Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 49 (2015) 82 – 91

Procedia
Computer Science

ICAC3'15

AN IDEAL APPROACH FOR DETECTION AND PREVENTION OF PHISHING ATTACKS

Narendra. M. Shekokar, Chaitali Shah, Mrunal Mahajan,Shruti Rachh*

D. J. Sanghvi College of Engineering, Vile Parle(W), Mumbai:400056, India
D. J. Sanghvi College of Engineering, Vile Parle(W), Mumbai:400056, India
D. J. Sanghvi College of Engineering, Vile Parle(W), Mumbai:400056, India

Abstract

Phishing is a treacherous attempt to embezzle personal information such as bank account details, credit card information, social security number, employment details, and online shopping account passwords and so on from internet users. Phishing, or stealing of sensitive information on the web, has dealt a major blow to Internet security in recent times. These attacks use spurious emails or websites designed to fool users into divulging personal financial data by emulating the trusted brands of well-known banks, e-commerce and credit card companies.

In this paper, we propose a phishing detection and prevention approach combining URL-based and Webpage similarity based detection. URL-based phishing detection involves extraction of actual URL (to which the website is actually directed) and the visual URL (which is visible to the user). LinkGuard Algorithm is used to analyze the two URLs and finally depending on the result produced by the algorithm the procedure proceeds to the next phase. If phishing is not detected or Phishing possibility is predicted in URL-based detection, the algorithm proceeds to the visual similarity based detection. A novel technique to visually compare a suspicious page with the legitimate one is presented.

Peer-review under responsibility of organizing committee of the 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15)

Keywords: Phishing; LinkGuard Algorithm;Discrete Cosine Transform (DCT)

1. Introduction

Phishing is an online deceitful activity wherein the objective of an attacker is to plagiarize a victim's sensitive information, such as online banking account details or social security number thus deceiving people into financial loss. Even though hoaxing people to make financial profit is an old idea, phishers have realized that social-engineering based attacks are easy to execute and highly profitable over the Internet.

* ShrutiRachh. Tel: +919757244133; Fax: -
E-mail Address: rachhshruts@gmail.com

Peer-review under responsibility of organizing committee of the 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15)
doi:10.1016/j.procs.2015.04.230

A typical phishing attack may be based on several techniques, including exploiting browser vulnerabilities or performing man-in-the-middle attacks using a proxy. However, the most straightforward and widespread method includes setting up a web page that is similar to the one which is known to the user.

Therefore, although well known, phishing still poses a significant security threat and still a large number ofInternet users fall victim to this fraud. Furthermore, such attacks are not just causing troubles for Internet users, but also for companies that provide financial services online. This is because when users fall a prey to such phishing attacks, the organization providing the online service often suffers a loss in reputation as well as financial damage.

2. Phishing attack procedure and prevention methods

In this paper, we will consider methods to detect phishing that uses emails since phishers mostly use them to defraud the victims. The method is explained below:

1) Phishers set up a phony Web site which looks identical to the legitimate Web site, including page layouts, styles(font families, sizes and so on), key regions, setting up the web server and applying the DNS server name.
2) They send a huge number of fake emails to various users by spoofing as legitimate companies and organizations, trying to lure the potential victims to visit their Web sites.
3) Victims who receive such emails, opens them, clicks on the hyperlink in the email which leads them to fake website created by the phisher, wherein they give in their significant personal information such as bank account passwords, credit card details and so on.
4) Phishers embezzle such personal information and uses it for their own benefit such as stealing money from other people's accounts.

As per a study, it was found that 40% of the times, Internet users ignored browser-based cues such as the address bar and the security indicators. Some counterfeit websites are so similar to the legitimate websites that can fool even the most sophisticated users. As standard security indicators are not effective in preventing a large number of users from falling a victim to such phishing attacks, alternate approaches to avoid such attacks are needed.

