensure to transmit the password via a secured SSL channel and to the page having SSL. Use of SSL avoids eavesdropping, message forgery or message tampering to a greater extent. As the cost of SSL is high and the registration process involves owner's contact details, phishers usually do not use SSL.

This method checks for the secured flow of the passwords. If there are no password fields then we can directly say that the page is not phishing. And if the page had unsecured flow then it is a '*vulnerable page*' and marked as potential phishing page.

5.3.4 Behavior Analysis

Manual inspection of visiting a URL in an Internet browser and predicting it as phishing or not is the most effective way if the visitor knows very well the different phishing techniques and patterns. The *Real-time Form Analysis* and *Real-time Login Bot* classifiers provide run-time behavior analysis based on an inspection of a handful of legitimate and phishing sites.

Filling all the required fields in the form with all invalid credentials, we expect that the page should give error message and ask us to re-enter the fields. This is the legitimate behavior. But, phishing pages will accept any data filled in the fields and either redirect us to actual target page or give us a message similar to *Thanks for logging in. All servers are down, please connect after some time.* After entering the fake confidential data on the forms, following behavior is noticed.

Legitimate Behavior:

- Keeps on the same domain and/or redirects to SSL certified same domain.
- Gives error message.
- Gives another chance to login or asks to reset password giving *Forgot Password?* link.

Phishing Behavior:

- Accepts the fake credentials and shows *Thank you* page.
- Redirects to the target domain (or any other domain than the input URL).
- Asks for more information.

This is very basic behavior and due to diversity, all the (rare) cases of more complex behaviors can not be listed. Advantage of such distinguishing behavior of login forms is studied and corresponding classifier components are created. *Real-time Form Analysis* mainly focuses on the action-URL of the forms and *Real-time Login Bot* classifier is more intelligent and complex. It mimics the actions of the victim, checks the behavior and takes the decision.

5.3.4.1 Real-time Form Analysis

Logins forms have action-URL. A bot is created to mimic a web browser and parse all the forms on the given page. The forms which don't ask for sensitive information can be bypassed and the remaining forms are processed. Action-URL is the biggest hint for observing the form behavior.

This is simulation of human interaction with the phishing attempt to check for the behavior. According to the literature study in phishing domain, this is a completely novel method. Algorithm 1 describes the detailed rules.

Sign-up forms have exceptional behavior. As they enable us to create new profile, they accept any syntactically correct email-ID and/or user-name and password. And

Algorithm 1 Behavior Analysis

```
1: procedure Real-time Form Analysis(URLs)
2:     Extract all the forms requesting input through password fields
3:     if not a single form for the input-URL then
4:         return Legitimate
5:     end if
6:     Parse the action-URLs
7:     if domain of the action-URL = the input domain then
8:         if the web-page of action-URL is SSL certified then
9:             return Legitimate
10:        end if
11:    end if
12:    if domain of the new URL ≠ domain of the input URL then
13:        return Phishing
14:    else if action-URL = given-URL then
15:        legitimateFlag := True
16:    end if
17:    open the page by the action-URL inside sandbox
18:    if the new page did not ask to login again then
19:        phishingFlag := True
20:    else if new page keeps on same domain & asks for some more
         sensitive information & none of the conditions above satisfies then
21:        create a set URLsToProcessSet of all such action-URLs from the page
22:    end if
23:    while URL in URLsToProcessSet do
24:        iterationCounter ← iterationCounter + 1
25:        Follow the similar steps from (step 2) for each URL
26:        Remove the processed URL from URLsToProcessSet
27:    end while
28:    if URLsToProcessSet is empty || iterationCounter > threshold then
29:        if legitimateFlag := True then
30:            return Legitimate
31:        else if phishingFlag = True then
32:            return Phishing
33:        else
34:            return Undetermined
35:        end if
36:    end if
37:    return Undetermined
38: end procedure
```

then they say profile created successfully and may give a form to login. From the bot perspective this is the combination of both phishing and legitimate behavior. But, step-15 in the algorithm 1 takes care of such behavior.

5.3.4.2 Real-time Login Bot

This classifier is similar to Real-time Form Analysis but this is more advanced and intelligent. URL obfuscation techniques, use of deceptive methods using javascripts etc. are some of the tricks from the expert phishers. Just looking at action URLs might not be the correct approach in such situations. The outputs from both the behavioral classifiers are trustworthy and both have their own advantages.

A sandbox is created and an artificially intelligent bot is started in it. The role of the bot is to open up the page for given URL, parse the content and follow the set of rules as explained in the algorithm. The basic idea of rules is to act as a victim, submit the complete (syntactically valid; ex: properly formatted fake email address in email fields) information asked and observe the changes in the page state.

This is also a simulation of human interaction with the phishing attempt to check for the behavior. And this is again completely novel method according to the phishing literature review. The bot is currently less complex and still adapting. The current bot handles the most common HTML attributes like text (used for user-name, card number etc.), email, passwords (used for asking password, CCV number[1] etc.), button (submitting the filled information). This covers almost all the fields

[1]Credit/ Debit card security number

Algorithm 2 Behavior Analysis

```
1: procedure Real-time Login Bot(URLs)
2:     Parse all the forms in the HTML
3:     Extract all the forms requesting input through password fields
4:     Fill the suitable information in the appropriate input fields
5:     Submit the form
6:     Check the behavior after submission
7:     if domain of the new URL ≠ domain of the input URL then
8:         return Phishing
9:     end if
10:    if the new page did not ask to login again then
11:        phishingFlag := True
12:    end if
13:    if new page keeps on same domain then
14:        legitimateFlag := True
15:    end if
16:    while the page requests more information do
17:        Follow the similar steps from (step 2)
18:        if URL of the new page matches the old one for threshold times then
19:            return Legitimate
20:        end if
21:    end while
22:    if legitimateFlag = True then
23:        return Legitimate
24:    else if phishingFlag = True then
25:        return Phishing
26:    else
27:        return Undetermined
28:    end if
29: end procedure
```

used in current phishing attacks, but to make the bot expert in phishing detection, the rules to process check-boxes, radio buttons, drop-down lists etc. can also be designed. The exceptional behavior of sign-up forms is handled at step-13 of the algorithm 2

The bot will not add junk information to the database of phishers and there will be no harm on genuine websites, as this will be treated just as an incorrect login.

5.4 Machine Learning Based Classifiers

5.4.1 Machine Learning: Advantages and Disadvantages

Machine learning methods are well known for its advantages like intelligence, processing of large data, decision making, more accuracy than human crafted rules [2] etc. Taking advantage of this, two classifiers (Redirection Analysis and Copy Detection) are designed. Apart from this, cost of training, reliability of the model over time, its adaptability to changing trends in phishing techniques, periodic need of retraining, and requirement of big noiseless data are some of the challenges in machine learning. Also, it may not perform as good of human-crafted rules for *Zero Hour* environment. Therefore, this classifiers should be used in combination with the other non-machine-learning classifiers for better efficiency in zero hour scenario. These components enhance sustainability of the model over time.

5.4.2 Machine Learning Algorithm

Researchers have done plenty of work on evaluating different machine learning based methods [1, 31]. PART [14] was chosen for classifiers because it performs reliably, faster and allows us to view the rules that the algorithm generates. Being able to view the rules gives us insight into the classifier, and allows us, for example, to identify reasons for false-positive results. The PART algorithm is a separate-and-conquer rule learner. It creates a partial C4.5 decision tree and chooses the best leaf as a rule for the classifier. The algorithm combines C4.5 and RIPPER rule learning algorithms.

5.4.3 Dataset, Training and Testing

Variety in training datasets is important to gain robustness in the model. Therefore, the *'Copy Detection'* and *'Redirection Analysis'* components are trained on different datasets. The required contents related to the URL from 1000 randomly selected URLs each from *Phishtank* dataset from March-2014, list of top 10,000 *Alexa* domain and *DMOZ* [34] set. Excluding the faulty URLs, and for matching the dataset numbers, 952 URLs are used from each of the dataset. Two sets are created: the first called TrainingSet-1 is a combination of *Phishtank* URLs with *Alexa* URLs and the second set called TrainingSet-2 combines *Phishtank* URLs with *DMOZ* URLs. It is confirmed that not a single URL is common between the training sets and testing sets. The feature-sets used for training are described in the subsection below.

5.4.4 Redirection Analysis

Because of the reasons discussed in 5.3.1, phishing sites show some pattern between number of external URLs, internal URLs and their ratio. Feature-set from fundamental characteristics such as number of internal redirections, number of external redirections, and the ratio of internal redirections to the total redirections is created. Special cases like zero total redirection is also handled. Training the feature-set with PART gives 4 simple decision rules, which is implemented in the model. Although, the Phishtank dataset is very noisy and contained many legit URLs, 10 fold cross-validation showed True Positive Rate (TPR) of 93.5% on TrainingSet-1 and 86.8% on TrainingSet-2 with respective False Positive Rate (FPR) of 14.4% and 19.8%, Precision (PR) of 86.7% and 82.7%, F-scores of 90% and 84.7%.

5.4.5 Copy Detection

This classifier is developed using the facts described in 5.3.2 that many phishers try to make the phishing page look almost similar to the target page to fool the visitor. Precision (computed as intersection of words divided by size of candidate page), recall (computed as intersection of words divided by size of target page) and F-score of the closeness of a given page is noted with respect to potential target pages.

All the words are extracted from the visible text of the HTML page and also IDs of the HTML tags and elements. From the extracted list of words and IDs, the classifier removes numbers, single or two letter words and special characters (ex: ASCII values). Four lists of words are created: *List-1* has all the words, *List-2*

contains top-20 most recurring words, *List-3* holds randomly selected words from *List-2*, and *List-4* contains all the IDs grabbed from HTML source code. Two similar pages will have higher F-score value than two distinct pages. Database is also created of these lists for the whitelist. Once a test URL is passed, the objective of the function is to calculate F-scores for a test URL for each of the four lists with respect to each URL in whitelist and then aggregate the F-scores for each list using the maximum function, and if the maximum crosses a certain threshold, then consider the URL as copying the whitelisted page. If the test URL is same as in whitelist, then the F-scores for all the lists will be the highest and hence, it will be mis-classified. But, such a URL will not come to this component, as it will be already bypassed as *Legitimate URL* by the *Preprocessing and Sanitization* steps.

