2009 33rd Annual IEEE International Computer Software and Applications Conference

PhishCatch – A Phishing Detection Tool

Weider D. Yu Shruti Nargundkar Nagapriya Tiruthani

Computer Engineering Department, San Jose State University
San Jose (Silicon Valley), California, USA 95192-0180

Abstract

Phishing has become the most popular practice among the criminals of the Web. Phishing attacks are becoming more frequent and sophisticated. The impact of phishing is drastic and significant since it can involve the risk of identity theft and financial losses. This paper explains the most popular methods used for phishing and the PhishCatch algorithm developed to detect phishing. The PhishCatch algorithm is a heuristic based algorithm which will detect phishing emails and alert the users about the phishing emails. The phishing filters and rules in the algorithm are formulated after extensive research of phishing methodologies and tactics. After testing, we determined that PhishCatch algorithm has a catch rate of 80% and an accuracy of 99%. The approach used in developing this algorithm, the implementation details and testing results are discussed in this paper.

1. Introduction

Phishing is an attempt by an individual or a group to harvest personal confidential information such as user names, passwords, credit card information, etc., from unsuspecting victims for identity theft, financial gain and other fraudulent activities. A phishing scam usually involves the phisher sending mass emails to intended victims and users replying to e-mails and in the process, divulging their personal information. But such scams have drastically reduced in recent times and recent scams involve the use of latest technologies to lure the victim to give away his sensitive personal information. Fake websites which appear very much similar to the original ones are being hosted to achieve this. Thus the users assume that they are entering information into a genuine website without realizing that they are giving away their precious information to a stranger who can misuse it for financial gains.

While most phishing attacks target the financial industry, more and more phishing incidents targeting other sectors such as retailers, online game operators and large ISPs have also been discovered. As phishers have advanced their local language capabilities and the level of sophistication of attacks, phishing attacks have expanded geographically to various European and Asian countries. The following graphs (Figure 1) show the rise in the number of phishing attacks since July 2005 on a monthly basis [1].

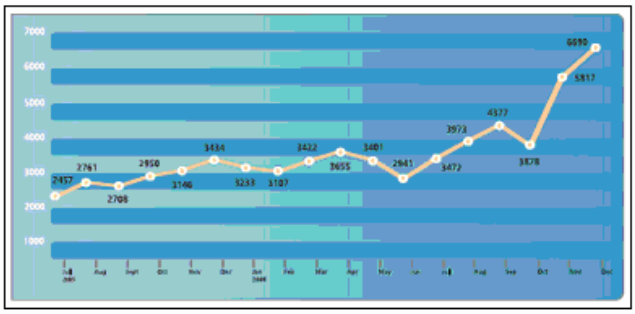

Figure 1: Phishing attacks per month – Worldwide [5]

According to Anti-Phishing Working Group, there were around 18000 unique phishing reports in 2006. The top countries hosting the phishing websites are China with 23.74 percent of the phishing sites, and closely followed by United States with 22.93 percent of the total phishing websites, China and the Republic of Korea [2].

Phishing attacks have been on a constant rise in the past year also. Phishing attacks in the United States soared in 2007 and $3.2 billion was lost to these attacks, according to a survey by Gartner, Inc. According to the survey, 3.6 million people lost money in phishing attacks in over the period of August 2006 to August 2007, as compared with the 2.3 million who did so the year before [3],[7].

The survey also predicts that the phishing and malware attacks will continue to rise through 2009 because it is still a very attractive business for phishers. Hence it becomes all the more important to devise effective anti phishing solutions to detect and prevent phishing.

In this paper, we study the common practices involved in phishing attacks and review some anti-phishing solutions. We eventually focus on an approach which we have developed to detect and prevent phishing. The rest of the paper is organized as follows: In section II we provide phishing email analysis. Section III discusses the phishing attack procedures, causes, motivations and prevention methods, while section IV describes in detail the PhishCatch algorithm. Section V presents the testing details and the test results of PhishCatch algorithm and Section VI concludes this paper.

