Enhanced Online Anti-Phishing System with EGuard Algorithm

Article · September 2009

CITATIONS
0

READS
76

2 authors, including:

Dr. Meka James Stephen
WELLFARE INSTITUTE OF SCIENCE TECHNOLOGY AND MANAGEMENT (WISTM)
22 PUBLICATIONS **52** CITATIONS

SEE PROFILE

ENHANCED ONLINE ANTI PHISHING WITH EGUARD ALGORITHM

M. James Stephen *, K. Venkata Rao**

Abstract

(fish'ing) (n.) The act of sending an e-mail to a user falsely claiming to be an established legitimate enterprise in an attempt to scam the user into surrendering private information that will be used for identity theft. The e-mail directs the user to visit a Web site where they are asked to update personal information, such as passwords and credit card, social security, and bank account numbers, that the legitimate organization already has. The Web site, however, is bogus and set up only to steal the user's information.

In this paper, we propose a new end-host based anti-phishing algorithm, which we call EGuard, by utilizing the generic characteristics of the hyperlinks in phishing attacks. These characteristics are derived by analyzing the phishing data archive provided by the Anti-Phishing Working Group (APWG). Because it is based on the generic characteristics of phishing attacks, EGuard can detect not only known but also unknown phishing attacks. We have implemented EGuard in Windows XP. Our experiments verified that EGuard is effective to detect and prevent both known and unknown phishing attacks with minimal false positive and false negatives. The success rate of EGuard is about 96%. Our experiments also showed that EGuard is light weighted and can detect and prevent phishing attacks in real time.

Index Terms: Network security, Phishing attacks, Hyperlink, EGuard algorithm

1. INTRODUCTION

The word phishing comes from the analogy that Internet scammers are using e-mail **lures** to **fish** for passwords and financial data from the **sea** of Internet users. The term was coined in 1996 by hackers who were stealing AOL Internet accounts by scamming passwords from unsuspecting AOL users. Since hackers have a tendency to replacing "f" with "ph" the term phishing was derived and produced a new word in the hacker's community, since they usually hack by phones.

So 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 makeup some causes,

* Associate Professor, Department of MCA, **ANITS Engineering College**, Visakhapatnam, INDIA

** Associate Professor and Webmaster(AU), Department Of **Computer** Science & Systems Engineering, **Andhra University**., Visakhapatnam, INDIA

e.g. the password of your credit card had been mis-entered for many times, or as if they are providing some 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. You will then be linked to a counterfeited Web site after clicking those links. The style, the functions performed, sometimes even the URL of these faked Web sites are similar to the real Web site. It's very difficult for you to know that you are actually visiting a malicious site. 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. We know what happened in China few years back include the events to counterfeit the Bank of China (real Web site www.bank-ofchina. com, counterfeited Web site www.bank-off-china.com), the Industrial and Commercial Bank of China (real Web site www.icbc.com.cn, faked web site www.1cbc.com.cn)

In this paper, we study the common procedure of phishing attacks and review possible anti-phishing approaches. We then focus on end-host based anti-phishing approach. We first analyze the common characteristics of the hyperlinks in phishing e-mails. Our analysis identifies that the phishing hyperlinks may 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 EGuard, based on the characteristics of the phishing hyperlink. Since EGuard is character-based, it can detect and prevent not only known phishing attacks but also unknown ones. We have implemented EGuard in Windows XP, and our experiments indicate that EGuard 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. EGuard detects about 96% of phishing archives provided by APWG without knowing any signatures of the attacks.

The rest of this paper is organized as follows. In Section II, we give the relevance of the problem with important stats. In Section III, we give the general procedure of a phishing attack; in section IV we provide the available methods to prevent phishing attacks. We then analyze the characteristics of the hyperlinks used in phishing attacks and present the EGuard algorithm in Section V. Section VI describes our implementation of the EGuard system and gives the experimental results. Section VII concludes this paper.

II. The Relevance of the Problem and Important Statistics

Phishing attacks are a growing threat and all computer users have to be aware of this. A phishing email is a message with a credible subject sent by email or instant messenger, asking for confidential data. Once entered, the user's information is no longer confidential and it is immediately used by the fraudsters in their own interest.

