A comprehensive and efficacious architecture for detecting phishing webpages

Article *in* Computers & Security · February 2014

DOI: 10.1016/j.cose.2013.10.004

CITATIONS
111

READS
751

2 authors:

Ramesh Gowtham
Amrita Vishwa Vidyapeetham
17 PUBLICATIONS **417** CITATIONS

SEE PROFILE

Ilango Krishnamurthi
Sri Krishna College of Engineering and Technology
44 PUBLICATIONS **655** CITATIONS

SEE PROFILE

A Comprehensive and Efficacious Architecture for Phishing webpage detection

Abstract

Phishing is a web based criminal activity of making innocent online users to reveal sensitive information into fake web sites. Such fake web sites lead to fraudulent charges against the individuals and corporations. It is also considered to be one of the most dangerous threats in the internet which may cause grave disappointment to users. Due to the short lifetime and rapid advancement of a phishing webpage - maintaining blacklists, white-lists or only heuristics base approaches are not very effective. The impact of this problem can be greatly reduced by any right combination these techniques. In this paper, we have studied the characteristics of legitimate and phishing web pages and based on that we have proposed heuristics to extract fifteen novel features of the webpage; these heuristic results are inputted to trained machine learning algorithm to find phish. In this system we have used two preliminary screening modules before we start applying heuristics on the webpage. The first module, preapproved site identifier checks the webpage against private white-list maintained by the user and the second module classifies the webpage right away as legitimate when there are no login forms in it. These modules help to reduce unnecessary computation of the system and also reduce the false positives without compromising on the false negatives. By enabling all the modules we would able to classify web pages with 99.8% of precision and 0.4% of false positive rate. The experiment results show that ours is an efficient method for protecting users from online identity attack.

Key Words: Phishing, Anti-phishing, Anti-phishing Framework, E-Commerce security, Machine learning.

1. Introduction

Victims of phishing scams always find their personal or financial information stolen. Credit card numbers, health information, e-mail address login credentials, answers to security questions and other sensitive data are the targets of phishers. Once this information is attained, these details are used to create fake accounts on victim's name, ruin their credits or even prevent the users from accessing their own accounts. One can easily surmise that this leads to one's demise. According to the RSA's online fraud report, there is 19% increase in phishing attacks in first half of 2012 as compared to second half of 2011.Also RSA estimates there is more than $687 million was lost due to phishing attacks by the global organizations in the same period (RSA Anti-Fraud Command Center, 2012).

Statistics from the Anti Phishing Working Group (APWG) shows that banks are the primary target of phishing attack. A major change over the past year (since March 2010) is the shift in banks being used. Targets shifted from being primarily regional banks to nationwide banks. About 65% of phishing attacks go after nationwide banks, 30% target regional banks and only about 5% go after credit unions. There has been some fluctuation in these numbers, naturally. Each one tends to stay within roughly 5-10% of the base number, excepted for the noted shift toward national banks. Social networking sites are now a prime target of phishing, since the

personal details in such sites can be used in identity theft. Microsoft's research report says that, phishing attacks on social networking platforms trick users into giving up sensitive information or clicking on malicious links have increase to 1,200% (Microsoft Security Intelligence Report, 2010). As per the report given by U.S. Computer Emergency Readiness Team recently, such phishing attacks are more commonly done against government networks, accounting to 51.2% of the attacks.

There is an increase in number of phishing websites today that are created using phishing toolkits (e.g., Super Phisher, Rock Phish), as it reduce the amount of work load by stealing the source code of the webpage. This makes their work easier when compared to manual method of creating a phishing web page. There is also a substantial increase in the sale of "traffic redirectors". Redirector is a malicious code that takes the Internet user to an undesirable website. Although simple, the tool is very effective in spreading malware. It can change the original DNS settings to connect to a fake DNS server. This makes an electronic fund transfer (EFT) sites lead to a fake website with similar interface, which novice users cannot detect. According to APWG, attackers find this method very promising as it can redirect any of the user's requests on any occasion. Users in turn can hardly know what is happening as they could themselves be entering the address and not receive an e-mail or Instant Message in return.

The above points clearly show that there is a need of robust Anti-Phishing solution against continuously evolving internet threat. There are several different anti-phishing techniques to fight phishing, but there is no single solution which can guarantee total protection against phishing. However, properly applied technology along with awareness significantly reduces the risk of identity theft. There are large numbers of possible countermeasures to avoid phishing scam. Most of the countermeasures would fall under one of these three categories.

- Governmental policies against online frauds The **Anti-Phishing Act of 2005** in United States to fight phishing and pharming ("A bill to criminalize Internet scams involving fraudulently obtaining personal information, commonly known as phishing." (Anti-phishing Act, 2005)). If a criminal is sentenced under this law they could risk spending upto five years in prison and/or fine of $250,000. Similarly, China Internet Network Information Center (CNNIC) announced the suspension of new overseas .cn domain registrations against online frauds (Symantec Global Intelligence Network, 2010).

- Creating awareness among users by educating and training, PhishGuru (Kumaraguru, 2009) is an anti-phishing education system which teaches users to avoid falling for phishing attacks by sending them simulated phishing emails. The system delivers a training message when the user clicks on the fake phishing URL in the mail. Anti-Phishing Phil (Kumaraguru, 2010) is an interactive anti-phishing training game that trains the users to identify fraudulent and malicious URLs in 10 minutes, if they make any mistake in the game instantly they learn why. It has learning Management System (LMS) which targets further training, based on the performance of the user.

- Technology countermeasures to detect phish at the website level falls into any of the categories: Heuristic approach, blacklist-based methods, White-list based methods,

Hash-based list methods, Hybrid approach, Visual similarity - based approach and approach which analyse response from websites by supplying random credentials.

There are number of anti-phishing inspection methods and heuristic based approach is the most prevalent method used. Heuristic evaluation is done as a systematic inspection of HTML code, content of the website, URL signatures and external sources to identify phish. Machine learning algorithms are usually applied to build classification models over the heuristic results to automate the webpage classification. For example, Ludl et al.(2007) discovered a total of 18 properties based on the page structure of phishing web pages. Zhang et al. (2007) proposed a content-based method using TF-IDF and six other heuristics to detect phish. Pan and ding (2006) proposed a method to inspect the discrepancy between the stated identity of the site and its structural features and HTTP transactions. Khonji et al. (2011) study aims to evaluate the legitimacy of the webpage by lexically analyzing URL tokens of the page. These Heuristic approaches can detect attacks as soon as they are launched. But, heuristic approaches suffer from false positives, incorrectly labeling a legitimate site as phishing.

The Blacklist approach maintains a list of known phishing sites to check currently visiting websites against the list. This Blacklist is usually gathered from multiple data sources like spam traps or identified by spam filters, user posts (eg. Phishtank), or verified phish compiled by other parties such as takedown vendors or financial institutions. For example, Prakash et al. (2010) used approximate matching algorithm that divides a URL into multiple components that are matched individually against entries in the blacklist. Zhang et al. (2008) proposed a system where customized blacklists are provided for the individuals who choose to contribute data to a centralized log-sharing infrastructure. This individual blacklist is generated by combining relevance ranking score and the severity score generated for each contributor. But blacklist needs frequent updates from their sources, when the blacklist isn't updated properly, it is impossible for the tools to identify phishing websites. Moreover the exponential growth of list needs great deal of system resources.

The white-list approach maintains a list of all safe web sites and its associated information. Any web sites that do not appear in the list are recognized as potential malicious web sites. The current white-list tools usually use a universal white-list where all legitimate web sites are required to be included in the white-list. Whereas Han et al. (2012) developed an approach to maintain individual white-list which records the well-known legitimate web sites of the user rather than maintaining universal legitimate sites list.

Hybrid approach combines the other mentioned approaches to yield the better result in phishing detection. For example, Xiang and Hong (2009) described a hybrid phish detection method which uses an identity-based detection component and a keyword-retrieval detection component along with the white-list. Similarly Xiang and Hong proposed CANTINA+ (2011) where heuristics are combined with blacklist to detect phishing web pages.