3. Related Work

Phishing is a growing problem on the internet today for both consumers and businesses. One of the most common approaches for an attacker is to create a similar website in order to capture personal information from consumers. A malicious website may look identical to an online bank or other financial institution in order to capture passwords, social security numbers, account numbers, and other confidential information. A victim may not identify the malicious site until after the confidential information has been leaked.
Some of the approaches for phishing detection are:

3.1. Email-level approach

This approach intends to amend the phishing attacksat the email level. The main concept is that when a spoofed email is not received by its victims, they cannot fall for the scam. Filters and content analysis techniques are often used to detectphishing emails before they can be delivered to users.For instance, by using training filters (e.g., Bayesianfilters), an enormous number of phishing emails can be thwarted.

In order to prevent spoofing of sender information in an email message, Microsoft and Yahoo have defined email authentication protocols (Sender ID and DomainKeys) that can be used to verify the credibility of a received email. If widely used, these solutions could help to prevent spam emails and, as a result, decrease the number of email-based phishing attacks.

3.2. Browser-integrated tool approach

These toolsdetect phishing by comparing the web page link in the address bar with the list of malicious site URLs mentionedin a blacklist.For example, the address bar turns red in Microsoft Internet Explorer (IE) 7 when a malicious page loads. Well-known, academic, browser-integrated solutions to slacken phishing attacks are SpoofGuard and PwdHash. SpoofGuard works by analysingfor phishing symptoms such as obfuscated URLS in web pages. On the other hand, PwdHash generates domain-specific passwords that are rendered ineffectual when submitted to another domain.

3.3. Webpage content analysis

It analyzes a Web page's content, such as the HTML code, text, images, input fields, forms and hyperlinks. Earlier, such content based approaches proved effective in detecting phishing pages. But recently, phishers have started creating web pages with non-HTML components, such as Flash objects, images and Java applets. For instance, a phisher might design a fake page that consists entirely of images, even if the original page contains only text information. In this case, content-based anti-phishing tools cannotanalyze the suspicious webpage because its HTML code contains nothing but HTML <img/>elements.

3.4. Visual similarity based approach

Liu et al.'s short paper suggests that authors define metrics by analysing and comparing legitimate and phishing web pageswhich can be used to detect a phishing page. The idea is to first disintegrate the web pages into pertinent blocks according to "visual cues." Then, based on the defined metrics, similarity between two web pages is determined. If the resemblance to the legitimate web page is above the predefined threshold, then the web page is considered as a phishing page.

4. Existing system

Phishing has become a severe problem in the Internet society. Researchers have developed models and guidelines for supporting online consumer trust. Existing literature deals with trustworthiness ofwebsite interface designs and policies,website content and methods to foster customer relations.

Table 1.Drawbacks in existing systems.

Techniques	False positives	Zero day attacks	Fake interface attack	Slow response time
Blacklist	No	Yes	No	No
Heuristics	Yes	Maybe	No	Maybe
User polling	Yes	Yes	Yes	Maybe
Third-party certification	No	No	Yes	Maybe

The techniques are described in detail below:

4.1. Blacklist check

The suspicious URL is compared with a list of malicious website links. This method is vulnerable to "zero day attacks". In addition,this method does not work when techniques like routing through alternate domain nameand URL obfuscation are employed.

4.2. Heuristics

It uses heuristics like domain registration information (owner, age, and country), the number of links to other known-good sites, image hashing, third-party cookies and user reviews. Most of the heuristics used are intuitive and leads to a large number of false positives.

4.3. User polling

It deems the URL as phished, based on user votes. However, it is ineffective against new phishing attacks and is very subjective. Our solution does not incorporate any kind of polling, thus reducing uncertainty.

4.4. Working with third-party certification authorities and reputation services

It requires an additional interface, which itself is susceptible to phishing. Phish detection in our solution is handled completely on the server side, without involving any third party service. Another technique is to use page rank methodology, domain analysis, URL type analysis, and word analysis, in order to detect a phishing URL. However, false positives have been observed in these methods.