Very recently I received an SMS from ICICI Bank that says "Please do not click on any e-mail links for registration of verified by Visa or MasterCardSecureCode.Register at www.icicibank.com or while shopping online" Probably this message had been sent to all of its customers. This shows the fear of Phishing these days because in the recent past most fraud costs were borne by consumer banks

According to Gartner, Inc. (NYSE: IT) is the world's leading information technology research and advisory company, the number of Phishing Attacks on U.S. Consumers Increased 40 Percent in 2008. More than 5 million U.S. consumers lost money to phishing attacks in the 12 months ending in September 2008, a 39.8 percent increase over the number of victims a

year earlier, according to Gartner, Inc. In September of 2008, Gartner surveyed 3,985 U.S. online adults to determine the number of U.S. adults who have been victimized by phishing attacks, as well as the methods being used by criminals to execute these crimes.

"The survey findings underline the fact that the war against phishing is far from over," said Avivah Litan, vice president and distinguished analyst at Gartner. "Despite the rollout of a wide range of security measures designed to stem phishing, the truth is that many of them are not yet adopted widely enough to reverse this tide and, in many cases, their effectiveness is only partial." [15]

The Virus Lab at Avira conducts **phishing origin analysis**, based on the data collected by numerous trap accounts. This information offers an overview of the spreading of phishing attacks world-wide. For a better view of the quantity of phishing emails in time, please observe the **All Phishing Statistics**.

The below graph shows the number phishing attacks in this month (from 28th July to 25th Aug'09)

Fig.1

The following graph shows the number phishing attacks in this week (from 19th to 25th Aug'09)

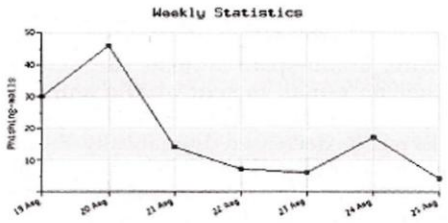

Fig.2

Suppose the spammer sends out 2,000,000 emails. Of that, 5% go to legitimate email addresses, or 100,000. Of that, 5% of the people receiving the phishing email respond, or 5,000. Of that number, only 2% are foolish enough to actually submit their personal information or 100 real people. MailFrontier's site quotes the FTC as saying the average phishing loss is about $1,200. So, for an up front investment of $200, a phisher can make $120,000! Even if those numbers are exaggerated by a factor of 10, no wonder there are so many people out there sending you phishing scams [14]. The below figure shows the number of Phishing attacks recorded in the last three months. The X-axis shows the victim companies and the Y-axis shows the PhishingNumber.

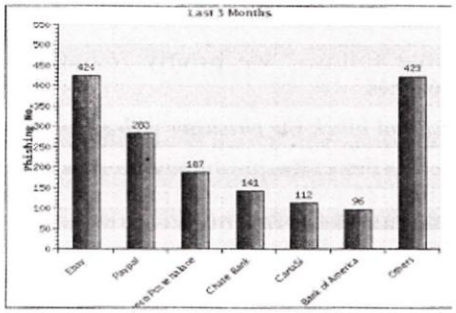

Fig.3

With the above statistics we understand, why it is worthwhile to answer this question and the need of the reliable anti-Phishing system

III GENERAL PHISHING ATTACK PROCEDURE

In general, Phishing attacks are performed with the following four steps:

1) Phishers set up a counterfeited Web site which looks exactly like the legitimate Web site, including setting up the web server, applying the DNS server name, and creating the web pages similar to the destination Web site, etc.

2) Send large amount of spoofed e-mails to target users in the name of those legitimate companies and organizations, trying to convince the potential victims to visit their Web sites.

3) Receivers receive the e-mail, open it, click the spoofed hyperlink in the e-mail, and input the required information.

4) Phishers steal the personal information and perform their fraud such as transferring money from the victims' account.

IV Approaches to Prevent Phishing Attacks

There are several (technical or non-technical) ways to prevent phishing attacks: 1) educate users to understand how phishing attacks work and make them alert when phishing-alike e-mails are received; and also use legal methods to punish phishing attackers; 2) use technical methods to stop phishing attackers. In this paper, we only focus on the second one. Technically, if we can cut off one or several of the steps that needed by a phishing attack, we then successfully prevent that attack.

In what follows, we briefly review these approaches.

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.

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

2) 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.

Enhance the security of the web sites with Hardware devices:

The business Web sites 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 [12]. 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 got 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.