0730-3157/09 $25.00 © 2009 IEEE
DOI 10.1109/COMPSAC.2009.175

2. Phishing Email Analysis

Looking at the fact that phishing scammers are reaping enormous financial gains, it can easily be concluded that the motivation behind phishing is almost always financial. Although financial gain is the major motivating factor for phishing, other factors such as identity theft, industrial espionage, malware distribution, etc., are the other motivating factors for phishers. A root cause analysis was done to identify the motivation for phishing and the following factors were identified:

- Financial Gain – almost always the most important motivating factor for phishing.
- Identity Theft – Stolen identities from unsuspecting victims can be used by the phisher for financial gain, for criminal activities, to commit fraud or to launch more phishing attacks by assuming the stolen identity. Sometime the stolen identities are sold to other interested parties for a premium. In recent times, identity theft is posing a serious threat to individuals since it can result in not just financial losses but also in other consequences like negative information about an individual, damage to reputation, etc.
- Identity Trafficking - Phishers indulge in identity theft and the stolen identities are sold to interested parties on online forums for a premium. These activities can be extremely difficult to track and identify since they can be spread across international boundaries.
- Industrial Espionage - Highly sophisticated phishing attacks are being launched against a victim to spy on the victim and to get comprehensive information about the victims browsing patterns, product loyalties, etc. These details are used by the phishers directly or are being sold to interested parties. Using this information, the victim can be targeted to make him shift loyalties from one product to another and to tarnish a brand. The monetary losses due to industrial espionage run into billions of dollars.
- Malware Distribution - Phishing attacks can be launched with the intent of distributing malware i.e., malicious software. Phishing emails are usually sent in bulk and hence, zombie networks are the best suited to launch large phishing attacks. Phishers send unsolicited phishing emails with malware attachments so that when an unsuspecting user clicks to open the attachment, the malware is installed on the victims machine converting the victims machine into a zombie. The phisher may distribute software such as Trojans, key loggers, browser overlays, fake browsers, etc., on machines for use in later scams such as harvesting further information as users unknowingly enter information into infected machines over weeks or months.
- Password Harvesting - Password harvesting is done by phishers using various methods like key loggers and other malware. The harvested user information is used again for financial gain, fraud, identity theft or sold to interested parties for financial gain.
- Fame and Notoriety - Sometimes phishing attacks are carried out by people mainly to gain recognition and notoriety among their peers. This is a very interesting psychological aspect of phishing wherein information is phished only for the purpose of gaining attention and glory in the online community.
- Exploit Security Holes - Hackers search for security flaws and find methods to exploit them to launch phishing attacks or to sell the compromised systems to other phishers. This is again motivated by financial gain and also by the prospect of gaining fame and notoriety.

The causes for phishing vulnerability can be summarized into the following categories.

- Weak Authentication Schemes – many websites and mail servers have weak authentication schemes which make it easier for fraudsters to launch phishing attacks. Insufficient use of digital signatures for authentication makes application more susceptible to phishing attacks.
- Browser vulnerabilities – attackers use browser vulnerabilities like address bar spoofing, cross site scripting, HTML frame injection, script injection, browser proxy configuration and multimedia auto play and auto execute extensions to launch phishing attacks.
- Security Flaws – like port redirection, man in the middle attack, session hijacking and client side vulnerability exploitation which are very difficult to detect, are used to launch phishing attacks
- Non secure desktop tools – like inefficient anti phishing toolbars, antivirus, spam filters, pop-up blockers, firewalls and spyware detectors make it easier for a phishing attack to succeed.
- Lack of user awareness
- Ease of impersonating a trusted source

The most commonly used techniques to launch a phishing attack are explained as follows:

- Exploit user willingness trust and fear by creating a sense of urgency and fear in the user
- Impersonate a trusted source by address spoofing

- Exploit weak authentication schemes of the websites and mail servers
- Address bar spoofing, URL spoofing and obfuscation
- Cross site scripting
- Script injection and HTML frame injection
- Installing Malware like keyloggers, phishing kits, rootkits and backdoors
- Exploit security flaws like port redirection, session hijacking and man in the middle attacks
- Using botnets and zombies to launch attacks
- DNS cache poisoning