In this paper we have proposed a hybrid anti-phishing approach with three modules to check the legitimacy of the webpage. The first two modules are preliminary filters which helps system

to reduce false positive (FP) and computation by eliminating pages which are preapproved by the user and are without login forms. In third module we have used fifteen pivotal heuristics which checks phishiness of the page by examining its structural and behavioural properties. These heuristic results are inputted as fifteen dimension vector to the machine learning algorithm for the classification. In the experiments sections we evaluated performance of the system in two schemes; In first scheme, we accessed the performance of the system only with the heuristics by disabling the filters which results F1 measure of 98.7%, 98.24% of true positive rate with 1.7% of false positive rate, this clearly shows that there is room for further improvement. In the next scheme, we accessed system performance by enabling both the filters which results significant improvement of 99.65% of true positive rate and very low false positive rate of 0.42% without compromising on the false negatives.

The rest of the paper is organized as follows: Section 2 presents an overview of literature review and related work. Section 3 illustrates the overall system architecture. In Section 4,5,6,7 we explained detailed design and methodology used in preapproved site identifier, Login form finder, and Webpage feature Generator and Phishing classifier modules respectively. Experimental results are discussed in Section 8. Conclusion presented in Section 9.

2. Related Work

In this section we will briefly review the previous works that detect phishing web pages. Table-1 provides brief description of the shortcomings in these works and how it is overcome in our work.

Kang et al. (2007) proposed a global white-list based approach that prevents accesses to explicit phishing sites by the URL similarity check. When user accesses a website, the websites URL and IP pair is passed to the Access Enforcement Facility (AEF) to check if the site is a phishing site. If the URL passed to AEF matches with the entry in the trusted site list then it checks the similarity of IP address, if this also matches then it allows the user to proceed otherwise it determines type of phishing attack using other modules and warns the user. *Han et al.* (2012) developed an Automated Individual White-List (AIWL) approach to protect user's online credentials. In this they have maintained individual white-list which records the well-known legitimate web sites of the user rather than maintaining all legitimate web sites in this world. In this white-list along with URL, AIWL also maintains features of the webpage (like legitimate IPs of the page, paths of the input widgets of the page) where the user input their details. This additional information in the white-list helps AIWL to protect users from different kind of online frauds. AIWL warns the user when the submitted account information does not match with the entry in the white-list. In our system we used private white-list as primary filtering module, the structure of the white list is adopted from *Han et al.* work, and its functions are adopted from Kang et al work.

Chou et al. (2004) developed the browser plug-in SpoofGuard that identify phishing web pages based on series of heuristics. These heuristics are grouped into stateless method and stateful method. The heuristics of stateless method identifies suspiciousness of a web page which is downloaded in the web browser (Url, image, link, password), while the heuristics of stateful

method evaluates the webpage credibility through its recent visits by the user (user's history file) and its heuristics to evaluate the input data. *Fette el al.* (2007) proposed a method for detecting phishing emails by including features specific to phishing. They propose ten different features to identify phishing email. Eight of these features can be extracted from the email itself, while the other two features, the age of linked-to domain names has to be obtained by a WHOIS query at the time the email is received and spam-filter output feature incorporate the class assigned to the email. *Zhang et al.* (2007) proposed a content-based approach CANTINA, based on the TF-IDF (term frequency and inverse document frequency) algorithm to identify top ranking keywords from the page content and meta keywords/description tags. These keywords are searched through a trusted search engine such as Google. Here, a webpage is considered legitimate if the page domain appears in the top N search results. The heuristics used in this method are primarily adopted from *Chou et al.* and *Fette el al.* work. CANTINA+ (2011) is an upgraded version of CANTINA proposed by *Xiang et al.*, where new components are included to achieve better results. Particularly, they have included ten other features along with four of the CANTINA features and one extended feature.

Prevost et al. (2011) developed an anti-phishing toolbar "Phishark". In this research they have analyzed and studied the characteristics of phishing attack and have defined twenty heuristics to detect phishing web pages. These twenty heuristics were then checked for the effectiveness and to determine which of these heuristics would take a major part to identify phishing web pages and legitimate web pages. In this paper we have adopted *Domain name in the path of the URL* and *Country code validation* heuristics from Prevost's work with several modifications along with which we have also added new TLD based heuristic *Presence of Multiple TLDs.*

Pan and Ding (2006) proposed a method where the webpage identities are extracted from selected parts of DOM properties to detect phishing web pages (e.g., page title, Meta description field, copyright related text, etc.) and HTTP transactions (e.g., Server Form Handler). These results are used to analyze anomalies in stated identities using page classifier component. *Xiang et al.* (2009) proposed a hybrid phishing detection approach that detects phishing web pages by discovering the inconsistency between a webpage's true identity and its claimed identity using Information Retrieval (IR) and Information Extraction (IE) techniques. Both these techniques manipulate the DOM after the webpage has been rendered in web browser to get around intended obfuscations.

He et al. (2011) adopted technique used by *Pan and Zhang* with combination of search engine results to determine whether a webpage is a phishing page or a legitimate page. The basic idea is that every website claims a certain identity, and its activities correspond to the identity. If a website claims a bogus identity then it would be abnormal, compared to a legitimate site. These deviations from the prescribed identity weigh the legitimacy of the given webpage. In our system we have used features from *Mingxing He's* paper to extract identity related information from DOM of a downloaded webpage. Abu-Nimeh et al. (2007) presented a study that compares the predictive accuracy of several machine learning methods for predicting phishing emails.

Fu et al. (2006) proposed an approach which uses Earth Mover's Distance (EMD) to measure Web page visual similarity. In this approach they first convert the involved Web pages into low resolution images and then use colour and coordinate features to represent the image signatures. EMD is used to calculate the signature distances of the images of the Web pages. They used trained EMD threshold vector for classifying a Web page as a phishing or legitimate. Medvet et al. (2008) proposed an approach which identifies phishing web pages, by considering text pieces and their style, images embedded in the page and the overall visual appearance features of the web page. Chen et al. (2009) present an image based anti-phishing system, which is built on discriminative key point features in Web pages. Their invariant content descriptor and the Contrast Context Histogram (CCH) compute the similarity degree between suspicious and legitimate pages. Chen et al. (2010) also proposed an approach which uses Gestalt theory for detecting visual similarity between two Web pages. They used the concept of supersignals to treat the web pages as indivisible unites; these indivisible supersignals are compared using algorithmic complexity theory.

Joshi et al. (2008) develop the PhishGuard tool that identifies phishing websites by submitting actual credentials after the bogus credentials during the login process of a website. They also proposed architecture for analysing the responses from server against the submission of all those credentials to determine if the website is legitimate or phished one. Chuan Yue and Haining Wang (2010) designed a component BogusBiter that submit a large number of bogus credentials along with the actual credential of users to nullify the attack. A similar approach has been applied by Joshi et al. but BogusBiter is triggered only when a login page is classified as a phishing page by a browser's built-in detection component.

Wenyin et al. (2010) propose a SLN (Semantic Link Network) based approach to identify the phishing website and its target. Here, SLN is composed of semantic nodes and semantic links, where semantic node can be considered as a page and a semantic link is a relationship which connects two nodes. This method concludes a suspected website as a phish when it targets at other associated web pages in all steps of reasoning on the SLN. *Wenyin et al.* (2011) has also developed an approach to detect phishing target based on webpage's strongest parasitic relationship to its community.

Hossain Shahriar and Mohammad Zulkernine (2011) proposed a model to test a suspected phishing websites based on trustworthiness testing approach. In a trustworthiness testing, they check behaviour (response) of websites matches with known behaviour of a phishing or legitimate website to decide whether a website is phishing or legitimate. The model is explained using the notion of Finite State Machine (FSM) to describe the website's behaviour. Based on the state of FSM they developed two heuristics to assist a testing process.