Use spam filters to block the phishing e-mails:

Phishers generally use e-mails as 'bait' to allure potential victims. SMTP (Simple Mail Transfer Protocol) [11] 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 [8]) can defeat phishing attacks efficiently. SIDF is a combination of Microsoft's

Caller ID for E-mail and the SPF (Sender Policy Framework) [13] 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 [1], [4] can also be used to filter those phishing e-mails. For example, blacklist, whitelist, 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 spamers 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.

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 anti-phishing tools in use today can be divided into two categories: blacklist/whitelist based and rule-based.

• Category I: When a user visits a Web site, the anti-phishing 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 ScamBlocker from the EarthLink company [5], PhishGuard [10], and Netcraft [9], 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) phishingsites.

• 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 SpoofGuard developed by Stanford [3], TrustWatch of the GeoTrust [7], etc. SpoofGuard checks the domain name, URL (includes the port number) of a 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, SpoofGuard will warn the users. In TrustWatch, the security of a Web site is determined by whether it has been reviewed by an independent trusted third party organization. Both SpoofGuard and TrustWatch 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. In the rest of the paper, we focus on end-host based approach and propose an endhost based EGuard algorithm for phishing detection and prevention. To this end, our work follows the same approach as [3]. Our work differs from [3] in that:

1) EGuard is based on our careful analysis of the characteristics of phishing hyperlinks whereas SpoofGuard is more like a framework;

2) EGuard has a verified very low false negative rate for unknown phishing attacks whereas the false negative property of SpoofGuard is still not known. In next section, we first study the characteristics of the hyperlinks in phishing e-mails and then we propose the EGuard algorithm. And it is true that the proposed algorithm is similar to the LinkGuard but with the usage of graylist in EGuard, uncertainty can be handled more efficiently.

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. 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 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.

Hyperlinks used in the phishing e-mail can be classified into the following categories:

1. Using false DNS domain name

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:

https://secure.icici.com/EBanking/logon/

appears to be linked to secure.icici.com, which is the portal of a bank, but it actually is linked to a phishing site www.profusenet.net.

2. Using Dotted Decimal IP Address in URI

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

3. Using Encoding schemes

The hyperlink is counterfeited maliciously by using certain encoding schemes. There are two cases:

a) ASCII codes

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

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

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:fvthsgbljhfcs83 infoupdate @69.10.142.34.

4. Uses DNS names in its URI

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

 Click here to confirm your account

5. Vulnerabilities of the target website

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.

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.

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

V. THE EGUARD ALGORITHM

EGuard 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.4

EGuard Algorithm

The following terminologies are used in the algorithm.

VL: visual link;

AL: actual_link;

VD: visual DNS name;

AD: actual DNS name;

SD: sender's DNS name.

```
int EGuard(VL, AL} {
1 VD = GetDNSName(VL);
2 AD = GetDNSName(AL);
3 if ((VD and AD are not
4 empty) and (VD != AD))
5 return PHISHING;
5a else return NO PHISHING
5b add DNS to whitelist
6 if (AD is dotted decimal)
7 Add AD to graylist, set_date to AD
8 return POSSIBLE_PHISHING;
9  return anlyze_graylist(IP address)
10if(AL or VL is encoded)
11 {
12 VL2 = decode (VL);
13 AL2 = decode (AL);
14 return EGuard(VL2, AL2);
15 }
16 /* analyze the domain name for
17 possible phishing */
18 if(VD is NULL)
19 return AnalyzeDNS(AL);
}
```

Fig. 4. Description of the EGuard algorithm.

The EGuard algorithm works as follows. In its main routine *EGuard*, 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 to 9). Here we introduced new list known as graylist to handle this uncertainty more effectively. So EGuard will not 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). But here we assume that the legal web site will not use IP addresses for long time instead of DNS names.

When IP address is found instead of DNS name, then it sets a date to that IP and adds it to graylist. Periodically match is searched for the corresponding DNS of the IP address. If a dns match is found for the IP address and it is validated as a legal web site then it will be removed from the graylist and will be added to the white list. If the time period crosses some

threshold and still the only IP address is present in place of actual DNS then we through it into black list.

Here, we assume that while storing in any list, if the entry is already exists, then the old one is over written to avoid redundancy. The below Figures shows the clear picture of the main routine of EGuard and graylist implementation.