The thresholds are learned using machine learning. Above features gave 12 different rules for PART model and it is cross-validated. Besides the noisy *Phishtank* dataset, 10 fold cross-validation gave TPR of 80% and 88.6%, FPR of 11.9% and 15.2%, PR of 87% and 88.2% and F-Score of 83.4% and 88.4% on TrainingSet-1 and TrainingSet-2 respectively.

If the classifier could not find any similar candidate of the given page in the dataset then it says the URL as undetermined. This is justified since the whitelist (and hence the dataset containing above mentioned lists) is not necessarily exhaustive. Keith Dyer has helped a lot in designing this classifier.

5.4.6 Combined URL classifier

The methods discussed in 'URL Based Classifier' section 5.2 are improvements and additions on prior research. The basic idea behind this classifier is to create more advanced URL classifier using various features and using PART machine learning algorithm. A multi-feature classifier that uses nine features to determine legitimate or phishing is designed working closely with Keith et al. [11]. The features used were the length of the domain, the number of @ symbols and hyphens in the URL, the number of punctuation symbols in the URL, the number of top-level domains present in the domain of the URL, the number of target words present, the number of suspicious words present, whether or not the URL is an IP address, and finally the Euclidean and Kolmogorov-Smirnov distances between the distribution of characters in the URL and the distribution of characters in standard English text. For the number of punctuation symbols, the count is incremented by 1 for every punctuation in the URL that was also in a small list of 13 common punctuations. In the number of top-level domains feature, it has been looked at the domain of the URL and counted every occurrence of a common TLD, such as *.com* and *.net*. The number of target words found in the URL is the number of times a word from the target list appeared in a URL. Some target word examples are PayPal, BattleNet and WellsFargo. The IP detection feature uses a simple method to determine if the URL is an IP address. The number of suspicious words is similar to the number of target words, with the difference being that while target words are typically businesses, suspicious words are action words such as confirm, login, and signin. The Euclidean distance is the sum of the squares of the differences between each character's normalized frequency in the

URL and in Standard English text. The Kolmogorov-Smirnov distance is calculated by constructing an empirical distribution function for the distribution of character (normalized) frequencies and the frequencies of the same characters in Standard English text. The two-sample Kolmogorov-Smirnov test statistic is calculated for the two empirical distributions.

The dataset for training and testing this classifier is different than those explained in section 5.4.3 and the details are mentioned below in the next paragraph.

To train the classifier, data set I is created that consisted of 10,600 legitimate and 9,640 phishing URLs. The legitimate URL list is taken from the top 12,000 websites provided by Alexa.com accessed on February 11, 2014. The phishing URLs are taken from 12,000 results from Phishtank.com accessed on February 12, 2014. The 12,000 legitimate URLs are taken removing all common occurrences with the testing data sets bringing the count to 10,600 and thereby ensuring that the training and testing sets are completely unique. The same process is used on the phishing set bringing the count to 9,640. Combining these two sets, the final training data set is built for the URL classifier. Additionally, data set II from Jianyi Zhang of Beijing Electronic Science and Technology Institute is provided. This data set consisted of 18,397 phishing URLs acquired from APWG combined with 20,000 less popular legitimate websites gathered by Zhang's crawler. 10-fold cross validation technique is used to evaluate the classifier.

Table 5.1: Confusion Matrix for ML based URL Classifier

	Data Set I		Data Set II	
	Legitimate	Phishing	Legitimate	Phishing
Classified as Legitimate	10384	515	16758	2091
Classified as Phishing	216	9125	1638	17909

5.5 Search Based Filtering

Search engines have special features which enable them to display highly ranked pages higher in the search results. A login page from a legitimate site would have been accessed by many people and hence, it will be ranked higher. In contrast, as the lifetime of the phishing websites is very less, there is low chance that they can make high page-rank. The average lifetime of a phishing domain is 3 days, 31 minutes and 8 seconds (*about 1/3rd of the phishing domains last 55 minutes*) [30]. Hence, we can take advantage of page-rank system by search engines.

If any model marks an URL potentially phish, then this filtering process is applied and top ten results from a regular Internet search is grabbed. However, this is an optional step in the algorithm (as opposed to previous work [51, 47] in which it is essential) and the results with and without Internet search are reported individually. Either Yahoo or Bing are used as search engines depending on the models and explained in the corresponding section.

Unlike other systems [51, 47], the search based filtering is not complex. Without any complicated processing like TF-IDF, most frequently appearing words in the page, and natural language processing, just the whole redirected URL is queried to search engine with its default settings. Top 10 results are taken from it and checked

if the domain of the URLs in the result matches with the domain of the queried URL. It is a clear sign of highly ranked domain if at least 2 domains are matched. Then the URL is considered as legitimate (independent of the results from the classifiers). If the domain matches are less than two, then the model continues with the decision given by the main algorithm. The filtering is optional and is focused on reducing the false positives.

Chapter 6

Model Using Real-time Bots And Fundamental Characteristics

In this chapter, an implementation of bots to attack directly on phishers' modus operandi of disguise is discussed and then the judgment is taken. The model involves analysis of URL, page source, website certification and form structure. Decisions are strictly made based on the live behavior of the web-pages and hence, unlike almost all other detection systems, the real time content on the web page is analyzed at runtime and take the decisions accordingly. It is obvious that getting a noise-free list of live phishing URLs is nearly impossible as the lifespan of phishing pages is small. Without having any blacklist, complex and expensive machine learning classifiers, this model is effective enough. Extensive tests on the model on 8000 URLs show that the true positive rate is more than 93% and the false positive rate is below 0.5%.

The model for zero-hour phishing and insecure site detection that is based on

fundamental characteristics of a phishing site and the innate differences between a phishing site's behavior versus a normal site's behavior is presented here. The common patterns of phishing websites is surveyed from a small dataset of 200-300 phishing URLs and these patterns are used to detect them.

The model is very effective and achieves over 93% detection rate with a low false positive rate of less than 0.5%. In contrast, the previous best zero-hour phishing classifier [49] achieved a detection rate of 73% with a lower false positive rate of 0.03% (more comparisons below).

6.1 Evaluation Dataset

The final experiment is tested on 4000 legitimate URLs and 4000 phishing URLs. Each type has two sets. The first set is used for feature extraction and training, and the second set is used for final testing on legitimate URLs. The first set is the set of 1000 random URLs constructed from Alexa top-ranked URLs (as mentioned in 3) from rank-4000 to rank-10000. It is preferred to extract URLs starting from rank-4000 to avoid the URL directly getting classified as legitimate by whitelist function. The second set is actual testing set with 4000 random URLs from Alexa rank-4000 to rank-10000 URLs. Less than just 20 URLs were common in both sets, and as the experiments are on 8000 URLs, it will not make any huge difference on the results.

The phishing URLs are extracted from phishtank but at different times to avoid duplicates in training and testing. For training, 200-300 URLs from json files starting from first week of November 2013 till February 21, 2014 were used. To extract URLs

for the final testing, json file from Phishtank dated 22 February 2014 is used and filtering steps are applied as explained in 3 and random 4000 URLs are recorded.

6.2 Detection Algorithm

The analysis is mainly focused on real-time behavior of the pages and on similarity detection and hence a need of some preprocessing to get the input data. The structure can be divided in four main parts. First part is preprocessing, second is analysis of URL, third is actual body/behavior analysis and last is filtering with search function.

The model is composed of URL based classifiers as well as content based classifiers. The algorithm is designed in such a way that first the URL will be checked to be error free using preprocessing and sanitization steps discussed in 4.1. If any error occurs while getting response from the web page this step bypasses the URL from further analysis. Then the domain of the URL is checked in the whitelist 4.2 and bypassed as legitimate if a match is found. Then the page content is grabbed and checked if the page is asking any sensitive information using password fields 4.3. If not a single password field was detected in the page, then the URL is bypassed as legitimate. Figures 6.1 and 6.2 show the analysis of input URLs.

At the next stage, if the URL is not bypassed by any of the steps above then different classifiers are applied to check the phishing behavior of the page. From URL based classifiers, the model uses targets in URL 5.2.1 and misplaced TLD 5.2.2. Then from content based classifiers, the page is checked for more redirections 5.3.1, copy detection 5.3.2, unsecured password handling 5.3.3 and behavior analysis is done by

