# DETECTION OF E-BANKING PHISHING WEBSITES

**E.Konda Reddy[1], Dr. Rajamani[2] and Dr. M. V. Vijaya Saradhi[3]**
[1]PG Scholar, [2]Dean Informatics & Professor and [3]Head Of the Department and Professor
Department of IT, Aurora Engineering College, Bhongir, Nalgonda, Andhra Pradesh, India.

***Abstract-*** Phishing is a new type of network attack where the attacker creates a replica of an existing web page to fool users in to submitting personal, financial, or password data to what they think is their service provider's website. The concept is an end-host based anti-phishing algorithm, called the Link Guard, by utilizing the generic characteristics of the hyperlinks in phishing attacks. The link Guard algorithm is the concept for finding the phishing emails sent by the phisher to grasp the information of the end user. Link Guard is based on the careful analysis of the characteristics of phishing hyperlinks. Each end user is implemented with Link Guard algorithm. After doing so the end user recognizes the phishing emails and can avoid responding to such mails. Since Link Guard is characteristics based it can detect and prevent not only known phishing attacks but also unknown ones. The project uses the Java technologies and Oracle.

**Keywords-** Phishing, Fuzzy Logic, Data Mining, Classification, association, e-banking risk assessment

## I.INTRODUCTION

Phishing is a new word produced from 'fishing', it refers to the act that the attacker allure users to visit a faked Web site by sending them faked e-mails (or instant messages), and stealthily get victim's personal information such as user name, password, and national security ID, etc.

This information then can be used for future target advertisements or even identity theft attacks (e.g., transfer money from victims' bank account). The frequently used attack method is to send e-mails to potential victims, which seemed to be sent by banks, online organizations, or ISPs. In these e-mails, they will make up some causes, e.g. the password of your credit card had been mis-entered for many times, or they are providing upgrading services, to allure you visit their Web site to conform or modify your account number and password through the hyperlink provided in the e-mail.

If you input the account number and password, the attackers then successfully collect the information at the server side, and is able to perform their next step actions with that information (e.g., withdraw money out from your account).Phishing itself is not a new concept, but it's increasingly used by phishers to steal user information and perform business crime in recent years. Within one to two years, the number of phishing attacks increased dramatically. Our analysis identifies that the phishing hyperlinks share one or more characteristics as listed below:

1) The visual link and the actual link are not the same;
2) The attackers often use dotted decimal IP address instead of DNS name;
3) Special tricks are used to encode the hyperlinks maliciously;
4) The attackers often use fake DNS names that are similar (but not identical) with the target Web site.

We then propose an end-host based anti-phishing algorithm which we call Link Guard, based on the characteristics of the phishing hyperlink. Since Link Guard is character-based, it can detect and prevent not only known phishing attacks but also unknown ones. We have implemented Link Guard in Windows XP, and our experiments indicate that Link Guard is light-weighted in that it consumes very little memory and CPU circles, and most importantly, it is very effective in detecting phishing attacks with minimal false negatives.

The paper is organized as follows: Section 2 presents the literature review and related work. Section 3 presents the existing anti phishing approaches. Section 4 introduces the system design and implementation of Link Guard approach. and then conclusions and future work are given in Section 5.

## II. LITERATURE REVIEW AND RELATED WORK

### A. Literature Review

Phishing website is a recent problem, nevertheless due to its huge impact on the financial and on-line retailing sectors and since preventing such attacks is an important step towards defending against e-banking phishing website attacks, there are several promising approaches to this problem and a comprehensive collection of related works. In this section, we briefly survey existing anti-phishing solutions and list of the related works. One approach is to stop phishing at the email level [3], since most current phishing attacks use broadcast email (spam) to lure victims to a phishing website [19]. Another approach is to use security toolbars. The phishing filter in IE7 [18] is a toolbar approach with more features such as blocking the user‘s activity with a detected phishing site. Other approach is to visually differentiate the phishing sites from the spoofed legitimate sites. Dynamic Security Skins [5] proposes to use a randomly generated visual hash to customize the browser window or web form elements to indicate the successfully authenticated sites. A fourth approach is two- factor authentication, which ensures that the user not only knows a secret but also presents a security token [6]. However, this approach is a server-side solution. Phishing can still happen at sites that do not support two-factor authentication. Sensitive information that is not related to a specific site, *e.g.*, credit card information and SSN, cannot be protected by this approach either [20].