This paper makes the following research contributions : 1. We have proposed three novel heuristics and five modified heuristics (i.e., Heuristics adopted from other research papers are fine tuned and its weakness eliminated after performing lot of experiments) that identifies the inherent characteristics of phishing attack from DOM of the webpage and external information repositories in internet. 2. Our private white-list of preliminary filter module reduces the false

positive of feature based system when in association with login from filter module. Also, DNS references used in it helps system to protect from pharming attack

Work	Shortcomings	Our work
Han et al. (2012)	• New login problem: when a user submits his account details to a website for the first time, this system warns the user, although the current website is a legitimate. This is mainly because the information of the website is not contained in the white-list. • Does not protect against pharming attack.	• The new login problem is completely eliminated in our system; this is primarily by the heuristics used in the feature generator modules followed by the pre-filters module. • Remote DNS lookups prevents from pharming attack.
Xiang et al.(2011)	• Significant problem with the performance, since the system relies on searching through two search engines to identify legitimate domains by consecutively querying search engines which negatively impacts the running time of the system. • Global Black-list • Does not attempt to find phishing website which is hosted on the compromised domains. • Fails to identify legitimate sites use IP addresses.	• Our systems prediction is not solely based on a domain name or search engine results and there by does not impact the running time of the system.
He et al. (2011)	• The heuristics in this system are based on result of TF-IDF algorithm, if it extracts wrong keywords then these heuristics results are unreliable, and this significantly affects the systems predication. • Fails to detect form based anomalies, multipage phishing attack and xss.	• The systems prediction does not largely depend on any heuristic.
Zhang et al. (2007)	• The systems prediction is largely depending on TF-IDF algorithm used in keyword extraction. • Heuristics used does not protect from pharming, multipage and cross site phishing attacks.	• The systems prediction does not largely depend on any heuristic.
Kang et al. (2007)	• Global white-list: practically impossible to maintain all legitimate sites in the internet. • Multiple DNS lookups for pharming detection significantly reduces the response time of the system. • Does not attempt to find phishing website which is hosted on the compromised domains.	• Maintains private white-list which includes URL and IP addresses of the webpage. • Heuristics defined in the "*Evaluations based on FORM element"* section attempts to detect phishing websites hosted on compromised domains.
Chou et al. (2004)	• Records user's private information. • Does not detect phishing site hosted in compromised domain. • Does not protect against pharming attack. • Fails to detect multiple-page phishes.	• Detects multi-page phishing attack (Heuristics- 7,12,13 and 15) • Our method does not record any personal data of user.

Pan and Ding (2004)	• Heuristics used does not protect from pharming, multipage phishing attacks.	• Our method protects from pharming, multipage phishing attacks.

Table-1. Summary of related work in comparison with our work

3. System Architecture

Figure-1 shows the overall system architecture. The main aim of this system is to identify and warn the users whether the opened web page is a phish or not. Before we start applying our heuristics on the downloaded webpage, two preliminary knock-out modules are put into the operation. The first module checks whether the site is in the white-list which is maintained by the user whereas the second module checks if the given webpage gets any sensitive information from user as part of the login process. If the webpage is not in the preapproved site list and still has a provision to get sensitive information from users, then the page will be pushed into feature generator module, where we have defined heuristics to extract 15 features from source of the webpage and external information repositories in the internet. Based on these heuristics results, the feature vector will be generated and passed to trained phishing classifier to predict the webpage class (phish or legitimate).

The feature generator module in this system is defined with 15 novel heuristics to extract feature values from the webpage as well as from the external sources. These heuristics are clustered into 6 groups based on the functions performed. Heuristics belonging to these groups are designed to execute in parallel to check the existence of phishing characteristics in the webpage. Finally heuristics results are grouped globally to generate feature vector that is passed on to the phishing classifier for its class identification. In the training phase, machine learning algorithm was trained using the feature values obtained from every entry in the training dataset by applying same heuristics used in the classification phase.

4. Preapproved site identifier

It is a self constructing private white-list containing legitimate websites approved by the user. Each entry in this list maintains URL of the site and corresponding IP addresses. Before adding IP address into white-list which is received from local DNS, it will be cross checked with remote DNS lookups (Google Public DNS, Dnsadvantag and Norton free dns) to ensure its legitimacy. This helps to prevent the list from pharming attack. The URL is used a pointer to organize and retrieve the list elements.

When user accesses a website, the URL and corresponding IP address is passed on to preapproved site identifier to check whether the (URL, IP) pair is in the white-list. If it matches with the entry in the list the website is believed to be a legitimate one, otherwise it means that the current website is unknown to the user. Then the URL will be forwarded down the line for further verification.

This preliminary screening module reduces unnecessary computation of heuristics defined in the phishing detection module and improves response time of the system when a requested URI is in the list.

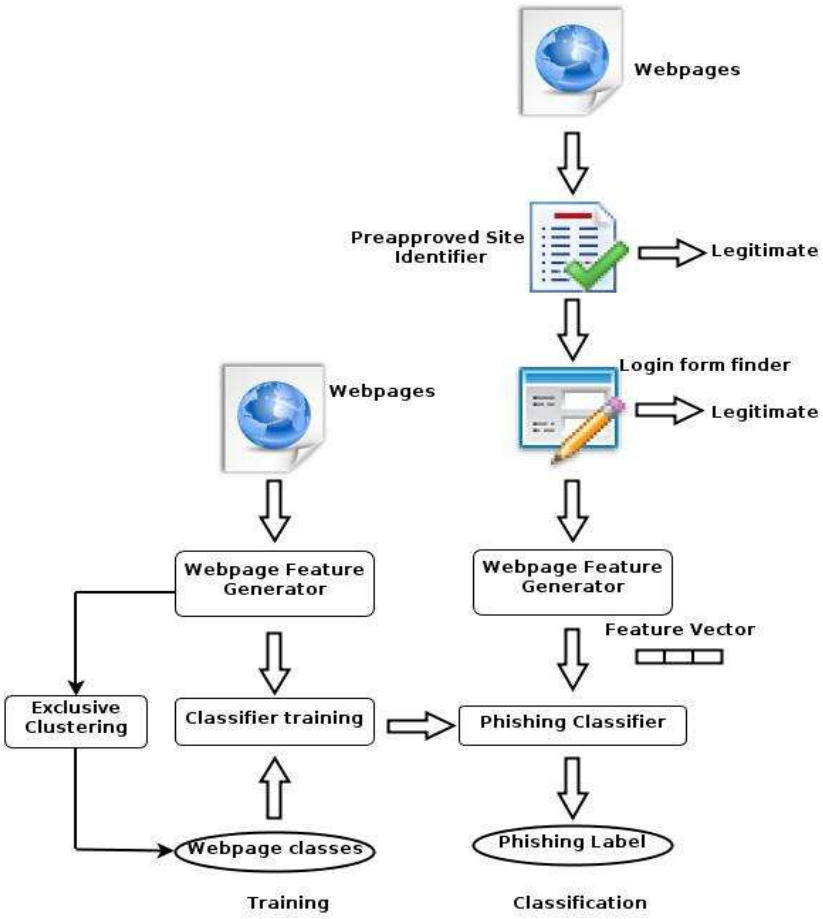

Figure-1. System Architecture

5. Login form finder

Websites at the present time uses registration and login systems that allow only customers to further navigate their network securely. Phishers usually try to steal the login details of users through a fake login forms. In this section, we check if the page has atleast one login form and then proceed with the next step of phishing detection technique. If there are no login forms the detection is not required as the user does not have a way to enter their secret information. This initial screening would prevent useless phishing detection in ordinary pages which do not have login forms. In Figure-2 we use an algorithm to check the existence of login form which is adopted from CANTINA+ (2011) paper.

Algorithm: Login form detection Input: HTML DOM of the webpage Output: (1: Login form exists, 0: Login form not exist)

1. *Check the HTML DOM for form tags and input tags, if present then*
   a. *Search for the search keyword in the form f and its scope to check whether the form is a search form. If the search keyword exists then continue the step 1.a for the next from, otherwise continue with next step.*
   b. *Search for the login keywords in the form f and its scope to ensure that it is a login form. If yes, it returns 1 otherwise it proceeds with the next step.*
   c. *Traverse the DOM tree upto 2 levels to its ancestor node n in the form of f, and search for login keywords under the sub tree rooted at n. If any keywords are found then it returns 1 otherwise it proceeds with next step.*
   d. *If only images exist in the form f and its scope, then it returns 1(phishers use images instead of text in webpage to escape being detected from anti-phishing tools), otherwise it continues with the step 1.a, for the next form. If no more forms are found then it returns 0.*
2. *If the HTML DOM has only the input tags then*
   a. *Search for the login keywords in the entire DOM tree, if yes it returns 1otherwise it proceeds with the next step.*
   b. *If only the images exist in the entire DOM tree then it returns 1 otherwise it returns 0.*
3. *If the HTML DOM has no form tags and input tags then it returns 0.*

Figure-2. Login form detection Algorithm

In this algorithm we have used 38 login keywords (e.g., members log in, password, e-mail, etc.,) and a search keyword which are gathered from our training corpus. These keywords reveal the types of the form we are actually facing.