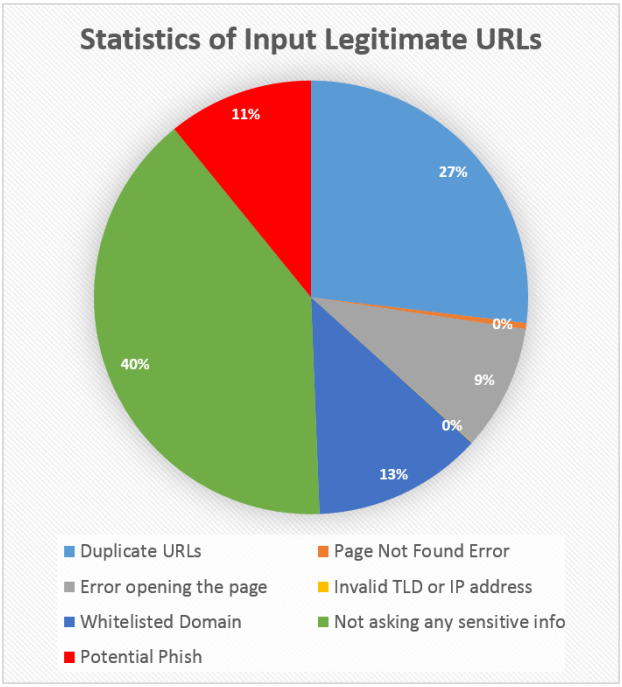

Figure 6.1: Statistics of preprocessing on Legitimate URLs

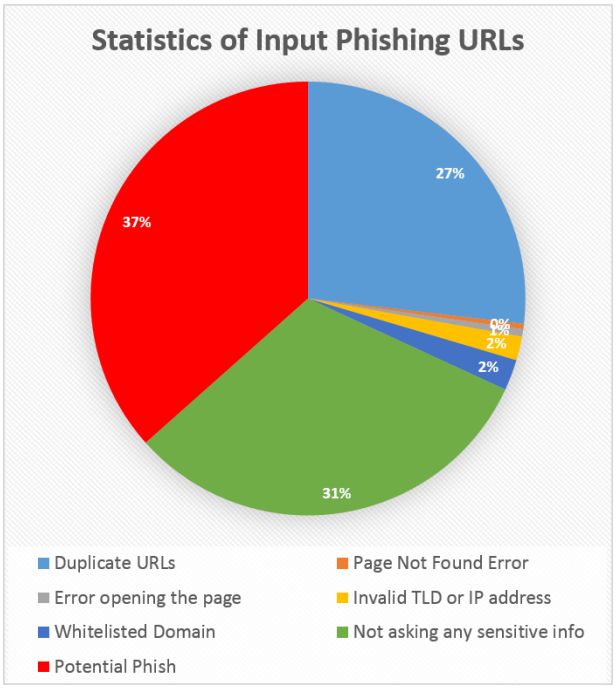

Figure 6.2: Statistics of preprocessing on Phishing URLs

real-time login bot 5.3.4.2. The potential of the URL of being phish is calculated by the total number of classifiers marking the URL as phishing.

If the potential is less than two then the model marks the URL as legitimate. Else, phishing flag is set and the URL goes for search based filtering 5.5. If it marks the URL as legitimate then the final decision will be legitimate, otherwise it is marked as phishing.

To summarize, following are the classifiers used in the model:

- URL based
  - Targets in URL
  - Misplaced TLD
- Content Based
  - More Redirections
  - Copy Detection
  - Unsecured Password Handling
- Behavior Based
  - Real-time Login Bot

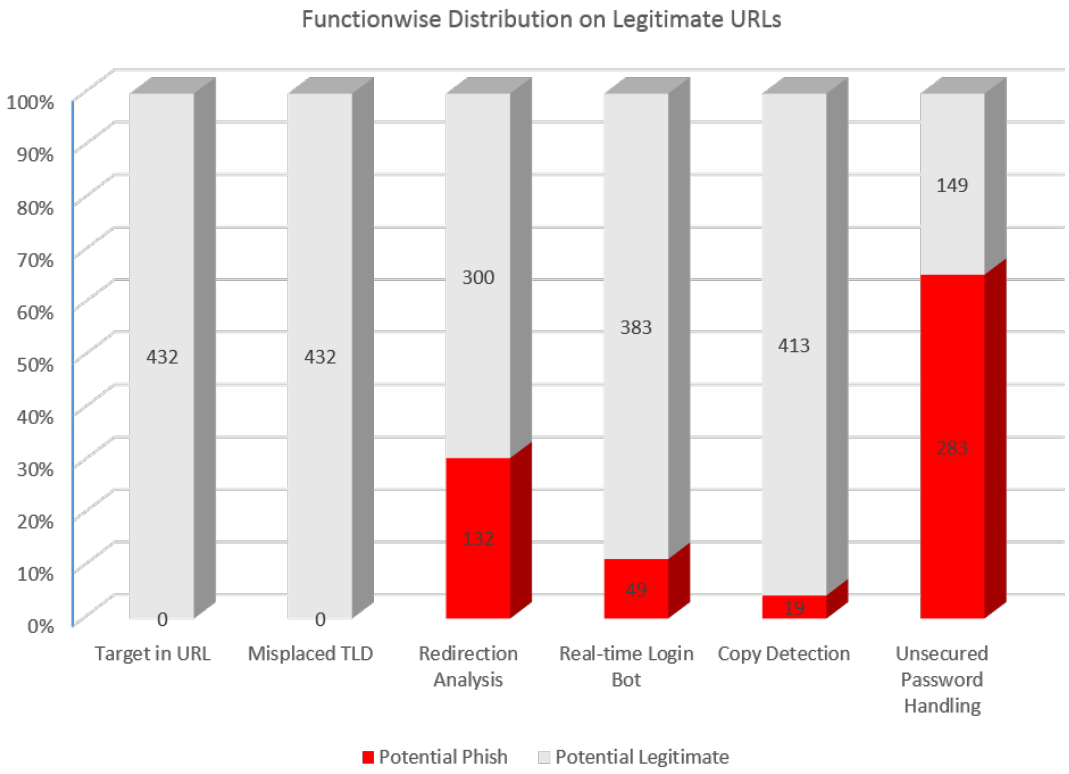

Figure 6.3: Statistics for Individual Classifiers on Legitimate URLs

6.3 Summary of Result

To check the performance of each classifier, the model runs them separately and the individual contribution is shown in the figures 6.3 and 6.4

The final results on 8000 URLs yield the following values:

True positive rate =1345/1440= 93.40%, False positive rate = 11/2432 = 0.46%, Precision = 1345/1356= 99.19% and F-score = 96.21%.

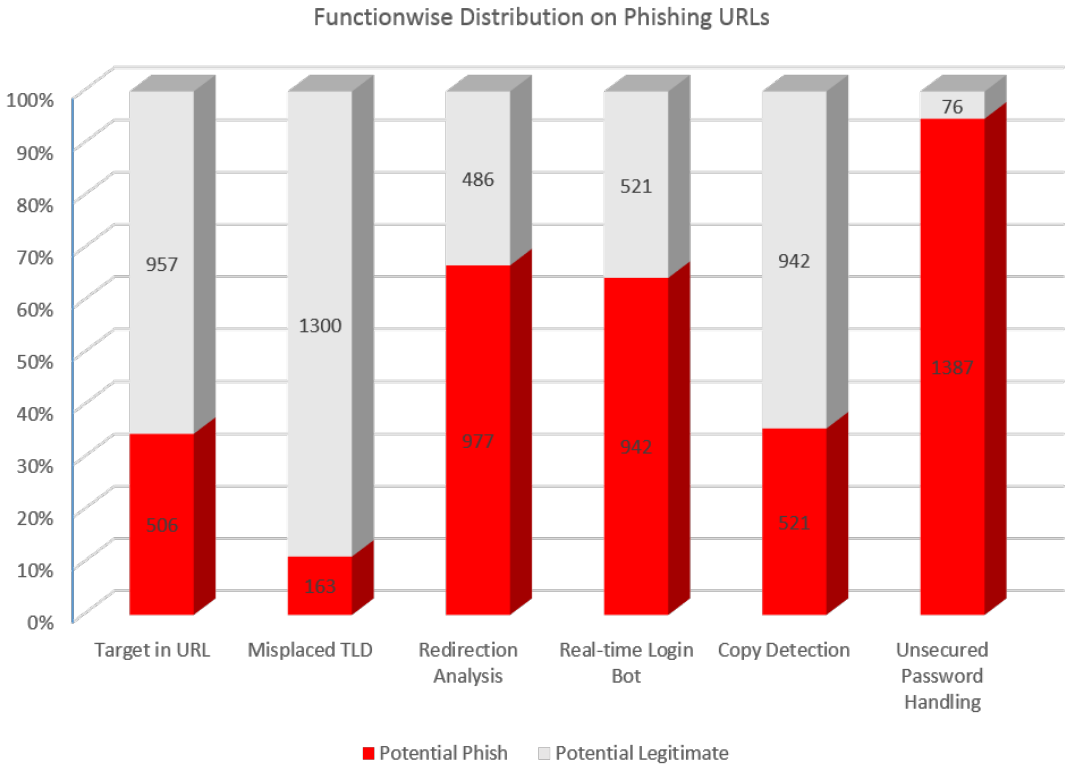

Figure 6.4: Statistics for Individual Classifiers on Phishing URLs

6.4 Conclusion

The main advantage of the model is robustness and use novel techniques of copy detection and real-time web login behavior apart from some other simplifications and improvements over existing methods. It performs competitively with the best previous methods and has the advantage of not needing periodic retraining as for most of the previous methods. One direction for future work is to combine it with a malware detector to detect and thwart sites that do not try to steal sensitive information but install malware on the web site visitor's machine.

Chapter 7

Model Using Combination of URL and Content-Based Classifiers

In this chapter, effective and comprehensive classifiers is presented for both kinds of attacks, classical or hijack-based. The techniques are also effective at zero-hour phishing web site detection. The main focus is on the fundamental characteristics of phishing web sites and decomposing the classification task for a phishing web site into a set of URL based classifier, a set of content-based classifier and ways of combining the two. Results of these classifiers and combination schemes are shown on datasets extracted from several sources. It has been found that: (i) the set of URL classifiers is highly accurate, (ii) the set of content-based classifiers achieve good performance considering the difficulty of the problem and the small size of the white list, and (iii) one of the combination methods achieves superior detection of phishing web sites (over 93%) with low false positives (less than 0.9% without Internet search and 0.2%

with Internet search). Moreover, the set of content-based classifier used in this model does not need any periodic retraining.

For classical attacks, a set of accurate URL classifiers is presented using some new ideas. Three combination schemes are proposed that combine the judgments of the set of URL classifiers and the set of content-based classifiers. Another advantage of the model is the language independence.

The common patterns of phishing websites are studied from a small dataset of 200-300 phishing URLs and used for the detection. All classifiers and the three combination schemes are evaluated separately. The best combination scheme achieves over 93% detection rate with a reasonable false positive rate of 0.9% without any Internet search, and 0.2% false positive rate with Internet search. In contrast, the previous best zero-hour phishing classifier [49] achieved a detection rate of 73% with a false positive rate of 0.03% (more details below) with Internet search.