Conventionally however, most phishing emails are sent asking the user to click on a hyperlink. After extensive analysis of hundreds of phishing emails and the methodologies used in phishing, phishing hyperlinks were categorized into the following general categories:

1) The actual link and the visual link in the email are different i.e., the hyperlink in the email does not point to the same location as the apparent hyperlink displayed to the users
2) The DNS name in the hyperlink is substituted by the quad-tuple IP address
3) DNS names used are manipulated to look similar to the genuine DNS name the phishers are trying to forge
4) The hyperlink is encoded so that it becomes very difficult to read for example, unusually long hyperlinks
5) When visiting the phishing hyperlink, it usually asks the user for various personal details like username, password, account number, SSN, etc.

Based on the extensive analysis conducted on the motivation, causes and techniques used in phishing and analysis conducted on the phishing emails, we propose an algorithm - PhishCatch - Detect, Defend and Deter. The focus of this algorithm will be to detect phishing emails and alert the user about phishing emails. Using the hyperlinks collected from the phishing emails, the algorithm will further gather valuable information about phishing hyperlinks like the country of origin of the phishing attack, the brand being phished, etc., and populate the details into a data warehouse. Also collected will be the characteristics of phishing emails, for example the subject of the email, which will be stored into the data warehouse. This data repository will be a valuable source of information to derive the latest trends in phishing and also to analyze the changing methods of phishing attacks. This can help in preventing further phishing attacks.

3. PhishCatch Algorithm

PhishCatch is a heuristic based algorithm that relies on a set of phishing rules to classify phishing emails. These phishing rules are formulated based on detailed analysis of phishing emails and various phishing methodologies. After extensive analysis of phishing emails we have formulated certain categories under which the phishing emails can be classified. A unique filter is associated with each category and rules are formulated using the various combinations of these filters. Further, each filter is associated with a certain weight which we have derived based on the significance of each filter and also by regression testing of the algorithm. The weight associated with each filter plays a very important role in determining whether an email is a phishing email or not, as explained in the following sections. Each rule has been tested extensively for effectiveness and finally we have derived an optimum set of rules which will bring about maximum efficiency and lower the number of false positives. The various filters used in the algorithm are discussed in the next section.

3.1 PhishCatch Implementation

We have implemented the PhishCatch algorithm in Windows XP. The programming language used for implementation is python and regular expressions are used extensively for filtering the emails. The block diagram in Figure 2 gives the overall picture of PhishCatch.

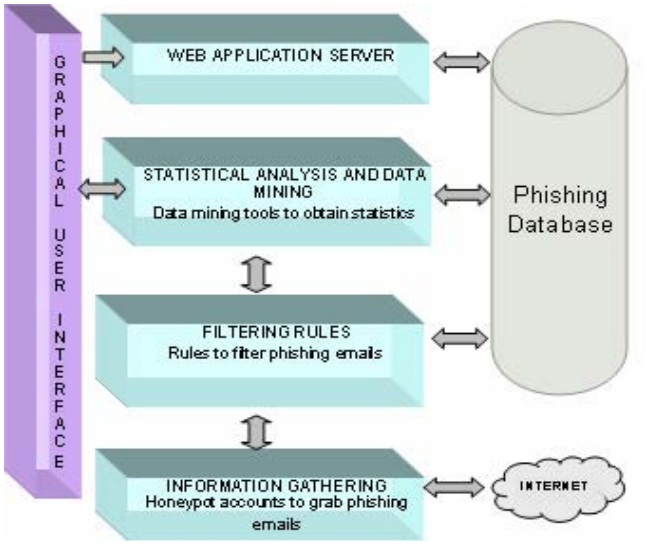

Figure 2: Block Diagram of PhishCatch algorithm implementation

The basic architecture of PhishCatch consists of a module to fetch emails, a module to filter emails and classify them as phishing, Alerter to issue an alert to the user and the data warehouse which will store all the information related to phishing emails. The functioning of the system will be explained in the following sections.