6. Webpage Feature Generator

In this phase, initially we extract webpage identity from its hyperlinks and content of the page which is then used by heuristics of this module. The heuristics defined in this module are clustered into six groups based on its working nature and its demand, to efficiently check the phishing characteristics of the webpage. Here, a single thread of control splits into multiple threads of control where each will be executed in parallel, thus allowing heuristics in each group to be executed at the same time as shown in Figure-3, It uses shared memory to communicate with heuristics in the other groups. Heuristics defined in the each group checks features in a given webpage and based on its results it creates local feature vector (LFV) LFV_i=<H1,H2..Hn>. These LFVs are further forwarded to synchronized feature vector merger module (SFVM). This SFVM provides synchronization between parallel activities to generate feature vector (FV) by merging the distinct LFVs forwarded to it FVp=<LFV1+LFV2+..+LFVn>, which then serves as an input to a trained *Page Classifier* component to determine whether a webpage is legitimate or phished.

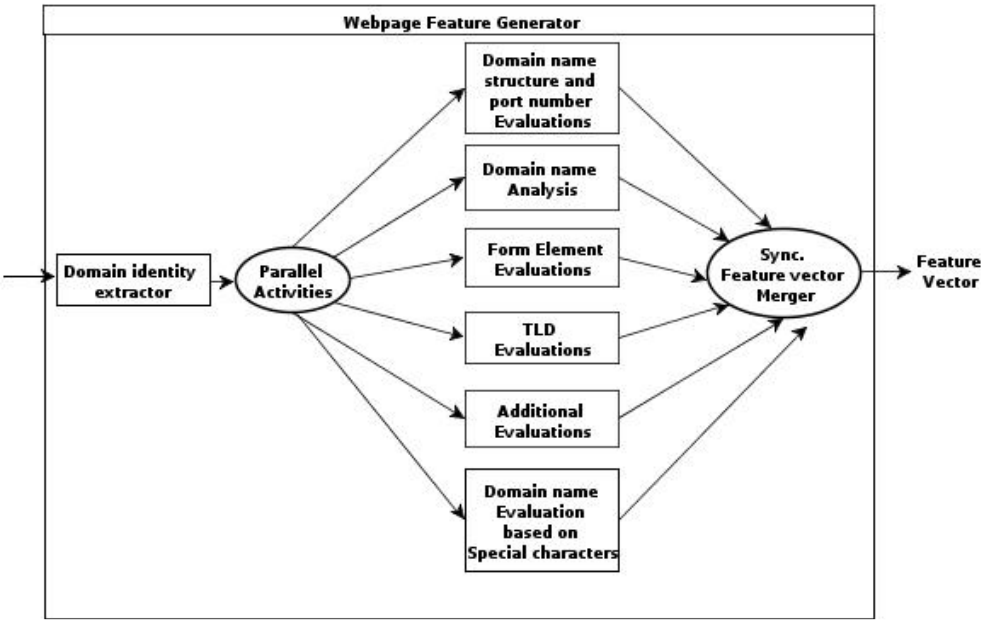

Figure-3. Webpage Feature Generator

6.1 Identity extraction

URL Identity (I_U) – URL identity of a webpage is determined by analyzing its hyperlink structure. In legitimate websites most of the links point to its own domain or associated domain, but in the phishing pages most of the links point to foreign domain to imitate the behavior of a legitimate page. Here, for the URL identity extraction, we have considered only "href" and "src" attributes of the anchor links, particularly <a>, <area>, <link>, <img> and <script> tags from the DOM tree of the webpage. For each anchor URL, we extract base domain part from the URL, and then we calculate the number of times each base domain appears. The base domain which has the highest occurrences will be an URL identity of the webpage. The identity extracted will be in an URL form.

Keyword identity set (I_K) – Keywords are important words in the webpage associated with a product or service. These keywords determine web identity of a page. Keyword identity set is usually extracted by considering following DOM objects of a web page namely title tag, meta tag (Meta description tag, Meta Keywords tag), Alt attribute of tags, Title attribute of tags, Body tag.

We apply Term frequency - Inverse Document Frequency (TF-IDF) method (2011) to extract the keyword set from document, which is collected from various portions of a webpage P. This Keyword identity set generation is adopted from CANTINA (2007).

6.2 Evaluations based on FORM element

The heuristics discussed in the section are also used to detect phishing web pages hosted in the compromised hosts.

Heuristic 1- Login-Form identity: The most common technique used in the phishing attack is to trick users to revel their credentials through login forms on the fake website. These forms

are generally displayed in the same way as that used on the genuine website. In this heuristic we developed an algorithm to check genuineness of the Login forms. For login forms in the legitimate websites, there is a relationship between value of its action attribute to webpage content and hyperlinks of the page. In this algorithm we have used two steps to verify the identity of the login forms. In the first step, domain name which is used in the action field of the form will be extracted and the form's identity will be checked from extracted domain name against URL identity *(I_U)* and keyword identity set *(I_K)* of the webpage and this step will be repeated for every login form in the web page. In the second step the heuristic result (-1,0,1) will be determined based on the identities of all the login forms in the webpage. The corresponding algorithm is shown in Figure-4.

Algorithm: Form identity

Input: Source of the webpage, I_K and I_U
Output: (1: suspicious, 0: neutral, -1: legitimate).

1. *For each form, extract value of the action attribute (U)*
    a. *If U is in relative URL form then convert it into absolute URL form*
    b. *Extract base domain (D) from U*
    c. *If D is a base domain of the webpage and URL identity (I_U) is a local domain then R_i=-1. (R_i is the result of i^{th} form)*
    d. *If D is base domain of the webpage and I_U is a foreign domain then*
        i. *If any of the keywords in I_K is part of the base domain D then R_i =0.*
        ii. *Otherwise, R_i =1.*
    e. *If D is not a base domain of the webpage and I_U is a foreign domain then R_i =1.*
    f. *If D is not a base domain of the webpage and I_U is a local domain then R_i =1.*
2. *If the action attribute does not exist or its value is null then R_i=0.*
3. *If any of the form's test result is 1 then set H1=1, otherwise if any of the form's result is 0 then assign H1=0, otherwise H1= -1 as a heuristic result.*

Figure-4. Form Identity Algorithm

Heuristic 2- Age of forms: The age is calculated for domain name in the URL, where the URL is the value of the form's action attribute extracted from DOM of the web page. If the domain name used in the form's action field is similar to base domain of the webpage then the website's age is taken as the form's age, otherwise age will be calculated from WHOIS search. If a webpage has more than one login forms then the age will be calculated individually for each form. This heuristic returns 1, if any of the form's age is less than a month or if the action field does not exist or is null. It returns 0 if the age is more than a month for all forms. The algorithm for this is shown in Figure-5.

Algorithm: Age of forms

Input: Source of the webpage
Output: (1: suspicious, 0:neutral, -1: legitimate)

1. *For each form, extract value of the action attribute (U)*
   i. *If U is in relative URL form then convert it into absolute URL form*
   ii. *Extract base domain (D) from U*
   iii. *Extract created date of D from WHOIS record and calculate its age.*
2. *If the action attribute does not exist or its value is null for any of the form then H2=0.*
3. *If any of the form's age is less than a month then H2=1, otherwise if all forms age is over a month then H2=-1, otherwise H2=0.*

Figure-5. Age of forms Algorithm

6.3 Domain Name Structure and Port Number Evaluations

Heuristic 3 - Irreversible IP address: This heuristics checks whether the URL address of a webpage is a permanent IP address. Many phishing web pages can only be accessed by IP address URL instead of a domain name. This is because of the low cost associated with phishing. Secondly, some phishing attacks are hosted from compromised computers, these machines may not have DNS entries, and the simplest way to refer them is by their IP address. But then, the legitimate websites are most of the times accessed by a domain name instead of IP address.

Heuristic 4 – Domain name in the path of the URL: Some phishing URLs add domain name of the legitimate website within the path segment of the URL to trick the user into trusting that they are communicating with the genuine site. This feature checks the presence of domain name structure in the path segment of such URLs. Here, we used precompiled list consisting of 5325 entries of top-level domains (e.g., net, in), second-level domains (e.g., co.in, ac.in) and all combinations of it (e.g., com.br, ac.be) for checking the existence of domain name in the URL path. The feature result is success, if any of the entry in our precompiled list matches in the path segment of the URL.

We have not used pattern matching as used in the CANTINA+ because it has a chance of giving false positives. According to this method "*three constraints must be met for a dot-separated string to be eligible for an embedded domain. First, at least three segments must exist. Second, each segment must have two or more characters. Third, each segment is composed of letters, numbers and underscores only*" (2011). This may not hold true in every case.

For example the following URL *http://www.jstor.org/about/**field.list.html*** is identified as suspicious page by CANTINA+ though it is a legitimate page.