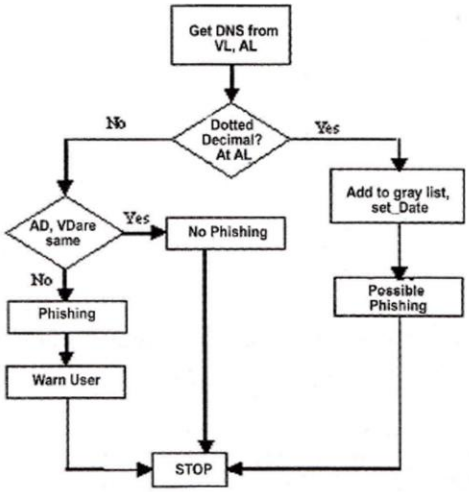

Fig5. flow chart of EGuard main routine

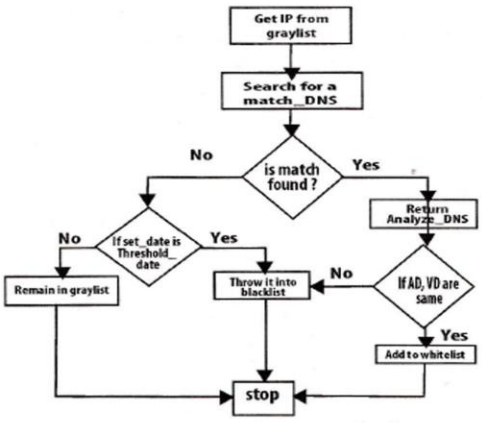

Fig6. Implementing and usage of graylist

```
int AnalyzeDNS (actual_link) {
/* Analyze the actual DNS name according
to the blacklist and whitelist*/
20 if (actual_dns in blacklist)
21 return PHISHING;
22 if (actual_dns in whitelist)
23 return NOTPHISHING;
24   Else add to graylist
25 return PatternMatching(actual_link);
}
/* Handling graylist */
26 int analyze_graylist (IP address)
27 search_match (IP address, dns)
28 If match found
29 return AnalyzeDNS(AL)
30 if set_date > threshold_date
31 put it into blacklist
32 else remain in graylist
33 int PatternMatching(actual_link){
34 if (SD and actual_dns are different)
35 Add actual_dns to blacklist
36 return POSSIBLE_PHISHING;
37 else add actual_dns to whitelist
38 for (each item preVD in seed_set)
39 {
40 bv = Similarity(preVD, actual_link);
41 if (bv == true)
42 return POSSIBLE_PHISHING;
43 }
44 return NO_PHISHING;
}
45 float Similarity (str, actual_link) {
46 if (str is part of actual_link)
47 return true;
```

```
48 int maxlen = the maximum string
49 lengths of str and actual_dns;
50 int minchange = the minimum number of
51 changes needed to transform str
52 to actual_dns (or vice verse);
53 if (thresh<(maxlen-minchange)/maxlen<1)
54 return true
55 return false;
}
```

Fig. 6. The subroutines used in the EGuard algorithm.

If the actual link or the visual link is encoded (categories 3 and 4), we first decode the links, then recursively call *EGuard* to return a result (lines 10-15). When there is no destination information (DNS name or dotted IP address) in the visual link (category 5), EGuard calls *AnalyzeDNS* to analyze the actual dns (lines 18and 19). EGuard therefore handles all the 5 categories of phishing attacks. *AnalyzeDNS* and the related subroutines are depicted in Fig. 6. In *AnalyzeDNS*, if the actual dns name is contained in the blacklist, then we are sure that it is a phishing attack (lines 20 and 21). Similarly, if the actual dns is contained in the whitelist, it is therefore not a phishing attack (lines 22 and 23). If the actual DNS is not contained in either whitelist, blacklist or graylist *PatternMatching* is then invoked (line 25).

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 he 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 34 and 35), and if it is quite similar (but not identical) with one or more names in the *seed set* by invoking the *Similarity* (lines 38-43) procedure.

Similarity checks the maximum likelihood of actual dns and the DNS names in *seed set*. As depicted in Fig. 6, 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 'windows' and 'wind0ws' is 6/7 (since we need change the 1 '0's in wind0ws 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)

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-commence 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)!