However, an automatic anti-phishing method is seldom reported. The typical technologies of anti-phishing from the User Interface aspect are done by [5] and [20]. They proposed methods that need Web page creators to follow certain rules to create Web pages, either by adding dynamic skin to Web pages or adding sensitive information location attributes to HTML code. However, it is difficult to convince all Web page creators to follow the rules [7].

**B. Main Characteristics of e-banking phishing websites.**

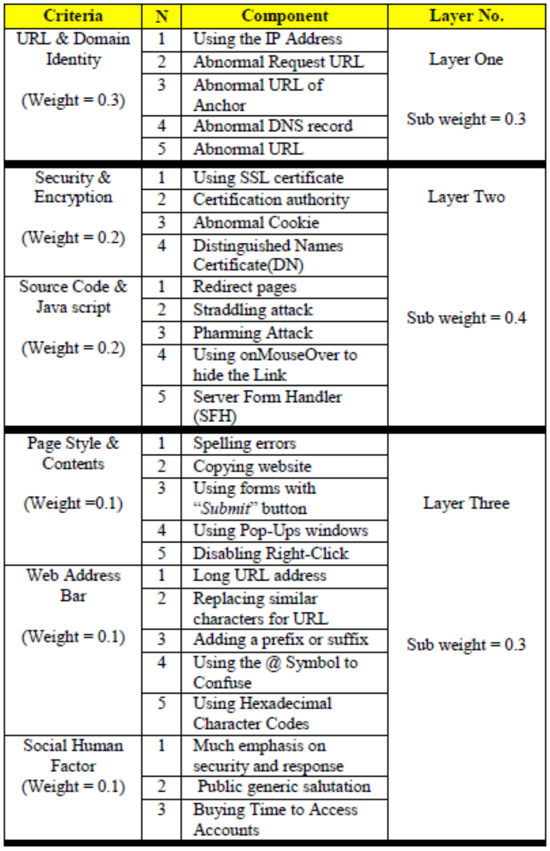
Evolving with the anti-phishing techniques, various phishing techniques and more complicated and hard-to-detect methods are used by phishers. The most straightforward way for a phisher to defraud people is to make the phishing Web pages similar to their targets. Actually, there are many characteristics and factors that can distinguish the original legitimate website from the forged e-banking phishing website like Spelling errors, Long URL address and Abnormal DNS record. The full list is shown in table I which will be used later on our analysis and methodology study.

## III. EXISTING SYSTEM

We briefly review the approaches for anti-phishing.

**1) Detect and block the phishing Web sites in time:** If we can detect the phishing Web sites in time, we then can block the sites and prevent phishing attacks. It's relatively easy to (manually) determine whether a site is a phishing site or not, but it's difficult to find those phishing sites out in time. Here we list two methods for phishing site detection.

Table I. COMPONENTS AND LAYERS OF E-BANKING PHISHING WEBSITE CRITERIA.



| Criteria | N | Component | Layer No. |
|---|---|---|---|
| URL & Domain Identity<br>(Weight = 0.3) | 1 | Using the IP Address | Layer One<br>Sub weight = 0.3 |
| | 2 | Abnormal Request URL | |
| | 3 | Abnormal URL of Anchor | |
| | 4 | Abnormal DNS record | |
| | 5 | Abnormal URL | |
| Security & Encryption<br>(Weight = 0.2) | 1 | Using SSL certificate | Layer Two<br>Sub weight = 0.4 |
| | 2 | Certification authority | |
| | 3 | Abnormal Cookie | |
| | 4 | Distinguished Names Certificate(DN) | |
| Source Code & Java script<br>(Weight = 0.2) | 1 | Redirect pages | |
| | 2 | Straddling attack | |
| | 3 | Pharming Attack | |
| | 4 | Using onMouseOver to hide the Link | |
| | 5 | Server Form Handler (SFH) | |
| Page Style & Contents<br>(Weight =0.1) | 1 | Spelling errors | Layer Three<br>Sub weight = 0.3 |
| | 2 | Copying website | |
| | 3 | Using forms with "*Submit*" button | |
| | 4 | Using Pop-Ups windows | |
| | 5 | Disabling Right-Click | |
| Web Address Bar<br>(Weight = 0.1) | 1 | Long URL address | |
| | 2 | Replacing similar characters for URL | |
| | 3 | Adding a prefix or suffix | |
| | 4 | Using the @ Symbol to Confuse | |
| | 5 | Using Hexadecimal Character Codes | |
| Social Human Factor<br>(Weight = 0.1) | 1 | Much emphasis on security and response | |
| | 2 | Public generic salutation | |
| | 3 | Buying Time to Access Accounts | |

A) The Web master of a legal Web site periodically scans the root DNS for suspicious sites (e.g. www. 1 cbc.com.cn vs. www.icbc.com.cn).