4. The PhishCatch Algorithm – To Detect, Defend and Deter

The PhishCatch algorithm is explained below.

```
getMail(SMTPserver,username,password)

if new mail found
    for each message
        get recvdFrom from header
        get From from header
        if message encoded
              decode message

        check for text_filters match
              set Phishrank[text_filters]

        if recvdFrom != From
             set Phishrank[received_mismatch]

       find_links in each_email
       if link found
              linkCharacteristics(link)
                compare anchor _tags
              check whois(recvfrom IP)
              check phishtank(phishing_link)
              check whois(phishing_link)

      if isPhishing
              Alert the user
              Connect to database
              Insert the phishing email details in db

….go to next email
```

Figure 3: Pseudocode of the PhishCatch Algorithm

```
def decode_message()
     Remove HTML encoding

 def linkcharacteristics(link)
    if special characteristics  found in link
        add link to phishingLink[]
     else if IP found in link
        add to phishrank [dotted_quad]
        add link to phishingLink[]
     else if no_of_folders in link > folder_threshold
        add to phishrank[no_of_folders]
        add link to phishingLink[]
     else if no_of_subdomain in link > domain_threshold
        add to phishrank[no_of_subdomain]
        add link to phishingLink[]
     else if len(link) > link_threshold
        add to phishrank[len_of_link]
        add link to phishingLink[]

def whois(link)
     if link == IP address
          contact appropriate whois server
          lookup IP
     else
          get the host name
          contact appropriate whois server
          lookup host

def phishtank(phishing_link)
     search phishing_link in phishtank
     return found/ not found

def isphishing()
     check phishrules matrix for phishing
          return true if phishing
```

Figure 4: Pseudocode of the subroutines of PhishCatch

In PhishCatch algorithm we first fetch the new e-mails from the SMTP server. The algorithm is designed to work with POP and IMAP mail servers. When any new email comes in, the email is retrieved and split up into headers and body. The body of an email is sometimes, HTML encoded and the type of encoding is indicated in the "Content-type" field in the email header. If the email is encoded, we decode it so that the phishing filters work correctly with the email. Once the email is retrieved and stripped into its component parts, the next step in the algorithm is to apply the phishing filters on the email to detect a phishing email. Firstly, the email is scanned for the presence of the text filters defined in the algorithm. The number of text filters detected in the email is recorded, which will be the weight of that filter. The weight of the filter is added to a list, Phishrank. Phishrank is a list which contains a mapping of the phishing filters to their respective weights.

In the next step, the received domain mismatch is checked in the email i.e., the domain similarity between the Received from and From fields in the email is verified. The first Received From and the From fields are obtained from the e-mail header. If both these fields do not have the same domain, then we can safely assume that the source address was spoofed in the email and hence the appropriate weight is assigned to the received domain mismatch filter.

Next we look for all the available hyperlinks in the e-mail. If a link is found, then the link is run against the linkCharacteristics() function to scan for any possible misrepresentation of the link and hence to check for link encoding. If there is any misrepresentation noticed in the link, then the appropriate weight is assigned to the link encoding filter. The length of the link is also checked and if it exceeds a certain predefined threshold for link length, then based on the length of the link, the appropriate weight is set for the length of link filter. Similarly, the number of folders and the number of sub domains in the hyperlinks are checked and the corresponding filter weights are set.

In the next step, the anchor tag for each hyperlink in the email is fetched from the source HTML code of the email. Each link is compared with its respective anchor tag to check for discrepancy, if any, between the visual link and the actual link. If there is a mismatch between the visual and the actual links, then the appropriate weight is assigned to the link mismatch filter.

Once the weights are assigned for all the filters, the PhishCatch rules, which are basically, verified combinations of the phishing filters and their weights, are referred to and the email is classified as a phishing email. Further, the phishing link is identified among all the hyperlinks found in the email using a ranking system to rank the hyperlinks. The principle behind the ranking system is that a rank is assigned to each link based on the probability of it being a phishing link. The probability is deduced by looking at which filters the phishing link triggers. The identified phishing link is stored and used for the information gathering and for cross verification with PhishTank data.