Heuristic 5 – Invalid port number: In this we check the port number part of the domain name with stated protocol part of the URL. This heuristics returns 0 either if the protocol matches with port number or port number part is not used in the URL and returns 1 when port number

does not match with stated protocol. For example, let us consider the phishing URL of amazon.com:

http://61.128.197.81:5800/signin.htm?_encoding=UTF8&...

Here, the heuristics result is 1 because the port number in URL is 5800, but stated protocol is "http", the default port number is either 80 or 8080.

6.4 Domain name analyses

Heuristic 6- Domain name Credibility: Domain name credibility feature evaluates the legitimacy of the target website using the rank given by the page rank system. The rank denotes importance of a particular page. Note that this feature checks the credibility of a hosting domain instead of target website. Since target website may be a new web site that has not been documented in the PageRank system; its hosting domain, however, is more likely to have a longer record.

- If the rank of the hosting domain is greater than or equal to threshold value then the domain hosting web site is credible and consequently determines that web site is also legitimate (H6=-1).
- If the rank is less than the threshold value then the trustworthiness of the web site is obscure (H6=0).
- If the rank is zero or not available then the value determines that web site is a phish (H6=1).

In the implementation we used Google's PageRank (PR) system (Sobek, 2003) to obtain the PageRank score of a site or domain. The PR score for any web site is a whole number between 0 and 10. The most popular websites have a PageRank of 10. The least have a PageRank of 0. In our experiments we set threshold value as 5 for the evaluation of domain hosting web site. The threshold value used in this feature is based on related work done by Kaigui Bian (2009).

For getting page rank score from Google PR system we used simple GET request to *toolbarqueries.google.com*. Along with it, we have appended the domain name and its hash code in the query string.

Heuristic 7- Age of Domain Name: We use a WHOIS search to implement this heuristic. This heuristic calculates the age of domain name from website's creation date in terms of number of months. For the legitimate sites the measured months will be always greater than 1, but for phishing sites this value will be 0. Since most of the phishing sites have domains that are registered which exist only for few hours to few days (According to the Anti-Phishing working group's report average uptime of phishing website is 46 hours and 3 minutes (2012)). This heuristic does not account for phishing pages hosted either on compromised web server or on a hosting provider. This heuristic returns 0 when a website's age greater than a month, otherwise it returns 1(H7=1).

Heuristic 8: Domain name identity: Most of the website's domain name will be related to its content. This characteristic of the URL clearly shows that there exist a relationship between a base domain of its webpage and its keyword identity set. For example, for the co-operative

bank website *https://personal.co-operativebank.co.uk/CBIBSWeb/start.do*, the base domain is *personal.co-operativebank.co.uk*. The keyword identity set (I_K) of co-operativebank's Personal banking page is {co-operative, internet, digit, bank, visa}, and one of them is a part of the base domain URI. Along with this, most of the hyperlinks in this page point to its own domain *www.co-operativebank.co.uk*.

On the contrary, phishing websites usually claim identity that is different from the webpage's content and hyperlink structure. Base domain of a phishing page may not be a part of its keyword identity set. For example, we take co-operativebank phishing page URI http://phantomvison.com/js/CBIBSWeb. start.html, the base domain is phantomvison.com. The keyword identity set of this page is {co-operative, internet, digit, bank, visa}, none of these keywords are part of the base domain URI. The majority of hyperlinks in this page point to foreign domain www.co-operativebank.co.uk.

If the URL identity of webpage is similar to the base domain of a URI, and also if one of the keywords from the keyword identity set is a part of the base domain, then the heuristic result would be 0 (H8=0).otherwise the result is 1(H8=1).

6.6 TLD Evaluations

Every URL will have a Top Level Domain (TLD) in its domain name. The TLD is the part of the URL that describes the information like kind of the site (eg., educational, personal, organizational, bank). In this section we verify the legitimacy of the webpage based on country code specified in the URL and existence of multiple TLDs in a domain name.

Heuristic 9 – Presence of Multiple TLDs: In this we check the occurrence of multiple top-level domains in the domain name part of the URL. This heuristics result is 1 when more than one generic top-level domains (gTLDs) or more than one country code top-level domains (ccTLDs) exists in the domain name followed by any special character, otherwise the result is 0 (H9=0). For example, let us consider phishing URL of paypal.com account holder's login page:

http://paypal.com.gpsoptions.com.au/uk/cgi-bin/

Here, .com (gTLD) occurs two times in the domain name part of the URL.

Heuristic 10 – Country code validation: This heuristic verifies the country code specified in the domain name against its hosting country information. Here, we extract the TLD of a given URL and compare it with the hosting country code of the website, if both matches then the site is considered as legitimate. If any website with the country code is hosted from another country, it is considered to be a suspicious as it is illegal to do so. According to McGrath et al. study (2008) most of the phishing domains were not always hosted in the country they were registered in. In addition to that, RSA's fraud report says, "*In April 2012, 55 percent of phishing attacks were hosted in the U.S., followed by Brazil which hosted 13 percent, in contrast over 90 percent of the entire volume of phishing attack target at the UK, Canada and the U.S. in the same month*" (RSA Anti-Fraud Command Center, 2012). This clearly indicates that several phishing attacks were hosted from only some countries. We have used country code validation algorithm (Figure-6) to verify ccTLD in the URL.

In this heuristics, we have used MaxMind's free IP geolocation database - *GeoLite* (MaxMind, 2012), to identify hosting country code of a website from its IP address.

Algorithm: *Country code validation*

Input: Domain name of the website
Output: (1: suspicious, 0: neutral, -1: legitimate)

1. *Find IP address of a domain name using DNS lookup*
2. *Find geographic location of the IP address by querying GeoLite's IP to Country database*
3. *Convert geographic location to hosting country code*
4. *Extract ccTLD from domain name, If the domain name does not have ccTLD then H10=0.*
5. *If hosting country code does not match with ccTLD then H10=1, otherwise H10=1.*

Figure-6. Country code validation Algorithm

6.7 Additional URL Evaluations

Heuristic 11- Phishing Keywords in URL: Here, keywords are words which occur in a phishing URL more often than the words that we would expect to occur by chance alone. In this function we check their presence in the path segment of the URL. When there are many phishing keywords in a URL, the webpage could be less credible. We have selected these keywords from previous work done by Garera et al. (2007), a study on structure of URLs employed in various phishing attacks.

Feature 12: SSL/TLS certificate: There are mainly two ways adopted by the attackers to create a protected phishing sites. One is to buy a forged SSL or TLS certificate from a certification authority (false-certificate attack), but this has a huge drawback of being jammed with a particular domain name. The second method is the attackers may use a self-signed certificate.

This heuristic checks whether a page has any input tags or login forms if then it verifies protocol part of the URL to be HTTPS. If the page is a secured page then it verifies the genuineness of the certificate by checking the certificate was issued by a trusted party (usually a trusted root CA), Distinguished Name (DN) in the certificate is matched against the page address which has the certificate, and one of the claimed identity (I_K) appear in the certificate attached, if all these are factual then the webpage would be considered as legitimate, otherwise the legitimacy of the webpage can be determined in combination with other heuristic results. If the DN in the certificate does not match with page address then the webpage would be considered as suspicious. The corresponding algorithm is shown in Figure-7.

***Algorithm**: Certificate validation*

Input: URL
Output: label (-1: legitimate, 0: neutral, 1: suspicious)

1. If the web site is SSL or TLS protected then
  a. Check whether the certificate is valid and issued by a trusted vendor, if so
    i. Verify if the Distinguished Name(DN) is the same as the domain name of webpage
      (i) If at least one member of I_K is part of DN then it returns -1
      (ii) else it returns 0
    ii. If DN and domain name of the website are different then it returns1
  b. Similarly if the certificate is not valid or issued by a non-trusted third party it returns 0.
2. If no SSL or TLS certificate present in the webpage that has a login form then it returns1.

Figure-7. Certificate validation Algorithm

6.2 Domain name Evaluation based on Special characters

Here we evaluate a URL by running a series of heuristic functions like counting no of dots, double slash (//) and presence of (@) sign.

Heuristic 13 - The '@ 'character in the Domain name: In this heuristic we check the presence of "at" characters (@) in the domain name part of the URL. The string left to '@' character is treated as "user info" of the URL, and the string to the right is treated as the actual domain for retrieving a page. *http(s)://username:password@domain-name/* this syntax is rarely used by genuine websites, but some malicious users might use this URL syntax to create a domain name or hyper link that opens up an illegitimate webpage which appears to be a genuine website. The heuristics result is 1(H13=1) if it exists, otherwise the result is 0 (H13=0).