B) Since the phisher must duplicate the content of the target site, he must use tools to (automatically) download the Web pages from the target site.

It is therefore possible to detect this kind of download at the Web server and trace back to the phisher. Both approaches have shortcomings. For DNS scanning, it increases the overhead of the DNS systems and may cause problem for normal DNS queries, and furthermore, many phishing attacks simply do not require a DNS name. For phishing download detection, clever phishers may easily write tools which can mimic the behavior of human beings to defeat the detection.

**2) Enhance the security of the web sites:**

The business Websites such as the Web sites of banks can take new methods to guarantee the security of users' personal information. One method to enhance the security is to use hardware devices. For example, the Barclays bank provides a hand-held card reader to the users. Before shopping in the net, users need to insert their credit card into the card reader, and input their (personal identification number) PIN code, then the card reader will produce a onetime security password, users can perform transactions only after the right password is input. Another method is to use the biometrics characteristic (e.g. voice, fingerprint, iris, etc.) for user authentication. For example, PayPal had tried to replace the single password verification by voice recognition to enhance the security of the Web site.

With these methods, the phishers cannot accomplish their tasks even after they have gotten part of the victims' information. However, all these techniques need additional hardware to realize the authentication between the users and the Web sites hence will increase the cost and bring certain inconvenience. Therefore, it still needs time for these techniques to be widely adopted.

**3) Block the phishing e-mails by various spam filters:** Phishers generally use e-mails as 'bait' to allure potential victims. SMTP (Simple Mail Transfer Protocol) is the protocol to deliver e-mails in the Internet. It is a very simple protocol which lacks necessary authentication mechanisms. Information related to sender, such as the name and email address of the sender, route of the message, etc., can be counterfeited in SMTP. Thus, the attackers can send out large amounts of spoofed e-mails which are seemed from legitimate organizations. The phishers hide their identities when sending the spoofed e-mails, therefore, if anti-spam systems can determine whether an e-mail is sent by the announced sender (Am I Whom I Say I Am?), the phishing attacks will be decreased dramatically.

From this point, the techniques that preventing senders from counterfeiting their Send ID (e.g. SIDF of Microsoft) can defeat phishing attacks efficiently. SIDF is a combination of Microsoft's Caller ID for E-mail and the SPF (Sender Policy Framework) developed by Meng Weng Wong. Both Caller ID and SPF check e-mail sender's domain name to verify if the e-mail is sent from a server that is authorized to send e-mails of that domain and from that to determine whether that e-mail use spoofed e-mail address. If it's faked, the Internet service provider can then determine that e-mail is a spam e-mail. The spoofed e-mails used by phishers are one type of spam e-mails. From this point of view, the spam filters can also be used to filter those phishing e-mails. For example, blacklist, white list, keyword filters, Bayesian filters with self learning abilities, and E-Mail Stamp, etc., can all be used at the e-mail server or client systems. Most of these anti-spam techniques perform filtering at the receiving side by scanning the contents and the address of the received e-mails. And they all have pros and cons as discussed below. Blacklist and whitelist cannot work if the names of the spammers are not known in advance. Keyword filter and Bayesian filters can detect spam based on content, hence can detect unknown spasm. But they can also result in false positives and false negatives. Furthermore, spam filters are designed for general spam e-mails and may not very suitable for filtering phishing e-mails since they generally do not consider the specific characteristics of phishing attacks.

**4) Install online anti-phishing software in user's computers:** Despite all the above efforts, it is still possible for the users to visit the spoofed Web sites. As a last defense, users can install anti-phishing tools in their computers. The antiphishing tools in use today can be divided into two categories: blacklist/white list based and rule-based.

**Category I:** When a user visits a Web site, the antiphishing tool searches the address of that site in a blacklist stored in the database. If the visited site is on the list, the anti-phishing tool then warns the users. Tools in this category include Scam Blocker from the EarthLink Company, Phish Guard, and Net craft, etc. Though the developers of these tools all announced that they can update the blacklist in time, they cannot prevent the attacks from the newly emerged (unknown) phishing sites.