7.1 Evaluation Dataset

Final experiment is tested on 4000 legitimate URLs and 4000 phishing URLs. The legitimate URLs are same as explained in the previous chapter i.e. each type has two sets. First set is used for feature extraction and training, and second set is used for final testing on legitimate URLs. First set is the set of 1000 random URLs constructed from Alexa top-ranked URLs (as mentioned in 3) from rank-4000 to rank-10000. It is preferred to extract URLs starting from rank-4000 to avoid the URL directly getting classified as legitimate by whitelist function. Second set is

actual testing set with 4000 random URLs from Alexa rank-4000 to rank-10000 URLs. Less than just 20 URLs were common in both sets, and as the experiments are on 8000 URLs, it will not make any huge difference on the results.

But the phishing set is different. The phishing URLs are extracted from phishtank but at different times to avoid duplicates in training and testing. For training, 200-300 URLs from json files starting from March 13, 2014 till April 6, 2014 were used. To extract URLs for the final testing, json file from Phishtank dated 7 April 2014 is used and filtering steps are applied as explained in 3. Out of 8488 phishing URLs, 6000 online and error free URLs are extracted and random 4000 URLs from them are recorded.

7.2 Detection Algorithm

As in every algorithm, the input should be preprocessed and verified legal for the algorithm. The structure can be divided in five main parts. First part is preprocessing4.1, whitelist 4.2 and sensitive information check 4.3, second is analysis of URL, third is actual body/behavior analysis, fourth is different combinational schemes and last is filtering with search function. In preprocessing, if any error occurs while getting response from the web page this step bypasses the URL from further analysis. Then the domain of the URL is checked in the whitelist and bypassed as legitimate if a match is found. Then the page content is grabbed and checked for sensitive information using password fields 4.3. If not a single password field was detected in the page, then the URL is bypassed as legitimate. The statistics of the URLs is clear

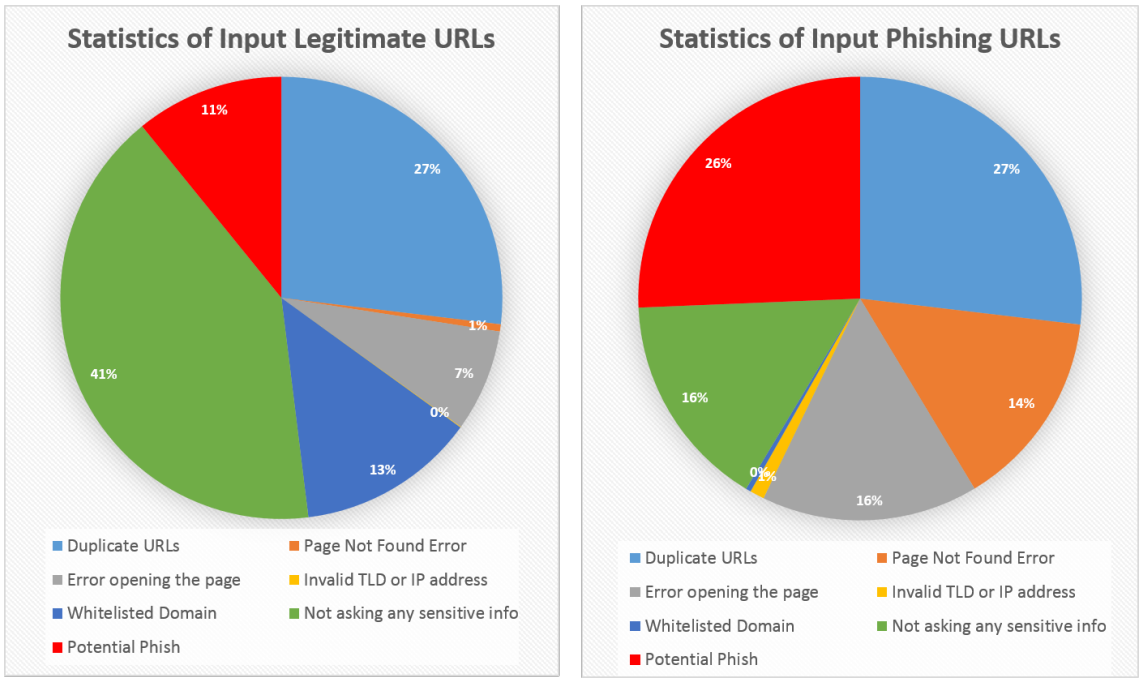

Figure 7.1: Statistics for Input URLs

from the Figure-7.1

Three classifiers from URL classifier section 5.2 and four classifiers from content based classifiers 5.3 are used in this algorithm. There is a vast study done on URL-only classifiers and its combination with content classifiers. But, this chapter describes the experiments based on just URL classifiers, just content based classifiers and various combinations of both.

7.2.1 Overall URL classifier

The final URL classifier score is a logical *OR* of the decisions of the (i) Targets in URL (ii) Misplaced TLDs and (iii) the Machine-learning based URL classifier 5.4.6,

which means a URL is classified as phishing if any of these three judges returns true for phishing and legitimate if none of them returns true.

7.2.2 Overall Content-based Classifier

The final content-based classifier score is a logical *OR* of the decisions of the (i) More redirections (ii) Real time form analysis, and (iii) Copy detection and (iv) Unsecured Password Handling, which means a URL is classified as phishing if either of these four judges returns true for phishing and legitimate if none of them returns true.

To summarize, following are the classifiers used in the model:

- URL based
  - Targets in URL
  - Misplaced TLD
- Content Based
  - More Redirections
  - Copy Detection
  - Unsecured Password Handling
- Behavior Based
  - Real-time Form Analysis
- Machine Learning Based

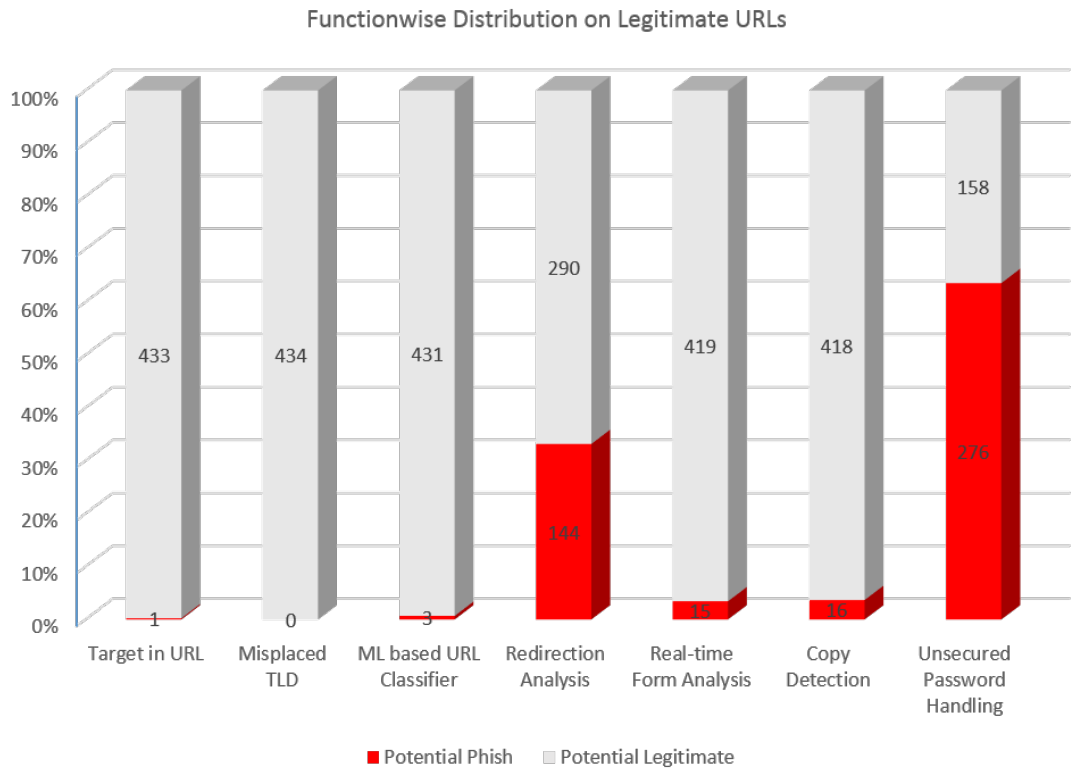

Figure 7.2: Statistics for Individual Classifiers on Legitimate URLs

– Combined URL classifier

The contribution of each classifier can be seen in Figure 7.2 and Figure 7.3.

7.2.3 Combination Schemes

Decision of each classifier is recorded and it is used in the schemes. For the final decision, there are three combination schemes. The first scheme (called *OR scheme*

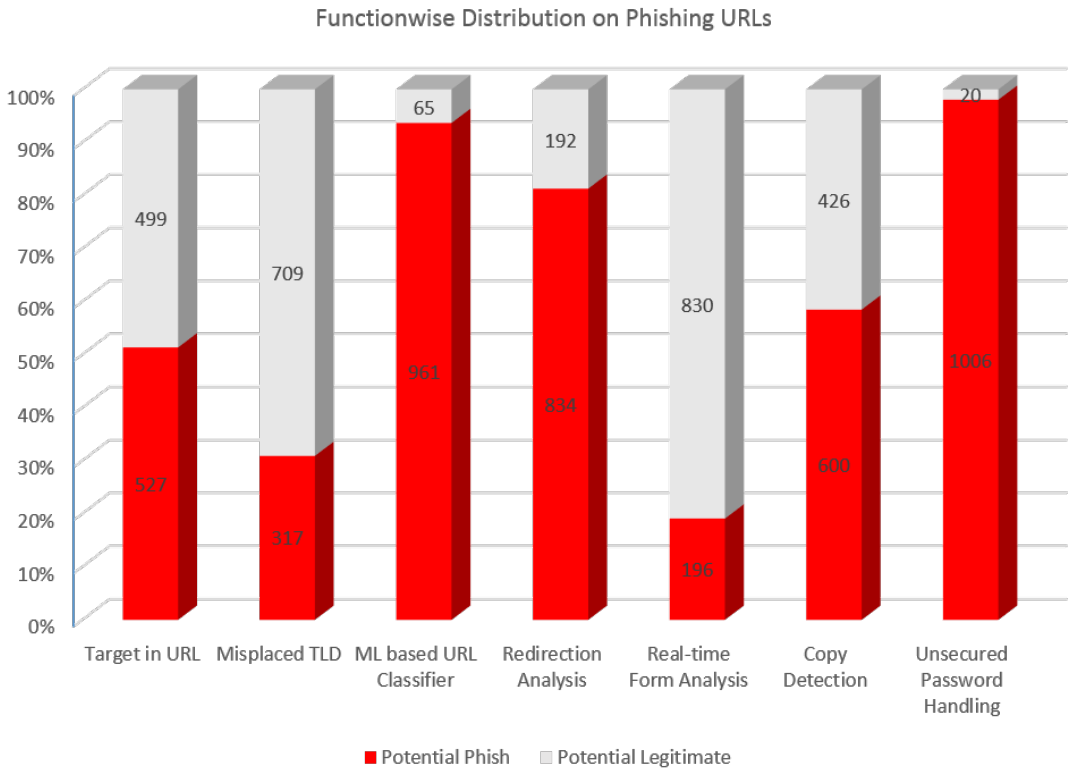

Figure 7.3: Statistics for Individual Classifiers on Phishing URLs

Table 7.1: Results for URL and Content-based classifiers (without Internet search)