Heuristic 14 – Dots in URL: In this test we count the number of dot characters (.) present in the domain name. Usually legitimate URLs will not have five or more dots (Zhang, 2008; Fette, 2007) but the phishing URLs may have many dots to confuse the users to believe that they are actually communicating with legitimate page. So if any URL is with more than five dots the heuristics result is success (H14=1) and the webpage is considered suspicious, otherwise the heuristics result is failure (H14=0).

Heuristic 15 – Double slash in URL path: This heuristic checks the occurrence of double slashes (//) in the path segment of the URL (H15=1). Usually phishing URLs have double slash in the path segment to mention legitimate domain names as part of it. This is to give an impression that it is legitimate. For example, let us consider phishing URL of *battle.net* member's login page:

http://blizzard-free-login-diablo3.vicp.net/login/zh/?ref=https://us.battle.net/account/...

Here double slash is used in the path segment to include domain name of legitimate URL as part of it.

7. Phishing Classifier

In order to classify the different features of a web page, we apply a method that analyzes data and recognizes patterns known as SVM (Support Vector Machines) invented by Vapnik in 1995.The SVM classifier takes a set of input data and predicts, which of two possible classes would form the input. This makes the SVM a non-probabilistic binary linear classifier. SVM is a Supervised Learning scheme; we need to divide our data into a training set and a testing set. The training set contains the class labels along with the known feature values of each and every train entity given. Based on the training set a model file will be generated which is a consolidated repository of the trained data and the rule which would be used to classify the test data. In testing phase the data are compared with the trained SVM model and a decision is made whether the page is legitimate or not.

The SVM classifier input used in this method is a 15 dimension feature vector FVp induced from the webpage feature generator module, representing a web page's 15 structural and behavioral characteristics. For SVM implementation we use LIBSVM (2012), a library for support vector machines classification and regression, developed by National Taiwan University.

7.1 SVM Training process and Class Prediction

Generate the training file: The training file is generated by giving URLs in the training dataset as input for the anti-phishing system, which will find the feature values and generates feature vector (V_P) for every webpage and store in the training file. The training file contains the class labels along with the feature values of each and every train entity given.

Syntax of the training files entry:

Class_label Feature1: Feature1_value Feature2: Feature2_value…

Example:

-1 1:3 2:0 3:1 4:1 5:1 6:2 7:2 8:1 9:1 10:0 11:1..

+1 1:1 2:0 3:1 4:1 5:1 6:0 7:0 8:2 9:1 10:1 11:1..

Generating the model file: The model file is a consolidated repository of the trained data and the rules which is used to classify the test data. Along with training data, the type of SVM and kernel type used for classification are also given as input for generating the model file. Here, we have used C_SVC (C-Support Vector Classification) as SVM type and RBF (radial basis function) as kernel type. The +1 and -1 in the first column of the train file denote the classes (Cristianini, 2000). The rest denote the feature values.

Generating the testing file: This is generated by giving the test dataset to the Anti-Phishing system, which will analyse the webpage as per the proposed models and produces the feature values. The test file is also of the same format as the input training file. But the only difference between the training and testing file is that the test file will not have a class label.

The classification: In this phase, the feature vectors of the test web pages are given as input to the model file and the class of each and every feature vector is predicted using the rule framed during the training period. This stage will produce an output which contains the class label of the corresponding feature vectors.

The output class is read and according to the class that is present, the webpage is assigned the label.

Class +1 – Phishing webpage

Class -1 – Legitimate webpage

8. Experiments

8.1 Metrics used in evaluation

In our experiment, primarily we used two metrics to evaluate the performance of the system they are, true positive rate (TP) and false positive rate (FP) respectively. Along with this we have also used standard measures, such as precision and F1-measure. F1-measure combines the precision and recall to measure test's accuracy.

True Positive Rate (TPR) is calculated by dividing the number of web pages retrieved as a phish by the total number of phishing website that should have been retrieved. TPR is computed using equation 5.1.

$$\text{TPR} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{(\text{TP}+\text{FN})} \qquad \text{(Eq. 5.1)}$$

Here, TP is the number of correctly classified phishing pages; P is the number of phishing pages which is equivalent to sum of correctly classified phishes (TP) and missed phishes (FN).

False Positive Rate (FPR) measures the percentage of legitimate sites wrongly classified as phishing. FPR is computed using equation 5.2.

$$\text{FPR} = \frac{FP}{L} = \frac{FP}{(FP+TN)} \qquad \text{(Eq. 5.2)}$$

Where FP is the number of legitimate web pages wrongly classified as phishing pages and L is the total number of legitimate pages.

Precision (PR) is calculated by the number of actual phishing web pages a classifier retrieves divided by total number of web pages retrieved as phishing. PR is computed using equation 5.3.

$$\text{PR} = \frac{TP}{(TP+FP)} \qquad \text{(Eq. 5.3)}$$

F1- measure is a harmonic mean of Precision and True Positive Rate as shown in equation 5.4.

$$\text{F1} = 2 * \frac{PR * TPR}{(PR+\text{TP}R)} \qquad \text{(Eq. 5.4)}$$

These metrics assess the overall performance of the system and its weaknesses, which helps us to tune the system.

8.2 Description of data

We have collected a real world dataset of 2464 live phishing and legitimate websites over a period of 90 days from March 2012 to June 2012. Specifically, our dataset consists of 700 legitimate pages and 1764 unique phishing pages.

Legitimate pages in our dataset are obtained from three sources as shown in Table-2. We only included the English language websites in our dataset to eliminate the impact of language heterogeneity on our webpage-ID generation. Our legitimate collection mainly focuses on the popular sites and common targeted sites published on the sources.

Source	Sites	Link
Google's top 1000 most-visited sites	400	http://www.google.com/adplanner/static/top1000/
Alexa's Top sites	150	http://www.alexa.com/topsites
Netcraft's Most Visited Sites	100	http://toolbar.netcraft.com/stats/topsites/
Millersmile's of top targeted sites	50	http://www.millersmiles.co.uk

Table-2. Legitimate data source

Each entry in our phishing dataset is unique which are downloaded from two sources as shown in Table-3. Phishing web sites are very transient and short-lived remains active for only a few hours to an average of 3 days before they are shut down. Some phishing URL's are moved through different hosts, with multiple take down and reactivations. Because of these factors, we downloaded the pages up to level one when they were alive and conducted our experiments in an offline mode.

Source	Sites	Link
Phishtank's open database	1264	http://www.phishtank.com/developer_info.php
Reasonable-Phishing Webpages List	500	http://antiphishing.reasonables.com/BlackList.aspx

Table-3. Phishing data source

The training data set for the SVM classifier was randomly drawn from our legitimate and phishing databases.

8.3 Experimental criteria

We conducted two experiments to evaluate the performance of our system. Both the experiments were provided with the same dataset during the time of testing process. In the first scheme, we assessed performance of the system only with the heuristics in *Webpage Feature Generator component* by disabling *Preapproved site identifier* and *Login form finder* components, and we also examined the correctness of heuristics in this component, to determine the best way to reduce False Positives (FP) without compromising on system's false negative rate (FN). In the second scheme, we evaluated the performance of overall system and

compared performance results to other Phishing detection methods. We used *precision* and *F1-Measure* metrics to evaluate the performance of the system in both the experiments.

8.4 Experimental results

8.4.1 Evaluation of login form finder

In this experiment we assessed how effective our login form finder filters web pages with login forms. As shown in the Table-4, this module correctly detected 98.05% of web sites with login forms. The results show that, this algorithm detects all login forms which is used in the legitimate web pages and detects only 97.27% of login forms which is used in the phishing web pages. This is because the key words in the phishing pages are not in our search list and phishers used no login forms in their pages.

	Phishing	**Legitimate**	**Total**
No. of pages	1764	700	2464
Successful Detections	1716	700	2416

Table-4. Performance of Login form finder

8.4.2 Evaluation of the system based on scheme-1

In the first experiment we assessed how effective our adoption of heuristics in *Webpage false negative Feature Generator component* would be in phishing web page identification. This was done primarily by disabling *preapproved site identifier* and *Login form finder* components. The classification results are shown in the Figure-8. The false positive rate is 1.71% with the false negative rate of 1.75%. Hereby the results prove to be comparatively poorer than the results obtained through other Anti-Phishing techniques. The results of scheme-1 are as shown in Table-5. Here, we see that the accuracy is also relatively low due to the false positives. Thereby this method certainly has room for further improvement.