**Category II:** this category of tools uses certain rules in their software, and checks the security of a Web site according to these rules. Examples of this type of tools include Spoof Guard developed by Stanford, Trust Watch of the Geo Trust, etc. Spoof Guard checks the domain name, URL (includes the port number) of Web site, it also checks whether the browser is directed to the current URL via the links in the contents of e-mails. If it finds that the domain name of the visited Web site is similar to a well-known domain name, or if they are not using the standard port, Spoof Guard will warn the users. In Trust Watch, the security of a Web site is determined by whether it has been reviewed by an independent trusted third party organization. Both Spoof Guard and Trust Watch provide a toolbar in the browsers to notify their users whether the Web site is verified and trusted.

It is easy to observe that all the above defense methods are useful and complementary to each other, but none of them are perfect at the current stage.

## IV. PROPOSED SYSTEM AND ITS IMPLEMENTATION

In this section we explain the basic algorithm of Link Guard Approach which can detect the phishing content, based on the characteristics of the phishing hyperlink.

## LINKGUARD

### A.Classification of the hyperlinks in the phishing e-mails

In order to (illegally) collect useful information from potential victims, phishers generally tries to convince the users to click the hyperlink embedded in the phishing e-mail. A hyperlink has a structure as follows.

<a href="URI"> Anchor text <\a>

where 'URI' (universal resource identifiers) provides the necessary information needed for the user to access the networked resource and 'Anchor text' is the text that will be displayed in user's Web browser. Examples of URIs are

http://www.google.com,
https://www.icbc.com.cn/login.html,

ftp://61.112.1.90:2345, etc. 'Anchor text' in general is used to display information related to the URI to help the user to better understand the resources provided by the hyperlink. In the following hyperlink, the URI links to the phishing archives provided by the APWG group, and its anchor text "Phishing Archive" informs the user what's the hyperlink is about.

<a href"http://www.antiphishing.org/phishing archive.html">
Phishing Archive
</a>

Note that the content of the URI will not be displayed in user's Web browser. Phishers therefore can utilize this fact to play trick in their 'bait' e-mails. In the rest of the paper, we call the URI in the hyperlink the actual link and the anchor text the visual link. After analyzing the 203 (there are altogether 210 phishing e-mails, with 7 of them with incomplete information or with malware attachment and do not have hyperlinks) phishing email archives from Sep. 21st 2003 to July 4th 2005 provided by APWG [6]. We classified the hyperlinks used in the phishing e-mail into the following categories:

1) The hyperlink provides DNS domain names in the anchor text, but the destination DNS name in the visible link doesn't match that in the actual link. For instance, the following hyperlink:
<a href = "http://www.profusenet.net/checksession.php">
https://secure.regionset.com/EBanking/logon/</a>
appears to be linked to secure.regionset.com, which is the portal of a bank, but it actually is linked to a phishing site www.profusenet.net.

2) Dotted decimal IP address is used directly in the URI or the anchor text instead of DNS name. See below for an example.

<a href= "http://61.129.33.105/secured site/www.skyfi.com/index.html?MfcISAPICommand=SignInFPP& UsingSSL=1"> SIGN IN</a>

3) The hyperlink is counterfeited maliciously by using certain encoding schemes. There are two cases: a) The link is formed by encoding alphabets into their corresponding ASCII codes. See below for such a hyperlink.
<a
href="http://%34%2E%33%34%2E%31%39%35%2E%34%31:%34%39%30%33/%6C/%69%6E%64%65%78%2E%68%74%6D"> www.citibank.com </a>
while this link is seemed pointed www.citibank.com, it actually points to http://4.34.195.41:34/l/index.htm.

b) Special characters (e.g. @ in the visible link) are used to fool the user to believe that the e-mail is from a trusted sender. For instance, the following link seems is linked to amazon, but it actually is linked to IP address 69.10.142.34.
http://www.amazon.com:fvthsgbljhfcs83infoupdate @69.10.142.34.

4) The hyperlink does not provide destination information in its anchor text and uses DNS names in its URI. The DNS name in the URI usually is similar with a famous company or organization. For instance, the following link seems to be sent from paypal, but it actually is not. Since paypal-cgi is actually registered by the phisher to let the users believe that it has something to do with paypal

<a href= "http://www.paypal-cgi.us/webscr.php? cmd=LogIn"> Click here to confirm your account
</a>

5) The attackers utilize the vulnerabilities of the target Web site to redirect users to their phishing sites or to launch CSS (cross site scripting) attacks. For example, the following link
<a href="http://usa.visa.com/track/dyredir.jsp?rDirl=
http://200.251.251.10/.verified/"> Click here <a>
Once clicked, will redirect the user to the phishing site 200.251.251.10 due to a vulnerability of usa.visa.com. Table 1 summarizes the number of hyperlinks and their percentages for all the categories. It can be observed that most of the phishing e-mails use faked DNS names (category 1,44.33%) or dotted decimal IP addresses (category 2, 41.87%).