	URL-Classifier scheme	Content-based Classifier scheme
TPR	91.71%	97.66%
FPR	0.92%	28.57%
PR	99.58%	88.99%
F-Score	95.48%	93.12%

hereafter) takes the logical *OR* of the results of the URL classifier and the content-based classifier, which means the page is marked as phishing if either of them declares the page to be a phishing page. The second scheme (called *AND scheme* hereafter) computes the logical *AND* of the URL classifier and the content-based classifiers. The third scheme (called *potential scheme*) works on the potential of the phishing nature. It calculates how many component functions of the two classifiers classify the page as phishing and keeps the count as potential. If the potential is at least two, then the page is marked as a phishing page.

7.2.4 Search-based filtering

At the very end, if the page is marked as phishing, search-based filtering is applied. However, this is an optional step as discussed earlier and all the results are also reported separately.

7.3 Summary of Results

Following are the respective matrices of the rates.

Table 7.2: Measurements of Combination of schemes

	With Search-based Filtering			Without Search-based Filtering		
	OR	AND	Potential	OR	AND	Potential
TPR	98.54%	93.47%	97.37%	99.90%	93.47%	98.35%
FPR	28.57%	0.23%	8.99%	78.34%	0.92%	24.42%
PR	89.07%	99.89%	96.24%	75.09%	99.58%	90.48%
F-Score	93.57%	96.57%	96.80%	85.73%	96.43%	94.21%

All the classifiers serve different purposes. URL classifier exclusively works on the given URL and can be considered to be doing static analysis. Even if the system is not connected to the internet, this scheme will work. Another advantage of this scheme is that this scheme is faster than the other schemes.

Content-based classifier will help to see the live and dynamic content of the web page. Hence, the analysis will be completely real-time. Real-time helps in taking the decisions on runtime and can detect hijack-based attacks. We know that the lifetime of the phishing URLs is very short. And even if the URL started hosting genuine content, unlike the content-based scheme, almost all other detection techniques would still classify the URL as phishing. This analysis is completely dynamic and requires a stable internet connection. Combination of these schemes makes the detection technique more accurate, robust, reliable and real-time. The above statistics show that the AND scheme performed the best, and search-based filtering has a small effect on its TPR as well as FPR and hence PR and F-Score.

7.4 Conclusion

In this chapter, a comprehensive solution is presented which is robust and uses novel techniques in the URL classifier (e.g., character frequencies, KS-distance) and in the content-based classifier (e.g., similarity detection using F-score and real-time web page behavior) apart from some other simplifications and improvements over existing methods. It performs competitively with the best previous methods. Furthermore, the important problem of hijack-based phishing attacks is also addressed through the content-based classifier and the problem of zero-hour phishing detection as well. The content-based classifier has the advantage of not needing any periodic retraining and the URL classifier also requires minimal training, which is fast and efficient. Even though character frequencies are used in URL classifier, the methods are still language independent as the experiments with the dataset show.

Chapter 8

Model Using Just Content-Based Classifiers

In this chapter, more robust, and more effective classifiers for classical as well as hijack-based attacks are presented, with a focus on the latter kind. The work is the first to consider hijack-based phishing attacks according to the literature study. The techniques could also be effective for zero-hour phishing web site detection. The focus is on the fundamental characteristics of attack web sites and introduce new features and techniques for detection. These classifiers and combination schemes produced good results on datasets extracted from several sources. The content-based classifier achieves good performance considering the difficulty of the problem, various patterns of phishing pages and the small size of the white list. One of the combination schemes achieved superior detection of phishing web sites over 92% with low false positives of less than 0.7% (without Internet search) and 0% false positives

is also possible with reasonable detection rate of over 74% (with Internet search). Moreover, the behavior-based classifiers do not need any retraining. The methods are also language independent and hence the model can work on phishing pages in any language.

8.1 Evaluation Datasets

Final experiments are conducted on two diverse datasets. Fresh phishing URLs are necessary for testing as the focus is in making the decision on real-time behavior of the page. Hence, freshly reported phishing URLs from phishtank on May 29, 2014 are gathered. The availability of the content on the URL is confirmed by asking for the response. 10,000 such online and error free phishing URLs are extracted and randomized to get 4000 URLs in phishing dataset for testing. Availability of the legitimate URLs can be trusted and hence 4000 random URLs from top 10,000 Alexa (as of mid November of 2013) domains and another 4000 random URLs from DMOZ are directly added to dataset. Two separate test data-sets called TestingSet-1 and TestingSet-2 are created. TestingSet-1 contains combination of 4000 'phishtank' phishing URLs and 4000 'DMOZ' legitimate URLs, and TestingSet-2 maintains combination of 4000 'phishtank' phishing URLs with 4000 'Alexa' legitimate URLs. It is confirmed that there is not a single overlapping URL between any phishing URL from TrainingSet and TestingSet. Also, not a single URL is found common between 4000 'Alexa' and 4000 'DMOZ' URLs. The robustness of the model is tested,

validated and no over-fitting is confirmed by training and testing on completely exclusive datasets. If the training is done with TrainingSet-1 ('phishtank' with 'Alexa' URLs) then testing is done on TestingSet-1 ('phishtank' with 'DMOZ'), and same for TrainingSet-2 and TestingSet-2. Not a single URL is repeated in either of the combinations of the training and testing sets.

8.2 Detection Algorithm

After preprocessing 4.1, whitelist 4.2 and sensitive information check 4.3, the test URL is processed with different content based and machine learning based classifiers. The model makes use of the advantages of classifiers trained with machine learning as well as heuristic based classifiers without a single URL classifier. Machine learning based classifiers improves the efficiency while heuristic based classifiers gives the model sustainability, robustness and zero hour detection capability.

Redirection Analysis and *Copy Detection* are the two machine learning based classifiers 5.4. *Real-time Form Analysis* and *Real-time Login Bot* are the heuristic (behavior) based classifiers 5.3.4

To summarize, following are the classifiers used in the model:

- Machine Learning Based
  - Redirection Analysis
  - Copy Detection

- Behavior Based
  - Real-time Form Analysis
  - Real-time Login Bot

To test the robustness and variations of classifiers, four different combinations are designed from the individual output of the classifiers.

Combination-1 (or scheme-1) is the simplest combination of all the other combinations. It just checks if at least two of the classifiers mark the URL as potentially phishing. If so, then mark the URL as phishing else mark legitimate. *Combination-2* (or scheme-2) checks if at least two classifiers from redirection analysis, copy detection and real-time form analysis marks an URL as phishing. If they do, then mark the URL as phishing else legitimate. *Combination-2* and *Combination-3* (or scheme-3) are similar, but *Combination-3* checks the results from redirection analysis, copy detection and real-time login bot. If at least, two classifiers mark the URL as phishing then the URL is marked as phishing else legitimate.

Combination-4 (or scheme-4) is more effective. It combines the pros and cons of the two classifiers viz. real-time form analysis and real-time login bot. It first calculates the boolean *OR* of the output of these two classifiers . In layman terms, it checks if either of the classifiers marks the URL as phishing and records the output. This output is checked with other two classifiers (redirection analysis, copy detection) and if at least two classifiers mark the URL as phishing then the URL is marked as phishing else legitimate.