After this, the next steps in the algorithm will be information. Once the phishing link has been identified, we work on gathering more information about the phishing attack in general and the phishing link in particular. Using the phishing email we gather the general characteristics of the phishing email like the senders address, the subject of the phishing email, number of links present in the email, the phishing link, etc. The purpose for collecting this data is to analyze the types of phishing attacks happening and to project the types of attacks which can occur in the future. Using the email headers, we collect information like the IP address where the email originated, size of the email, the IP address where the phishing website is located, etc. Further, using the IP address, more information about the host and the phishing website like the organization name owning the website, the geographical location of the host/ website, etc., is fetched using Whois database. A separate module which performs the whois lookup and the IP to country mapping is referred to perform this function. All this information is stored and updated in the data warehouse.

As and when the phishing email is detected, the alerter module will issue an audible alert to the user warning the user not to click on any links in the email.

5. Testing Results

Extensive testing of PhishCatch algorithm was done to test for the efficiency of the algorithm in detecting phishing emails. For this purpose, honeypot accounts were set up and seeded over the internet to attract phishing emails. PhishCatch was tested against these email ID's. More than 2000 phishing e-mails were also obtained from the online phishing corpus available at the website **http://monkey.org/~jose/wiki/doku.php?id=PhishingCorpus.** This phishing corpus contains phishing emails over the period of August 2005 to August 2007 and the honeypot email ID's attract the most current types of phishing attacks. Overall, about 5000 emails were tested against PhishCatch. The results of the tests are as follows:

- Total Emails tested : 4804
- Phishing Emails Present : 3710 (A)
- Phishing Emails Detected : 2967 (B)
- Effectiveness for PhishCatch = B / A = 0.7997

PhishCatch Effectiveness: 80%

- Legitimate Non-Phishing Emails Present : 1094 (C)
- Legitimate Non-Phishing Emails Detected by PhishCatch : 11 (D)

Hence, PhishCatch accuracy = (C – D) / C = 0.9899

PhishCatch Accuracy: 99%

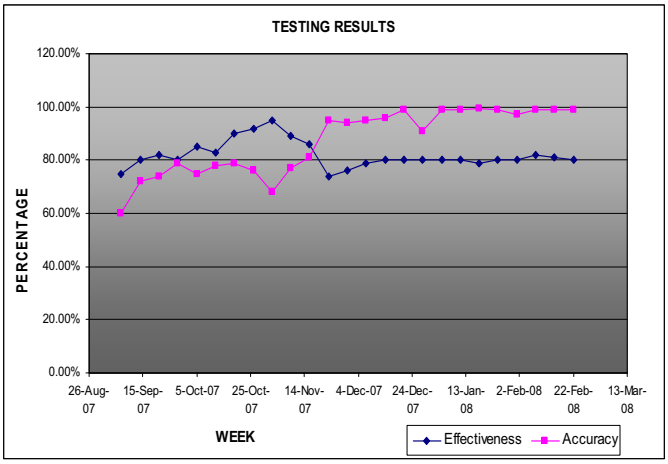

Figure 5: Chart showing the test results over a period of testing for PhishCatch

Based on the tests conducted, we observe that PhishCatch's correctness is evaluated to be 99% i.e., PhishCatch incorrectly classified only 1% of legitimate emails as phishing emails. PhishCatch's catch rate is 80% i.e., PhishCatch can classify 80% of the phishing emails correctly. The remaining 20% of the emails which PhishCatch did not detect were manually analyzed. All these emails belong to the phishing corpus and are a mix of other language phishing emails, emails which do not have a body, emails whose HTML source page is not correctly formatted and some emails of the Nigerian 419 scam. Since PhishCatch does not provide other language support, some of these emails are not being detected. Also, since the 419 scam emails are not computer generated and do not usually contain a link, these are not detected by PhishCatch. We are continually testing PhishCatch against the emails obtained in the honeypot accounts and fine tuning our phishing rules to increase the catch rate of PhishCatch at the same time, keeping the accuracy constant or better. PhishCatch can be further enhanced to provide support for other languages.