Encoding tricks are also frequently used (category 3a and 3b, 17.24%). And phishing attackers often try to fool users by setting up DNS names that are very similar with the real ecommerce sites or by not providing destination information in the anchor text (category 4).

Phishing attacks that utilize the vulnerability of Web sites (category 5) are of small number (2%) and we leave this type of attacks for future study.

Note that a phishing hyperlink can belong to several categories at the same time. For instance, an attacker may use tricks from both categories 1 and 3 at the same time to increase his success chance. Hence the sum of percentages is larger than 1.

| Category | Number of links | Percentage |
|---|---|---|
| 1 | 90 | 44.33% |
| 2 | 85 | 41.87% |
| 3.a | 19 | 9.36% |
| 3.b | 16 | 7.88% |
| 4 | 6 | 7.33% |
| 5 | 4 | 2% |

TABLE 2
THE CATEGORIES OF HYPERLINKS IN PHISHING E-MAILS.

Once the characteristics of the phishing hyperlinks and understood, we are able to design anti-phishing algorithms that can detect known or unknown phishing attacks in real-time. We present our LinkGuard algorithm in the next subsection.

### B. The LinkGuard algorithm

LinkGuard works by analyzing the differences between the visual link and the actual link. It also calculates the similarities of a URI with a known trusted site. The algorithm is illustrated in Fig. 1. The following terminologies are used in the algorithm.

v_link: visual link;
a_link: actual_link;
v_dns: visual DNS name;
a_dns: actual DNS name;
sender_dns: sender's DNS name.

```
int LinkGuard(v_link, a_link} {
1 v_dns = GetDNSName(v_link);
2 a_dns = GetDNSName(a_link);
3 if ((v_dns and a_dns are not
4 empty) and (v_dns != a_dns))
5 return PHISHING;
6 if (a_dns is dotted decimal)
7 return POSSIBLE_PHISHING;
8 if(a_link or v_link is encoded)
9 {
10 v_link2 = decode (v_link);
11 a_link2 = decode (a_link);
12 return LinkGuard(v_link2, a_link2);
13 }
14 /* analyze the domain name for
15 possible phishing */
16 if(v_dns is NULL)
17 return AnalyzeDNS(a_link);
}
```

Fig. 1. Description of the LinkGuard algorithm.

The LinkGuard algorithm works as follows. In its main routine *LinkGuard*, it first extracts the DNS names from the actual and the visual links (lines 1 and 2). It then compares the actual and visual DNS names, if these names are not the same, then it is phishing of category 1 (lines 3-5). If dotted decimal IP address is directly used in actual dns , it is then a possible phishing attack of category 2 (lines 6 and 7). We will delay the discussion of how to handle possible phishing attacks later. If the actual link or the visual link is encoded (categories 3 and 4), we first decode the links, then recursively call LinkGuard to return a result (lines 8-13). When there is no destination information (DNS name or dotted IP address) in the visual link (category 5), LinkGuard calls AnalyzeDNS to analyze the actual dns (lines 16 and 17). LinkGuard therefore handles all the 5 categories of phishing attacks.

```
int AnalyzeDNS (actual_link) {
/* Analyze the actual DNS name according
to the blacklist and whitelist*/
18 if (actual_dns in blacklist)
19 return PHISHING;
20 if (actual_dns in whitelist)
21 return NOTPHISHING;
22 return PatternMatching(actual_link);
}
int PatternMatching(actual_link){
23 if (sender_dns and actual_dns are different)
24 return POSSIBLE_PHISHING;
25 for (each item prev_dns in seed_set)
26 {
27 bv = Similarity(prev_dns, actual_link);
28 if (bv == true)
29 return POSSIBLE_PHISHING;
30 }
31 return NO_PHISHING;
}
float Similarity (str, actual_link) {
32 if (str is part of actual_link)
33 return true;
34 int maxlen = the maximum string
35 lengths of str and actual_dns;
36 int minchange = the minimum number of
37 changes needed to transform str
38 to actual_dns (or vice verse);
39 if (thresh<(maxlen-minchange)/maxlen<1)
40 return true
41 return false;
}
```

Fig. 2. The subroutines used in the LinkGuard algorithm.