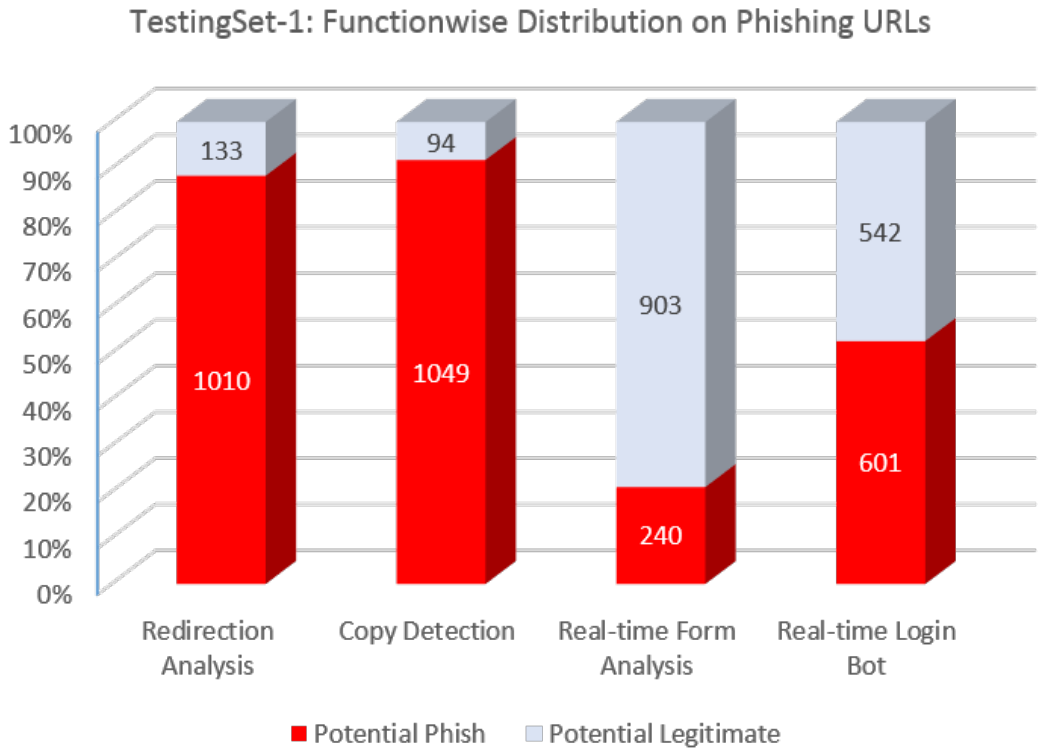

Figure 8.1: Statistics for Individual Classifiers on Phishing URLs

8.3 Results and Evaluation

It is important to measure the individual contribution of the classifiers. The results are from each classifiers independently and the various combinations make decision based on it. Figures 8.1, 8.2, 8.3, 8.4 demonstrate the output of each classifier separately on different TestingSets with and without *Search based Filtering*. For TestingSet-1, out of 4000 URLs, 1824 phishing and 1950 legitimate URLs gave error free responses and considered in the classification. Similarly, 3425 legitimate and 2240 phishing URLs are valid from TestingSet-2.

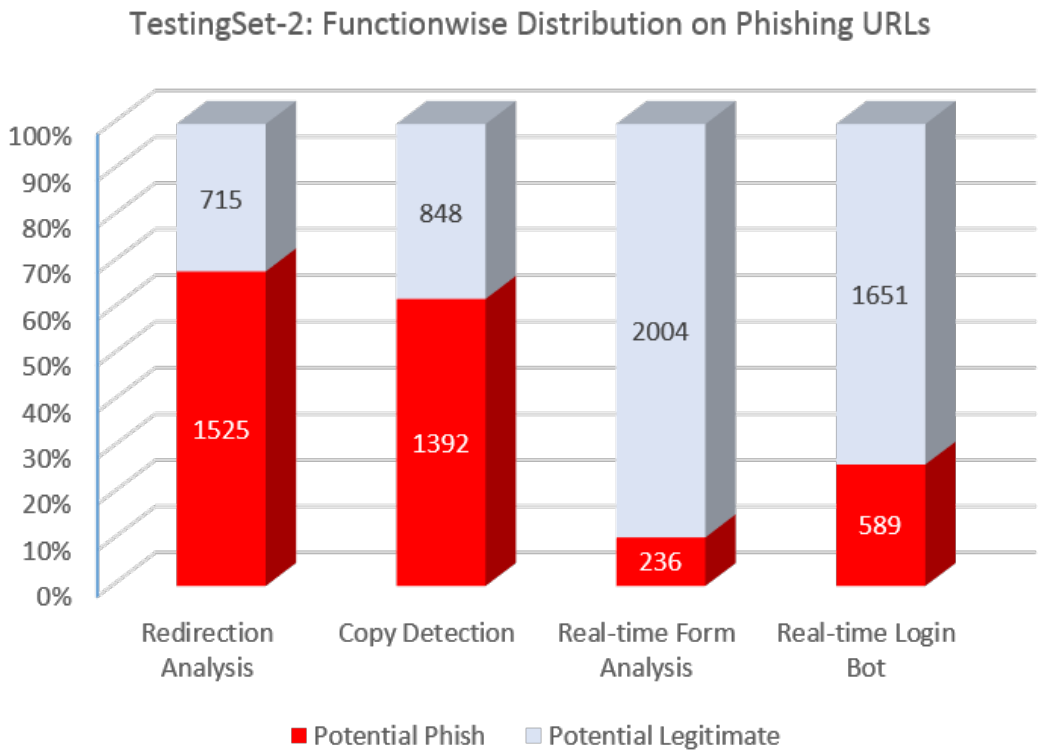

Figure 8.2: Statistics for Individual Classifiers on Phishing URLs

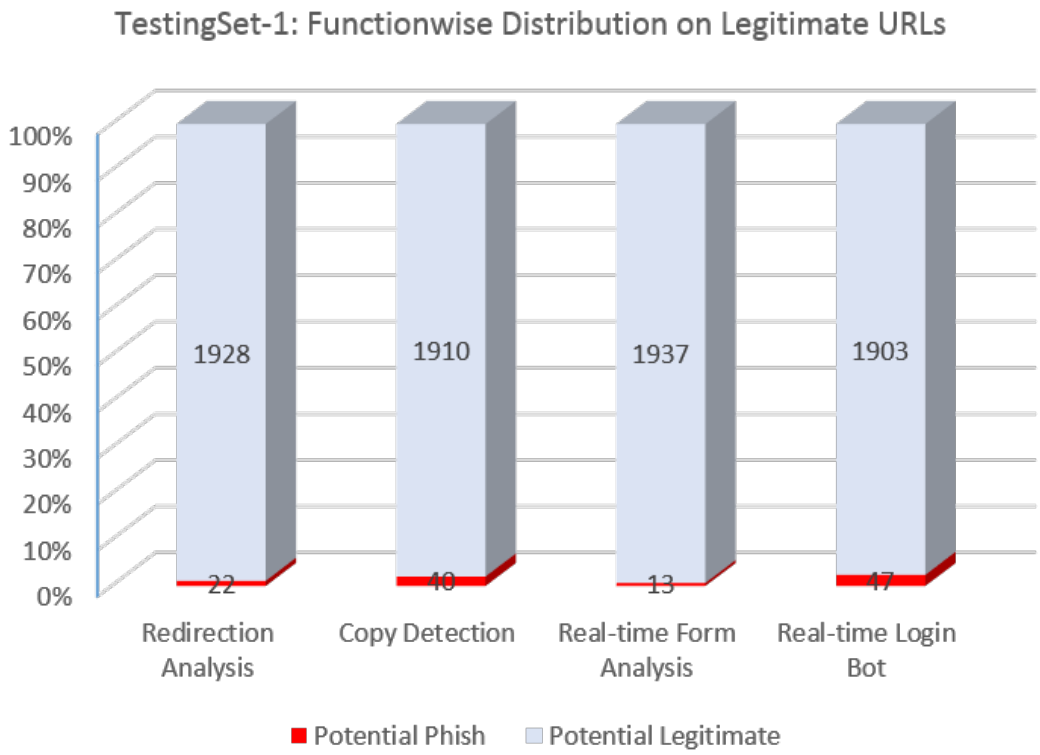

Figure 8.3: Statistics for Individual Classifiers on Legitimate URLs

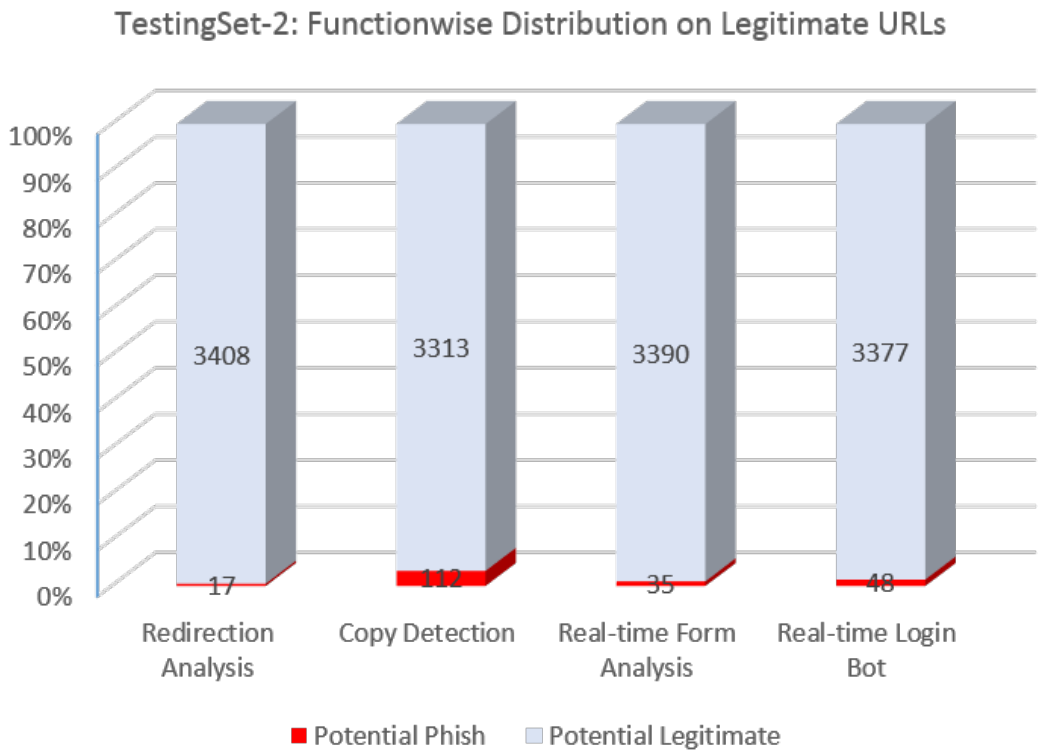

Figure 8.4: Statistics for Individual Classifiers on Legitimate URLs

Table 8.1: Model Performance without Search Based Filtering (TestingSet-1)

	TPR	**FPR**	**PR**	**F-score**
Combination-1	92.74%	1.08%	98.0674%	95.32%
Combination-2	86.96%	0.21%	99.60%	92.85%
Combination-3	88.45%	0.36%	99.31%	93.57%
Combination-4	92.21%	0.67%	98.78%	95.38%

Table 8.2: Model Performance with Search Based Filtering (TestingSet-1)

	TPR	**FPR**	**PR**	**F-score**
Combination-1	74.63%	0.00%	100.00%	85.47%
Combination-2	69.99%	0.00%	100.00%	82.35%
Combination-3	71.30%	0.00%	100.00%	83.25%
Combination-4	74.28%	0.00%	100.00%	85.24%

The testing in done under different environments. Not a single training URL is overlapped with testing URL. The model provides flexibility to have 'Search Based Filtering' mode on and off. Complete measurements of the results is shown in the tables (Table 8.1, 8.2, 8.3 and 8.4).