False positives and false negatives handling

Since EGuard 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 non-phishing site). In what follows, we show that EGuard 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 are 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 EGuard handles categories 3 and 4 correctly since the encoded links are first decoded before further analysis. For category 2 also EGuard reduced the false positive upto negligible rate due to the graylist usage. For category 5, EGuard may result in false positives. For example, we know that both 'www.iee.org' (respected site of electrical engineers and 'www.ieee.org'(electrical and electronics engineers site) 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, EGuard 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. For category 5, EGuard 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, EGuard 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

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). We are trying our best to get it also solved.

VI.IMPLEMENTATION AND VERIFICATION OF EGUARD

We have implemented the EGuard algorithm in Windows XP. It is similar to the implementation of the LinkGuard but here in EGuard, the database in the EGuard executive consists graylist along with blacklist and whitelist.

It includes two parts: a whook.dll dynamic library and an EGuard executive. The structure of the implementation is depicted in Fig. 7

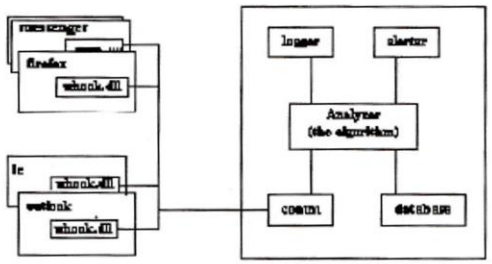

Fig.7 Architecture diagram of EGuard

The structure of the EGuard implementation, which consists of a whook.dll and a EGuard executive. whook is a dynamic link library, it is dynamically loaded into the address spaces of the executing processes by the operating system. whook is responsible for collecting data, such as the called links and visual links, the user input URLs. More specifically, whook.dll is used to: 1) install a BHO (browser helper object) for IE to monitor user input URLs; 2) install an event hook with the *SetWinEventHook* provided by the Windows operating system to collect relevant information; 3) retrieve sender's e-mail address from Outlook; 4) analyze and filter the received windows and browser events passed by the BHO and the hook, and pass the analyzed data to the EGuard executive.

EGuard is the key component of the implementation. It is a stand alone windows program with GUI (graphic user interface). It's composed of 5 parts as illustrated in Fig. 7:

Analyzer, Alerter, Logger, Comm, and Database. The functionalities of these 5 parts are given below:

Comm: Communicate with the whook.dll of all of the monitored processes, collect data related to user input from other processes (e.g. IE, outlook, firefox, etc.), and send these data to the Analyzer, it can also send commands (such as block the phishing sites) from the EGuard executive to whook.dll. The communication between the EGuard process and other processes is realized by the shared memory mechanism provided by the operating system.

Database: Store the whitelist, blacklist, graylist and the user input URLs.

Analyzer: It is the key component of EGuard, which implements the EGuard algorithm,. It uses data provided by Comm and Database, and sends the results to the Alert and Logger modules.

Alerter: When receiving a 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. After implemented the EGuard system, we have designed experiments to verify the effectiveness of our algorithm.

RESULTS

Since we are interested in testing EGuard's ability to detect unknown phishing attacks, we set whitelist, blacklist and graylist to empty in our experiments. Our experiments showed that EGuard can detect about 96% phishing attacks of the APWG archives. Our experiment also showed that our implementation uses small amount of CPU time and memory space of the system. In a computer with 1.6G Pentium CPU and 512MB memory, our implementation consumes less than 1% CPU time and its memory footprint is less than 7MB. Our experiment only used the phishing archive provided by APWG. We are planning to use EGuard in daily life to further evaluate and validate its effectiveness. Since we believe that a hybrid approach may be more effective for phishing defense, we are also planning to include a mechanism to update the blacklist and whitelist in real-time and also implementing graylist in its fullness.

VII. CONCLUSION

It is becoming increasingly common to tune in to the news or load your favorite news Web site and read about yet another Internet e-mail scam. So 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, EGuard, based on the derived characteristics. Since Phishing-Guard is characteristic based, it can not only detect known attacks, but also is effective to the unknown ones. We have implemented EGuard for Windows XP. Our experiment showed that EGuard is light-weighted and can detect up to 96% unknown Phishing attacks in real-time.

We believe that the EGuard will be useful for detecting Phishing attacks, and also can shield users from malicious or unsolicited links in Web pages and Instant messages

REFERENCES

[1] I. 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/.

[3] 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.

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

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

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

[7] John Leyden. Trusted search software labels fraud site as 'safe'.