6. Cross Testing against PhishTank for Effectiveness.

PhishTank is a collaborative clearing house for data and information about phishing on the Internet [2],[4].

PhishTank is a publicly available phishing database that receives phishing links from users. These links are voted upon by the users and based on the number of votes the links receive, they are classified as phishing links or not. Popular web browsers like Mozilla Firefox use PhishTank data to detect phishing links and alert the user about the phishing link.

A few noticeable differences between PhishTank and PhishCatch are:

- PhishTank was started in October 2006 and in community based while PhishCatch was implemented in November 2007 and is algorithm based [3],[4].
- PhishTank is a community based website wherein a link is classified as phishing based on the number of votes it receives from the user while PhishCatch is a heuristic based algorithm which is independent of user input for classification of phishing links.
- For PhishTank to classify a link as phishing, it usually takes a few days time as it is user based unlike PhishCatch which can detect a link as phishing at runtime and alert the user as soon as the phishing email arrives in the user's inbox.

The phishing links detected by PhishCatch were compared with the phishing links existing in PhishTank and the following statistics were obtained.

- Total links tested : 2048
- PhishCatch links found in PhishTank : 502
- PhishCatch links not found in PhishTank : 1546

Figure 5 below shows the graphical representation of the numbers given above.

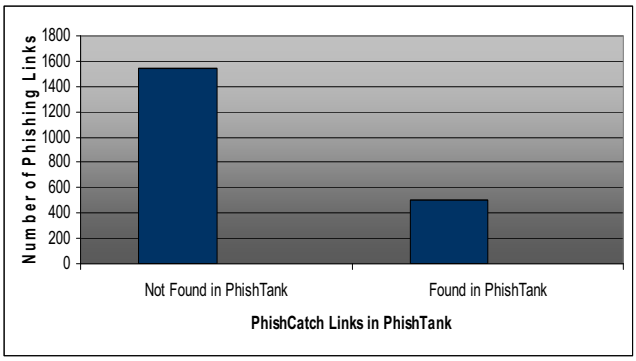

Figure 6: Chart showing the phishing links caught by PhishCatch that were present and not present in PhishTank

7. Conclusions

In this paper we have analyzed the various types of phishing attacks and have designed an algorithm PhishCatch to detect phishing. PhishCatch is a heuristic algorithm that is focused on detecting phishing links, alerting the user about a suspected phishing link and building an extensive data warehouse containing a wealth of information about phishing. This data warehouse can be further used to analyze trends in phishing and to derive statistics about phishing. PhishCatch is a very lightweight algorithm requiring very less memory and CPU time and has a catch rate of 80% and an accuracy of 99%. In the future, PhishCatch algorithm is going to be enhanced to strip and look into attachments. Our further work will be to enhance the capability of PhishCatch to interact with other mail servers like Microsoft Outlook and also to fine tune the phishing rules to enhance the catch rate and hence the efficiency of PhishCatch.

8. References

[1] Antiphishing.org Report, http://www.antiphishing.org/reports/APWG_MemoOnDomainWhoisTake-Downs.pdf

[2] Latest Antiphishing.org Report http://www.antiphishing.org/reports/apwg_report_july_2007.pdf

[3] Gartner Survey Shows Phishing Attacks Escalated in 2007; "More than $3 Billion Lost to These Attacks". Accessed: December 18, 2007 http://www.gartner.com/it/page.jsp?id=565125

[4] http://www.phishtank.com/

[5] Cranor, L., Egelman, S., Hong, J., Zhang, Y. "Phinding Phish: An Evaluation of Anti-Phishing Toolbars", November 13, 2006; CMU-CyLab-06-018; Carnegie Mellon University

[6] http://anti-phishing.org/sponsors_technical_papers/rsaPHISH2_WP_0107.pdf

[7] Symantec Internet Security Threat Report: "Trends for July 2006 – December 2006".