Table 8.3: Model Performance without Search Based Filtering (TestingSet-2)

	TPR	**FPR**	**PR**	**F-score**
Combination-1	93.23%	1.52%	96.59%	94.88%
Combination-2	89.11%	1.26%	97.04%	92.91%
Combination-3	89.61%	1.28%	96.31%	92.84%
Combination-4	93.04%	1.40%	96.84%	94.90%

Table 8.4: Model Performance with Search Based Filtering (TestingSet-2)

	TPR	**FPR**	**PR**	**F-score**
Combination-1	80.70%	0.06%	99.84%	89.25%
Combination-2	77.15%	0.03%	99.92%	87.07%
Combination-3	79.18%	0.03%	99.92%	88.35%
Combination-4	80.51%	0.03%	99.92%	89.17%

8.4 Conclusion

This model has no URL-based classifier business and hence it is very tough for phishers to hide from this model. Different combinations in the model have own benefits. Combination-1 gives the best TPR, Combination-2 performs faster, whereas Combination-3 is better in performance than Combination-2 but comparatively slower and Combination-4 is optimal with almost same TPR as Combination-1 with less FPR (close to zero). The model is believed to sustain for a long time because of the behavior (heuristic) based classifiers. All the classifiers and hence the model is language independent.

Chapter 9

Performance

The chapter describes the adversarial attacks, summary of the results of the various models, and direct comparison of one of the model with Google's Safe Browsing system.

9.1 Security Analysis

The determined phisher who reads this work can try and thwart detection. Such potential adversarial attacks are explained here. Copying a legal web-site's content is *almost* a necessary step to lure victims (the only other mechanism is to offer some kind of incentive to people, which may not attract many victims since these strategies are quite dated now, e.g., the Nigerian emails [46]).

Hiding from the URL classifiers looks easy but it is not practical. Phishers need

to register for SSL and hence need to provide complete information about them for verification process. This reveals the identity of the phishers. It also adds to the cost of the design. Other way is to host phishing page on highly reputed domain, buy more hosting space to add multimedia (like images, videos) etc. All this strategies reveals the information of the phishers and/ or reduces the cost-benefit ratio. Even though, phishers manage to host phishing content on highly reputed domain, with SSL, without any targets in URL etc., the content based classifiers will detect the phishing attempt and take the decisions accordingly. Hence, the clever phisher must make use of all such URL hiding strategies along with some changes in the content and behavior of the page. Following are the adversarial strategies for non URL based classifiers.

First, phishers must ensure that the number of external links is smaller than the number of internal links. Second, they must change the behavior of the website to that of a legal one to avoid detection, which means that the site should show an error message or two and then keep the user on the same page with asking credentials again. There are a couple of problems with this approach. First, this means that the phisher cannot be lazy and use some kind of kit for building the site. Thus the work of the phisher is increased and the cost-benefit ratio becomes less attractive. Second, a user that is redirected to the same page after entering valid credentials may smell a rat very quickly after a few attempts to get into what seems like the legal site and thus the time left for the phisher to carry out any exploits on the user's accounts will be diminished. Responsible user will report the URL as phishing and the URL would be blacklisted sooner. Even if all the phisher does is sell those credentials in

some underground network, those credentials will be usable for a shorter period and such credentials will be worth less and less over time.

Legal websites that receive bogus information from the real-time login bot are not really affected by the approach, since there is no essential difference between random information injected by the bot versus mistyped information from a registered user.

9.2 Overall Result

The work produced three main models as discussed in chapters- 6, 7 and 8. Each model has different combinations and it is better to report the results of the best combination from each of the main models. There is a single scheme from *Model Using Real-time Bots And Fundamental Characteristics*. *AND* scheme from *Model Using Combination of URL and Content Based Classifiers* is better than *OR* and *potential* scheme. *Combination-4* from *Model Using Just Content Based Classifiers* outperformed over other combinations. Let's call them scheme-1, scheme-2 and scheme-3 respectively.

All the experiments are done on 4000 phishing URLs and 4000 legitimate URLs. As the datasets are different for all schemes, readers are requested not to compare the results directly. The table-9.1 summarizes the results of the schemes.

Table 9.1: Result of Different Schemes

	Search Based Filtering	**TPR**	**FPR**	**PR**	**F-score**
Scheme-1	On	93.40%	0.46%	99.19%	96.21%
Scheme-2	On	93.47%	0.23%	99.89%	96.57%
Scheme-2	Off	93.47%	0.92%	99.58%	96.43%
Scheme-3	On	74.28%	0.00%	100.0%	85.24%
Scheme-3	Off	92.21%	0.67%	98.78%	95.38%

9.3 Direct Comparison

Each browser, now a days, has integrated phishing detector. Also, extensions (also called as plugins) like netcraft [32],Web of Trust [43] are available to even provide more security to the users. Antivirus programs like *McAfee* install extra layer of protection from phishing URLs to the browser. The only way compare the models with such detection systems is to manually visit each of the URLs from dataset and see the visual response on the web browser. This is very tedious and almost impossible way to get the results for each of the URLs from such a huge dataset. It has been also tried to get tools phishing detection from other researchers, but they either didn't have any public API or they did not respond for the request of the detection system.

The work done by Xiang et al. [49] can not be directly compared to this work as the datasets are different. However, their TPR of 67.74% with 0% FPR is less than TPR of all the models with same FPR.

Fortunately, Google has API for phishing detection called as Google Safe Browsing. *Safe Browsing is a Google service that enables applications to check URLs against Google's constantly updated lists of suspected phishing and malware pages* [21].

To compare all models again with Google's safe browsing API, 17200 freshly reported, verified phishing URLs on PhishTank are taken from July 15, 2014. Similarly, 17200 legitimate URLs are gathered from DMOZ set. All the models including Google's safe browsing is tested on the same URLs at the same time. Figure-9.1 shows the performance of individual classifiers on those URLs. To summarize, following are the abbreviations used for the classifiers in the figure.

Abbreviations for classifiers in Figure-9.1:

- U1: Targets in URL
- U2: Misplaced TLD
- U3: Combined URL
- C1: More Redirection
- C2: Copy Detection
- C3: Unsecured Password Handling
- C4: C1 with Machine Learning
- C5: C2 with Machine Learning
- B1: Real-time Form Analysis
- B2: Real-time Login Bot

And, then different models are tested with mentioned combinations and the final results are shown in table-9.2.

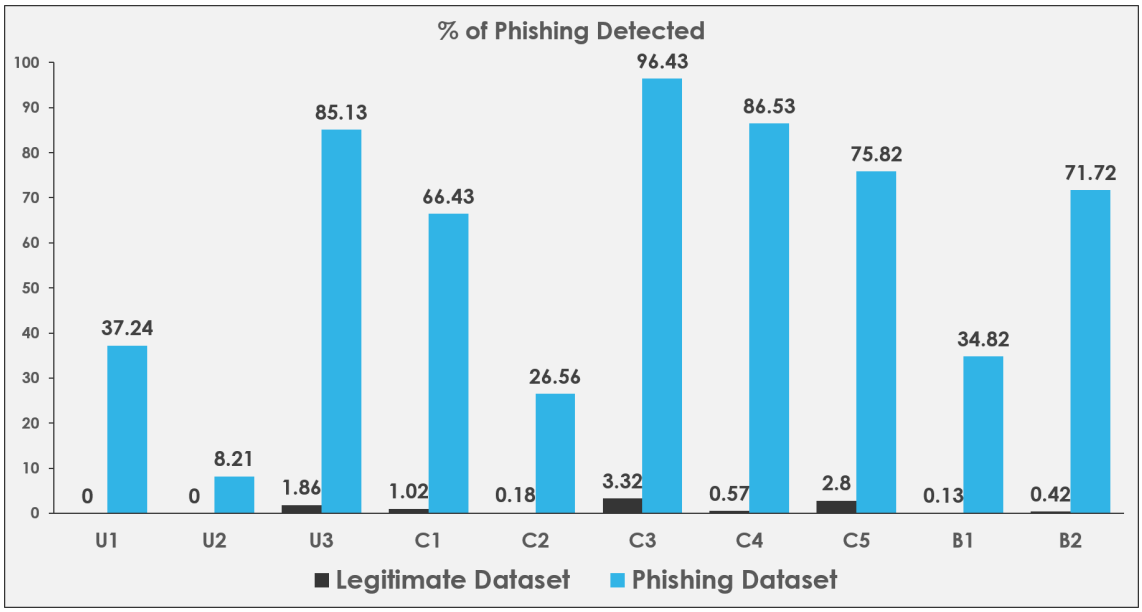

Figure 9.1: Statistics for Individual Classifiers on Legitimate URLs

Table 9.2: Direct comparison of all models with Google Safe Browsing

Model	Combinations	Search Based Filtering = OFF				Search Based Filtering = ON			
		TPR	FPR	PR	F-score	TPR	FPR	PR	F-score
Model-1		93.01	1.24	96.25	94.60	87.18	0.28	99.08	92.75
Model-2	1	**99.97**	3.50	88.25	93.75	93.37	0.54	97.84	95.55
	2	87.64	1.80	92.76	90.13	82.30	0.22	98.98	89.88
	3	97.94	2.48	91.24	94.47	91.55	0.36	98.52	94.91
Model-3	4	86.51	0.63	97.34	91.60	80.92	0.12	99.46	89.23
	5	77.43	0.48	97.74	86.41	72.20	0.08	99.57	83.71
	6	80.11	0.48	97.81	88.08	**75.09**	**0.08**	99.59	85.62
	7	86.22	0.62	97.39	91.46	80.63	0.11	99.49	89.07
GSB		**51.46**	**0.03**	99.80	67.91				

The abbreviations used in table-9.2 are:

- Model-1: Model Using Real-time Bots And Fundamental Characteristics (Chapter-6)
- Model-2: Model Using Combination of URL and Content Based Classifiers (Chapter-7)

  Combination-1: OR scheme

  Combination-2: AND scheme

  Combination-3: Potential scheme
- Model-3: Model Using Just Content Based Classifiers (Chapter-8)

  Combination-4: Scheme-1 (as discussed in the chapter)

  Combination-5: Scheme-2

  Combination-6: Scheme-3

  Combination-7:Scheme-4

It is found that, Google's safe browsing is very good in maintaining the FPR always close to zero, but it failed miserably in detecting the phishing sites.