AnalyzeDNS and the related subroutines are depicted in Fig.2. In AnalyzeDNS, if the actual dns name is contained in the blacklist, then we are sure that it is a phishing attack (lines 18 and 19). Similarly, if the actual dns is contained in the

whitelist, it is therefore not a phishing attack (lines 20 and 21). If the actual dns is not contained in either whitelist or blacklist, PatternMatching is then invoked (line 22). PatternMatching is designed to handle unknown attacks (blacklist/whitelist is useless in this case). For category 5 of the phishing attacks, all the information we have is the actual link from the hyperlink (since the visual link does not contain

DNS or IP address of the destination site), which provide very little information for further analysis. In order to resolve this problem, we try two methods: First, we extract the sender email address from the e-mail. Since phishers generally try to fool users by using (spoofed) legal DNS names in the sender e-mail address, we expect that the DNS name in the sender address will be different from that in the actual link. Second, we proactively collect DNS names that are manually input by the user when she surfs the Internet and store the names into a seed set, and since these names are input by the user by hand, we assume that these names are trustworthy. PatternMatching then checks if the actual DNS name of a hyperlink is different from the DNS name in the sender's address (lines 23 and 24), and if it is quite similar (but not identical) with one or more names in the seed set by invoking the Similarity (lines 25-30) procedure. Similarity checks the maximum likelihood of actual dns and the DNS names in seed set. As depicted in Fig. 2, the similarity index between two strings are determined by calculating the minimal number of changes (including insertion, deletion, or revision of a character in the string) needed to transform a string to the other string. If the number of changes is 0, then the two strings are identical; if the number of changes is small, then they are of high similarity; otherwise, they are of low similarity. For example, the similarity index of 'microsoft' and 'micr0s0ft' is 7/9 (since we need change the 2 '0's in micr0s0ft to 'o'. Similarly, the similarity index of 'paypal' and 'paypal-cgi' is 6/10 (since we need to remove the last 4 chars from paypal-cgi), and the similarity index of '95559'

and '955559' is 5/6 (since we need to insert a '5' to change '95559' to '955559').

If the two DNS names are similar but not identical, then it is a possible phishing attack. For instance, PatternMatching can easily detect the difference between www.icbc.com.cn (which is a good e-commerce Web site) and www.1cbc.com.cn (which is a phishing site), which has similarity index 75%. Note that PatternMatching may treat www.1cbc.com.cn as

a normal site if the user had never visit www.1cbc.com.cn before. This false negative, however, is unlikely to cause any severe privacy or financial lose to the user, since she actually does not have anything to lose regarding the Web site www.icbc.com.cn (since she never visits that Web site before)!

### *C. False positives and false negatives handling*

Since LinkGuard is a rule-based heuristic algorithm, it may cause false positives (i.e., treat non-phishing site as phishing site) and false negatives (i.e., treat phishing site as nonphishing site). In what follows, we show that LinkGuard may result in false positives but is very unlikely to cause harmful false negatives.

For phishing attacks of category 1, we are sure that there is no false positives or false negatives, since the DNS names of the visual and actual links are not the same. It is also easy to observe that LinkGuard handles categories 3 and 4 correctly since the encoded links are first decoded before further analysis. For category 2, LinkGuard may result in false positives, since using dotted decimal IP addresses instead of domain names may be desirable in some special circumstances (e.g., when the DNS names are still not registered). For category 5, LinkGuard may also result in false positives. For example, we know that both 'www.iee.org' and 'www.ieee.org' are legal Web sites. But these two DNS names have a similarity index of 3/4, hence is very likely to trigger a false positive.

When it is a possible false positive, LinkGuard will return a POSSIBLE PHISHING. In our implementation (which will be described in the next section), we leverage the user to judge if it is a phishing attack by prompting a dialogue box with detailed information of the hyperlink. The rationale behind this choice is that users generally may have more knowledge of a link than a computer in certain circumstances (e.g., the user may know that the dotted decimal IP address is the address of his friend's computer and that www.iee.org is a respected site for electrical engineers).

For category 5, LinkGuard may also result in false negatives. False negatives are more harmful than false positives, since attackers in this case will succeed in leading the victim to the phishing sites. For instance, when the sender's e-mail address and the DNS name in the actual link are the same and the DNS name in the actual link has a very low similarity index with the target site, LinkGuard will return NO PHISHING. For instance, PatternMatching will treat the below link as NO PHISHING.

```
<a href="http://fdic-secure.com/
application.htm"> Click here </a>
```

with "securehq@fdic-secure.com" as the sender address. We note that this kind of false negatives is very unlikely to result in information leakage, since the end user is very unlikely to have information the attack interested (since the DNS name in this link is not similar with any legal Web sites).