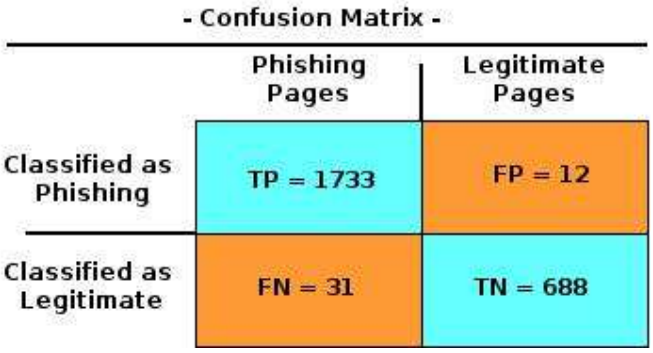

Figure-8. Test results when filters are disable

8.4.3 Evaluation of the system based on scheme-2

In second experiment we assessed the performance of the entire system by enabling *preapproved site identifier* and *Login form finder* component. We tested the proposed system

with same testing set used in the first experiment. The experiment results are shown in Figure-9. False positive rate is 0.42% and false negative rate of 0.34%. This statistics clearly shows that heuristics combined with pre filtering mechanism detected more phishing pages than system without it. Also, the results indicate that this system detects phishes with less false positives without compromising on the false negatives. Hereby, the results prove to be comparatively better than the results obtained through other Anti-Phishing methods as shown in Table-10.

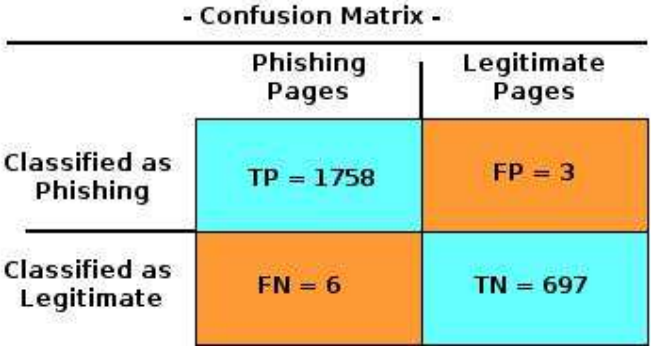

Figure-9. Test results when filters are enabled

Pre approved site identifier	Login Filtering	TPR (%)	FPR (%)	PR (%)	F1 (%)
Disabled	Disabled	98.24	1.71	99.25	98.74242
Enabled	Enabled	99.65	0.42	99.82	99.73493

Table-5. Performance of our AntiPhishing system under scheme-1 and scheme-2.

The test results of PhishTackle are compared to those of *An efficient phishing web page detector* (AEPWD), *CANTINA+, Discovering phishing target based on semantic link network (SLN)* anti-phishing methods. The performance results are shown in Table-6. Our method outperforms any other methods in phishing webpage detection, as it still has a fairly low false positive rate and high accuracy. The testing dataset for each method is different. Note that the results of other methods are collected from the respective papers.

AntiPhishing Method	TPR (%)	FPR (%)	PR	F1
Anomaly	94	14	95.94	94.49
AEPWD	97	4	96.03	96.51
CANTINA+	99.63	0.407	99.75	99.61
SLN	83.4	13.8	85.8	84.58
Our Method	99.65	0.42	99.82	99.73493

Table-6. Comparison of our test results with other AntiPhishing methods

8.4.4 Discussions

Preapproved site identifier is a preliminary filtering module. Whenever the user accesses a site, this module checks the entry in the private white-list which is maintained in the client. When

a domain name and its IP matches with the entry in the list, further checks will be bypassed and the user will be informed that the website currently visited is legitimate. When this preapproved list is polluted or lost by any client side attacks then the dependability of the whole system would be lost. In addition the preapproved list maintained in one machine will not be available in other machine for the same user. If we store the list in the centralised server in the respective user's space with sufficient privileges, then the list can be made available anywhere.

The reliability of the login form finder module is solely based on the keywords which we used to check its presence in the form's scope. If a login form uses login keyword which is not in our keywords list then without further processing the webpage is considered to be legitimate. This problem can be eliminated either by designing a dynamically expandable keywords list, which updates the keywords by itself based on user's browsing behaviour or by maintaining a domain specific login keywords.

The tf-idf algorithm used in *Identity Extraction module* can be assaulted through three methods; in first method of attack, the tf-idf algorithm extracts no identity from webpage. This would be done by representing the page content through images instead of words. A second method is to include additional words or changing words in the phishing page which would result in extraction of incorrect page identity. Also sometime attackers include additional fake references or self references in a page to have a phishing webpage to be indexed by search engines. However, this requires significant effort to be done in practice. A third method would be to include invisible texts (text that is visible to computer but not to users) in phishing web page to extract wrong term as page identity. Thus the phishing webpage can still be detected even if the identity extraction process is attacked, through other heuristics in the *Webpage Feature Generator* component.

Domain credibility feature has limitation in validating phishing pages hosted on compromised domains. For example; The Amazon phishing site hosted on Polish government TLD "*opole.uw.gov.pl*" (Netcraft, 2011) has a PageRank score of 5. Since the score is equal to threshold value this site may be misinterpreted as legitimate page. Also this feature has other limitations in validating legitimate sites whose hosting domains have low PageRank scores. These false positives can be minimized by combining this feature results with other features like Credible In-neighbor Search (CIS) (Bian, 2009).

In domain name identity heuristic, the identity of the webpage's domain name is verified exclusively using keyword identity set (I_K) of the webpage which is extracted from its content. We used tf-idf algorithm to generate keyword identity set (I_K), if this algorithm extracts wrong identity set from page content by means as already discussed then the dependability of this heuristic and all other heuristic works based on this would be lost. Therefore whole system prediction would be based on other heuristics.

In country code validation heuristic we used GeoLite Country database which is one of the freely downloadable geolocation databases updated on first week of every month. The GeoLite database draws primarily from publicly available data and is less accurate approximately 98% on the country level. On the other hand, GeoIP is a commercial IP geolocation database and its

accuracy is over 99%. Because of these reasons the accuracy of this heuristic solely depends on the accuracy of the database used in the backend. Moreover this heuristic evaluates the URLs that have country code in its TLD.

9 Conclusion

In this paper we have presented architecture for an AntiPhishing system with pre-filter mechanism. The main aim of this system is to detect phishing websites accurately with low false positives. This is achieved through rich set of heuristics defined in *Webpage Feature Generator* module which precisely identifies phishing web pages by checking the presence of phishing characteristics in it, which is the core work of this paper. Our system uses, Preapproved Site Identifier to reduce the unnecessary computation of the system to detect legitimate pages which is already visited by user, and the Login Form Finder filters pages without login forms and stops it from further processing, since, login forms are the only means through which phishers lure to reveal users private credentials. These filters reduce false positives of the system without compromising on the cost of false negatives.

As a conclusion, with the help of these heuristics we achieved 98% of TPR and FPR of 1.17% in the absence of pre-filters. The FPR can be even more reduced to 0.42% and TPR increased to 99.6% with the help of pre-filters.

References

[1] A library for support vector machines classification and regression, developed by National Taiwan University. www.csie.ntu.edu.tw/~cjlin/libsvm/

[2] Anti-phishing Act of 2005, http://www.govtrack.us/congress/bills/109/hr1099/text, Visited on June 2012.

[3] Anti-Phishing Working Group, Global Phishing Survey: Trends and Domain Name Use in 2H2011. http://www.antiphishing.org/reports/APWG_GlobalPhishingSurvey_2H2011.pdf, April 2012.

[4] Chen, T.C., Dick, S., and Miller, J. 2010. Detecting visually similar Web pages: Application to phishing detection. ACM Trans. Intern. Tech. 10, 2, Article 5 (May 2010), 38 pages. DOI = 10.1145/1754393.1754394 http://doi.acm.org/10.1145/1754393.1754394

[5] Christian Ludl, Sean Mcallister, Engin Kirda, Christopher Kruegel On the Effectiveness of Techniques to Detect Phishing Sites, In Proc. of 4th international conference on Detection of Intrusions and Malware and Vulnerability Assessment, 2007, pp. 20 - 39.

[6] D. K. McGrath and M. Gupta, Behind phishing: An examination of phisher modi operandi, in Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats. San Francisco, California, USA: USENIX Association Berkely, CA, USA, 2008, Article No. 4.