Chapter 10

Conclusion

10.1 Challenges and Future Work

The very first challenge is to get noiseless datasets. Many of the verified URLs were off-line within few seconds. Also, genuine domains like paypal.com are found reported and verified in phishtank dataset. Not a single good whitelist was publicly found and hence creating a clean whitelist was a challenging task. Getting the list of targets required data mining through large dataset from phishtank and study of surveys from APWG and other research papers. Studying the infinite patterns of the phishing sites and designing rules which will cover most of the patterns was another challenge apart from implementing those in login bot.

Another big challenge was to improve the classifiers and decide perfect combinations (and logical schemes of such combinations) of specific classifiers.

In future work, it is planned to make the *Real-time Login Bot* more advanced by including complex rules to distinguish phishing behavior from regular behavior. The whitelist is evolving and by making it larger (or using a big and verified whitelist), the detection rate will automatically increase. PART algorithm is used for its simplicity and accuracy. Other complex machine learning algorithm can be implemented as per requirements after checking the tradeoffs between accuracy and speed. By publishing an API, the detection system is planned to release to public. An extension for web browsers is also possible to benefit users.

10.2 Conclusion

In this thesis, fundamental characteristics of websites are discussed. They are categorized as URL, content and behavioral characteristics. Almost all the effective classifiers based on such characteristics are developed and analyzed. Different models with different combinations make use of advantages of various classifiers and they add to the efficiency of the model. All the models are capable of having F-score more than 85% even on large dataset of 34400 URLs. One of the model does not use a single URL based features, which makes the phishers very difficult to hide from such technique and still performed good with small FPR. The classifiers are very robust even though they are trained on small and noisy data. One of the combination scheme can be modified for the lowest false positive rate of 0% with reasonable true positive rate of 74.28 %. The classifiers are language independent and work on any languages. Also, the work is supposed to be pioneer in hijack based attacked

detection systems. User reporting, manual verification and daily retraining is not required for all the approaches. It is anticipated that, after the decided improvements mentioned in the future work, the system will be more accurate and stable.

Bibliography

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair. A comparison of machine learning techniques for phishing detection. In *Proc. anti-phishing working group's 2nd annual eCrime researchers summit*, pages 60–69. ACM, 2007.

[2] E. Alpaydin. *Introduction to machine learning*. MIT press, 2004.

[3] . Anti-Phishing Working Group, January June. Phishing activity trends report - h1 2011. In *APWG Phishing Trends Reports*, 2011.

[4] Anti-Phishing Working Group, January - March, 2014. Phishing activity trends report - q1 2014. In *APWG- Phishing Activity Trends Report*, 2014.

[5] R. Basnet, S. Mukkamala, and A. Sung. Detection of phishing attacks: A machine learning approach. *Soft Computing Applications in Industry*, pages 373–383, 2008.

[6] A. Bergholz, J. D. Beer, S. Glahn, M.-F. Moens, G. Paaß, and S. Strobel. New filtering approaches for phishing email. *Journal of Computer Security*, 18(1):7–35, 2010.

[7] A. Bergholz, J. Chang, G. Paaß, F. Reichartz, and S. Strobel. Improved phishing detection using model-based features. In *Proc. Conf. on Email and Anti-Spam (CEAS)*, 2008.

[8] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya. Phishing email detection based on structural properties. In *NYS CyberSecurity Conf.*, 2006.

[9] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell. Client-side defense against web-based identity theft. In *NDSS*, 2004.

[10] R. Dhamija and J. Tygar. The battle against phishing: Dynamic security skins. In *Proc. 2005 symp. on Usable privacy and security*, pages 77–88. ACM, 2005.

[11] K. Dyer and R. Verma. Phishing url classification with statistical machine learning. *Unpublished Manuscript*, 2014.

[12] FBI.gov, U.S. government, U.S. Department of Justice. Common fraud schemes. `http://www.fbi.gov/scams-safety/fraud`, 2014.

[13] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. In *Proc. 16th int'l conf. on World Wide Web*, pages 649–656. ACM, 2007.

[14] E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. 1998.

[15] W. N. Gansterer and D. Pölz. E-mail classification for phishing defense. In *ECIR*, pages 449–460, 2009.

[16] S. Garera, N. Provos, M. Chew, and A. Rubin. A framework for detection and measurement of phishing attacks. In *Proc. 2007 ACM workshop on Recurring malcode*, pages 1–8, 2007.

[17] Gartner. Gartner says number of phishing attacks on u.s. consumers increased 40 percent in 2008. `http://www.gartner.com/newsroom/id/936913`, 2007.

[18] Gartner. Gartner survey shows phishing attacks escalated in 2007. `http://www.gartner.com/newsroom/id/565125`, 2007.

[19] G.-G. Geng, X.-D. Lee, W. Wang, and S.-S. Tseng. Favicon-a clue to phishing sites detection. In *eCrime Researchers Summit (eCRS), 2013*, pages 1–10. IEEE, 2013.

[20] Google. My drive- google drive. `https://drive.google.com/#my-drive`, 2014.

[21] Google Developers. Safe browsing api – google developers. `https://developers.google.com/safe-browsing/`, 2013.

[22] J. Hong. The state of phishing attacks. *Commun. ACM*, 55(1):74–81, 2012.

[23] D. Irani, S. Webb, J. Giffin, and C. Pu. Evolutionary study of phishing. In *3rd Anti-Phishing Working Group eCrime Researchers Summit*, 2008.

[24] M. Jakobsson and S. Myers. *Phishing and countermeasures: understanding the increasing problem of electronic identity theft*. Wiley-Interscience, 2006.

[25] L. James. *Phishing exposed*. Syngress Publishing, 2005.

[26] E. Lee and R. Verma. Automatic detection of phishing sites by matching. *Poster Presentation*, 2013.

[27] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel. On the effectiveness of techniques to detect phishing sites. In *DIMVA*, pages 20–39, 2007.

[28] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Learning to detect malicious urls. *ACM TIST*, 2(3):30, 2011.

[29] D. K. McGrath and M. Gupta. Behind phishing: An examination of phisher modi operandi. In *LEET*, 2008.

[30] D. K. McGrath, A. Kalafut, and M. Gupta. Phishing infrastructure fluxes all the way. *IEEE Security & Privacy*, 7(5):0021–28, 2009.

[31] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi. An evaluation of machine learning-based methods for detection of phishing sites. In *Advances in Neuro-Information Processing*, pages 539–546. Springer, 2009.

[32] Netcraft. Netcraft extension - phishing protection and site reports. `http://toolbar.netcraft.com/`, 2014.

[33] Netcraft Ltd. . .aero air safety site hijacked. `http://news.netcraft.com/archives/2014/04/04/aero-air-safety-site-hijacked.html`, 2014.

[34] Netscape Communications Corporation. Open directory rdf dump. `http://rdf.dmoz.org/`, 2004.

[35] G. Ollmann. *The phishing guide*. Next Generation Security Software Ltd., 2004.

[36] OpenDNS-PhishTank. The phishtank database. `http://www.phishtank.com/developer_info.php`, 2012.

[37] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta. Phishnet: predictive blacklisting to detect phishing attacks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–5. IEEE, 2010.

[38] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell. Stronger password authentication using browser extensions. In *Usenix security*, pages 17–32. Baltimore, MD, USA, 2005.

[39] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang. An empirical analysis of phishing blacklists. In *Proc. 6th Conf. on Email and Anti-Spam*, 2009.

[40] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. A. Kemmerer, C. Kruegel, and G. Vigna. Your botnet is my botnet: analysis of a botnet takeover. In *ACM Conference on Computer and Communications Security*, pages 635–647, 2009.

[41] R. Verma, N. Shashidhar, and N. Hossain. Phishing email detection the natural language way. In *ESORICS*, 2012.

[42] R. Verma, N. Shashidhar, and N. Hossain. Semantic feature selection for text data with application to phishing detection. In *ICISC*, 2012.

[43] Web of Trust. Safe browsing tool — wot (web of trust). `https://www.mywot.com/`, 2014.

[44] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng. Detection of phishing webpages based on visual similarity. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1060–1061. ACM, 2005.

[45] C. Whittaker, B. Ryner, and M. Nazif. Large-scale automatic classification of phishing pages. *Proc. of 17th NDSS*, 2010.

[46] Wikimedia Foundation, Inc. 419 scams. `http://en.wikipedia.org/wiki/419_scams`, 2014.

[47] G. Xiang, J. Hong, C. P. Rose, and L. Cranor. Cantina+: a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)*, 14(2):21, 2011.

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

[49] G. Xiang, B. A. Pendleton, J. Hong, and C. P. Rose. A hierarchical adaptive probabilistic approach for zero hour phish detection. In *Computer Security–ESORICS 2010*, pages 268–285. Springer, 2010.

[50] W. Yu, S. Nargundkar, and N. Tiruthani. Phishcatch-a phishing detection tool. In *33rd IEEE Int'l Computer Software and Applications Conf.*, pages 451–456, 2009.

[51] Y. Zhang, J. Hong, and L. Cranor. Cantina: a content-based approach to detecting phishing web sites. In *Proc. 16th int'l conf. on World Wide Web*, pages 639–648. ACM, 2007.