## V. IMPLEMENTATION AND VERIFICATION OF LINKGUARD

We have implemented the LinkGuard algorithm in Windows XP. It includes two parts: a whook.dll dynamic library and a LinkGuard executive. The structure of the implementation is depicted in Fig. 3.

This Link Guard algorithm is the concept for finding the phishing e-mails Sent by the phishers to grasp the information's of the end user. Link Guard is based the careful analysis of the characteristics of phishing hyperlinks. Link Guard has a verified very low false negative rate for unknown phishing attacks . This Link Guard algorithm is the concept for finding the phishing e-mails Sent by the phishers to grasp the information of the end user **,** So each end user will be implemented with the Link Guard algorithm , After implementing the Link Guard algorithm now the end user may able to find the phishing attacks, and can avoid responding phishing e-mails .

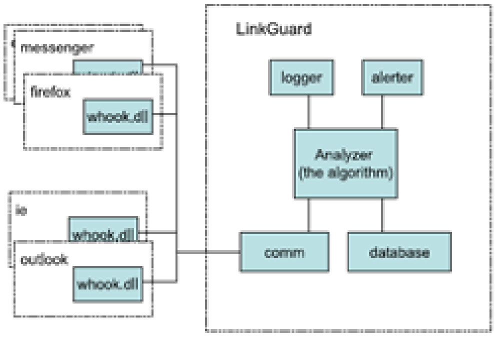


Fig. 3. The structure of the LinkGuard implementation, which consists of a whook.dll and a LinkGuard executive.

Since Link Guard is character-based, it can detect and prevent not only known phishing attacks but also unknown ones.

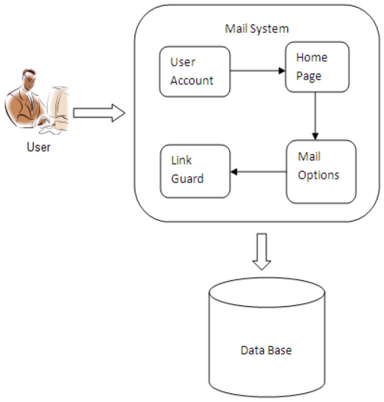


**Figure 4: System Architecture of Anti phishing system along with Link Guard Approach**

## MODULES OF PROPOSED SYSTEM:

- Creation of a mail system and database operations
- Composes, send and receive a mail
- Implementation of the Link Guard algorithm

The mail system module deals with the user interface for the home page, sign-in, sign-up and forgot your password pages. This module enables a new user to Sing-Up. It also enables an existing user to Sign-In. The user may use the Forget password link if he did forget his password. The password is retrieved on the basis of security question and answer given by the user. Database operation manages the users. Every time a new user signs in his details are written in to the database. Every time an existing user logs on his details are checked up for with the database.

The second module enables the user to compose and send a mail. It also allows the user to read a received mail. Once a mail is sent the date and the subject of the mail gets displayed. The received mail can be checked if it is phishing or not, the implementation of which is given in the next module. The compose mail option contains an option for spoof id. The spoof id allows the mail of the composer to be delivered with a different from address. This is being incorporated to demonstrate the Link Guard algorithm.

The module contains the implementation of the Link Guard algorithm. It is possible for the user to add domain names and categorize them as either white list or black list under settings. Whenever a mail is detected as phishing the domain name in that mail automatically gets added as black list. The Link Guard algorithm checks if the domain names fall under any of the 5 categories of hyperlinks for phishing emails. It also refers to the database of black and white list entries and sets the status of the mail as either **Phishing** or **Non-Phishing.**

Once the mail is categorized as Phishing the user can take care that he does not open the link or submit any personal, critical information on to the website.

**Communicator:** This collects the information of the input process, and sends these related information's to the Analyzer.

**Database:** Store the white list, blacklist, and the user input URLs.

**Analyzer**: It is the key component of Link Guard, which implements the Link Guard algorithm; it uses data provided by Communicator and Database, and sends the results to the Alert and Logger modules.

**Alerter**:When receiving warning messages from Analyzer, it shows the related information to alert the users and send back the reactions of the user back to the Analyzer.

**Logger:**Archive the history information, such as user events, alert information, for future use.

## VI. CONCLUSION AND FUTURE WORK

Phishing has becoming a serious network security problem, causing finical lose of billions of dollars to both consumers and e-commerce companies. And perhaps more fundamentally, phishing has made e-commerce distrusted and less attractive to normal consumers. In this paper, we have studied the characteristics of the hyperlinks that were embedded in phishing e-mails.