[7] Dao.T : Term frequency-Inverse document frequency implementation in C#, The Code Project - C# Programming, http://www.codeproject.com/csharp/tfidf.asp, Visited on Nov 2011.

[8] E. Medvet, E. Kirda, and C. Kruegel. Visual-similarity-based phishing detection. In IEEE International Conference on Security and Privcay in Communication Networks, Istanbul, Turkey, September 2008. IEEE Computer Society Press.

[9] Fu, a. Y., wenyin, l., and deng, X. 2006. Detecting phishing Web pages with visual similarity assessment based on earth mover's distance (EMD). IEEE Trans. Depend. Secure Comput. 3, 4, 301-311.

[10] Guang Xiang, Jason I. Hong: A Hybrid Phish Detection Approach by Identity Discovery and Keywords Retrieval, Proceedings of the 18th international conference on World Wide Web, New York, NY, USA, 2009. ACM Press. pp. 571-580.

[11] H.Shahriar, M.Zulkernine: Trustworthiness testing of phishing websites: A behavior model-based approach, Future Generation Computer Systems (2011), doi: 10.1016/j.feture.2011.02.001

[12] Han.W, Ye Cao, Elisa Bertino, Jianming Yong, Using automated individual white-list to protect web digital identities. Expert Systems with Applications (2012), doi:10.1016/j.eswa.2012.02.020

[13] I. Fette, N. Sadeh, A. Tomasic: Learning to detect phishing emails, in Proc. Of the 16th Intl. Conf. on World Wide Web, Banff, Alberta, Canada, May 2007, pp. 649-656.

[14] Jian Zhang, Phillip Porras and Johannes Ullrich, Highly predictive blacklisting, In proc. of the 17th conference on Security symposium, USENIX Association Berkeley, CA, USA, 2008, Pages 107-122.

[15] JungMin Kang and DoHoon Lee, Advanced White List Approach for Preventing Access to Phishing Sites, International Conference on Convergence Information Technology, 2007, pp:491-496.

[16] K.T. Chen, J.-Y. Chen, C.R. Huang, and C.-S. Chen, Fighting phishing with discriminative keypoint features, IEEE Internet Computing, vol. 13, no. 3, pp. 56-63, 2009.
[17] Kaigui Bian, Jung-Min Park, Hsiao.M.S, Belanger. F, Hiller.J: Evaluation of Online Resources in Assisting Phishing Detection, Ninth Annual International Symposium on Applications and the Internet, 20-24 July 2009, pp. 30-36.
[18] Kaigui Bian, Jung-Min Park, Hsiao.M.S, Belanger. F, Hiller.J: Evaluation of Online Resources in Assisting Phishing Detection, Ninth Annual International Symposium on Applications and the Internet, 20-24 July 2009, pp. 30-36.
[19] L Wenyin, G Liu, B Qiu, X Quan, Anti-phishing through phishing target discovery. IEEE J Internet Comput. 16(2):52-61 (2011)
[20] L. Wenyin, N. Fang, X. Quan, B. Qiu, G. Liu, Discovering phishing target based on semantic link network, Future Generation Computer Systems, Vol. 26, No. 3, 2010
[21] Mahmoud Khonji , Youssef Iraqi, Andrew Jones, Lexical URL Analysis for Discriminating Phishing and Legitimate Websites, CEAS '11 September 1-2, 2011, Perth, Western Australia, Australia,pp:109-115.
[22] Markus Sobek : A Survey of Google's PageRank, Available at: http://pr.efactory.de/
[23] MaxMind GeoLite Databases: http://www.maxmind.com/app/geolite, visited on June 2012.
[24] Microsoft, Microsoft Security Intelligence Report, 2010. http://go.microsoft.com/?linkid=9771781, visited on June 2012.
[25] Mingxing He, Shi-Jinn Horng, Pingzhi Fan, Muhammad Khurram Khan, Ray-Shine Run, Jui-Lin Lai, Rong-Jian Chen, Adi Sutanto : An efficient phishing webpage detector. In Expert Systems with Applications: An International Journal, Volume 38 Issue 10, September, 2011, pp. 18-27.
[26] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, J. Mitchell : Client-side defense against web-based identify theft, in Proc. of the 11th Annual Network and Distributed System Security Symposium, NDSS'04, San Diego, CA, February 2004, Vol. 380.
[27] Nello Cristianini and John Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", Cambridge University Press, 2000.
[28] Netcraft : Governments hosted 146 new phishing sites in July, July 2011. http://news.netcraft.com/archives/2011/08/19/governments-hosted-146-new-phishing-sites-in-july.html, visited on Feb 2012.
[29] Pan. Y and Ding. X : Anomaly Based Web Phishing Page Detection, In Proc. of the 22nd Annual Computer Security Applications Conference (ACSAC'06), 2006, pp. 381-392.
[30] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, Minaxi Gupta, PhishNet: Predictive Blacklisting to Detect Phishing Attacks, International Conference on Computer Communications, 14-19 March 2010. pp:1-5.
[31] Ponnurangam Kumaraguru, Justin Cranshaw, Alessandro Acquisti, Lorrie Cranor, Jason Hong, Mary Ann Blair, Theodore Pham, School of Phish: A Real-World Evaluation of Anti-Phishing Training, Symposium on Usable Privacy and Security , July 15-17,2009, Mountain View, CA USA
[32] Ponnurangam Kumaraguru, Steve Sheng, Alessandro Acquisti, Lorrie Faith Cranor, And Jason Hong : Teaching Johnny Not to Fall for Phish, ACM Transactions on Internet Technology, Vol. 10, No. 2, Article 7 (May 2010), 31 pages. DOI 10.1145/1754393.1754396 http://doi.acm.org/10.1145/1754393.1754396

[33] RSA Anti-Fraud Command Center, RSA Monthly Online Fraud Report, May, 2012: http://www.rsa.com/solutions/consumer_authentication/intelreport/11713_Online_Fraud_report_0512.pdf, visited on June 2012.
[34] RSA Anti-Fraud Command Center, RSA Monthly Online Fraud Report. July, 2012: http://www.rsa.com/solutions/consumer_authentication/intelreport/11752_Online_Fraud_report_0712.pdf?M=50f5edbd-cabe-4502-bd6c-eab4c4952bc3
[35] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nai. A comparison of machine learning techniques for phishing detection. In APWG eCrime Researchers Summit (eCRS), Pittsburgh, PA, October 2007.
[36] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nai. A comparison of machine learning techniques for phishing detection. In APWG eCrime Researchers Summit (eCRS), Pittsburgh, PA, October 2007.
[37] S. Garera, N. Provos, M. Chew, and A. D. Rubin: A framework for detection and measurement of phishing attacks, Alexandria, Viriginia, USA: ACM, 2007, pp. 1-8.
[38] Sophie Gastellier-Prevost, Gustavo Gonzalez Granadillo and Maryline Laurent, Decisive heuristics to differentiate legitimate from phishing sites, In Proc. of Conference on Network and Information Systems Security (SAR-SSI), May 2011, La Rochelle, France. pp.1-9.
[39] Symantec Global Intelligence Network, State of Spam and Phishing Report, February, 2010. http://eval.symantec.com/mktginfo/enterprise/other_resources/b-state_of_spam_and_phishing_report_02-2010.en-us.pdf, Visited on June 2012.
[40] Xiang G, Hong J, Rose, C. P., and Cranor, L: CANTINA+: A feature-rich machine learning framework for detecting phishing Web sites. ACM Trans. Inf. Syst. Secur. 14, 2, Article 21 (September 2011), 28 pages. DOI = 10.1145/2019599.2019606 http://doi.acm.org/10.1145/2019599.2019606
[41] Y. Joshi, S. Saklikar, D. Das, S. Saha, PhishGuard: a browser plug-in for protection from phishing, in: Proc. of the 2nd International Conference on Internet Multimedia Services Architecture and Applications, Bangalore, India, December 2008, pp. 1-6
[42] Yue Zhang , Jason I. Hong , Lorrie F. Cranor: CANTINA - a content-based approach to detecting phishing web sites, In Proc. of the 16th international conference on World Wide Web, Banff, Alberta, Canada, May 08-12, 2007, pp. 639-648.
[43] Yue, C. and Wang, H., BogusBiter: A transparent protection against phishing attacks. ACM Trans. Internet Technol. 10, 2, Article 6 (May 2010), 31 pages. DOI = 10.1145/1754393.1754395 http://doi.acm.org/10.1145/1754393.1754395