We then designed an anti-phishing algorithm, Link-Guard, based on the derived characteristics. Since Link-Guard is characteristic based, it can not only detect known attacks, but also is effective to the unknown ones. We have implemented Link Guard for Windows XP. Our experiment showed that Link Guard is light-weighted and can detect up to 96% unknown phishing attacks in real-time. We believe that Link Guard is not only useful for detecting phishing attacks, but also can shield users from malicious or unsolicited links in Web pages and Instant messages.

As we have implemented this approach by considering the URL and Domain Identity Criteria, there are the different criteria needs to work in future.

## REFERENCES

[1] Androutsopoulos, J. Koutsias, K.V Chandrinos, and C.D. Spyropoulos.An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Encrypted Personal E-mail Message.In Proc. SIGIR 2000, 2000.

[2] The Anti-phishing working group. http://www.antiphishing.org/.Neil Chou, Robert Ledesma, Yuka Teraguchi, Dan Boneh, and John C.Mitchell. Client-side defense against web-based identity theft. In Proc.NDSS 2004, 2004.

[3] B. Adida, S. Hohenberger and R. Rivest, ―Lightweight Encryption for Email,‖ USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), 2005.

[4] Cynthia Dwork, Andrew Goldberg, and Moni Naor. On Memory-Bound.Functions for Fighting Spam. In Proc. Crypto 2003, 2003.

[5] R. Dhamija and J.D. Tygar, ―The Battle against Phishing: Dynamic Security Skins,‖ Proc. Symp. Usable Privacy and Security, 2005.

[6] FDIC., ―Putting an End to Account-Hijacking Identity Theft,‖ http://www.fdic.gov/consumers/consumer/idtheftstudy/identity_theft.pdf, 2004.

[7] A. Y. Fu, L. Wenyin and X. Deng, ― Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover‘s Distance (EMD) ,‖ IEEE transactions on dependable and secure computing, vol. 3, no. 4, 2006.

[8] EarthLink. ScamBlocker. http://www.earthlink.net/software/free/toolbar/ .

[9] David Geer. Security Technologies Go Phishing. IEEE Computer, 38 (6):18-21, 2005.

[10] John Leyden. Trusted search software labels fraud site as safe'. http://www.theregister.co.uk/2005/09/27/untrusted-search/.

[11] Microsoft. Sender ID Framework. http://www.microsoft.com/

[12] mscorp/safety/technologies/senderid/default.mspx.

[13] Net craft. Net craft toolbar. http://toolbar.netcraft.com/.

[14] PhishGuard.com. Protect Against Internet Phishing Scams http://www.phishguard.com/.

[15] Jonathan B. Postel. Simple Mail Transfer Protocol. RFC821:http:llwww.ietf. org/rfc/rfcO82 1 .txt.

[16] Georgina Stanley. Internet Security - Gone phishing.http://www.cyota.com/news.asp?id=1 14.

[17] Meng Weng Wong. Sender ID SPF. http://www.openspf.org/whitepaper.pdf.

[18] T. Sharif, Phishing Filter in IE7, http://blogs.msdn.com/ie/archive/2005/09/09/463204.aspx, September 9, 2006.

[19] M. Wu, R. C. Miller and S. L. Garfinkel , ―*Do Security Toolbars Actually Prevent Phishing Attacks?,"* CHI April 2006.

[20] M. Wu, R. C. Miller and G. Little, ―Web Wallet: Preventing Phishing Attacks by Revealing User Intentions,‖ MIT Computer Science and Artificial Intelligence Lab, 2006.

**Author Profile**

E Konda Reddy, Pursuing M.Tech in the department of Information Technology, in Aurora's Engineerng College, Bhongir, Nalgonda Dist, Andhra Pradesh, India.

Dr.A.Rajamani, Professor, Dean Computer Science & Information Technology, He is currently working with Aurora's Engineering college, Bhongir, Nalgonda Dist, Andhra Pradesh, India.

**Dr. M.V.Vijaya Saradhi** received his Ph.D degree from Faculty of Engineering, Osmania University (OU), Hyderabad, Andhra Pradesh, India**. He** is Currently Working as Professor in the Department of Information Technology (IT) at Aurora's Engineering College, Bhongiri, Andhra Pradesh, India. His main research interests are Software Metrics, Distributed Systems, Object-Oriented Modeling, Data Mining, Design Patterns, Object- Oriented Design Measurements and Empirical Software Engineering. He is a life member of various Professional bodies like MIETE, MCSI, MIE, MISTE.