5. Our approach

In the following subsection, we provide an overview of how our system can be used to detect phishing. Our system proposes a scheme for phishing page detection based on two phases:

5.1. URL and DomainIdentity Verification[3]

Normally, phishing is done via sending mails to thousands of users, urging them to visit the fake website through the link or *URL* present in it. In order to embezzle sensitive information from potential victims, phishers generally try to persuade the users to click on the hyperlink embedded in the spoofed email. A hyperlink has a structure as follows:

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

We have classified the hyperlinks used in the phishing email into the following categories:

1) The hyperlink provides DNS domain names in the anchor text, but the destination DNS name in the visible link does not match as that in the actual link. For instance, the following hyperlink:

   <ahref=http://www.profusenet.net/checksession.php> https://secure.regionset.com/EBanking/logon/</a>

   The above hyperlink 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) Dotteddecimal IP address is used directly in the URI or the anchor text instead of DNS name. For example,

   <a href=
   http://61.129.33.105/securedsite/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. Forexample,

<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 seemed to point to 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 email 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:fvthljhfcs83infoupdate@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:// 200.251.251.11/.verified /”> Click here <a>

Once the user clicks on the above link, it will redirect to the phishing site 200.251.251.11 due to a vulnerability of usa.visa.com.

A phishing hyperlink can belong to several categories at the same time. For instance, an attacker may use tricks from both categories first and third at the same time to increase his success chance.

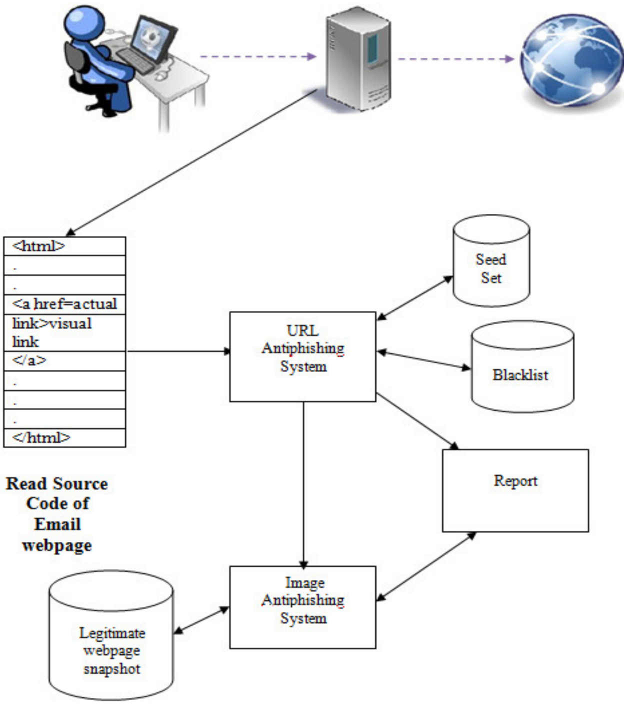
<html>
.
.
<a href=actual link>visual link
</a>
.
.
.
</html>
Read Source Code of Email webpage
URL Antiphishing System
Seed Set
Blacklist
Report
Image Antiphishing System
Legitimate webpage snapshot

Fig. 1. System Architecture

5.1.1. 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 following terminologies are used in the algorithm.

Nomenclature	
v_link	Visual linki.e. the link that is seen by the user
a_link	Actual link i.e. the link to which user is redirected when clicked
v_dns	Visual Domain Name System (DNS) name
a_dns	Actual DNS name

```
intLinkGuard (v_link, a_link)
Step 1: v_dns = GetDNSName(v_link);
Step 2:  a_dns = GetDNSName(a_link);
Step 3:  if v_dns and a_dns are not empty and v_dns!=a_dns
    Step 3. 1.  return PHISHING and goto end
Step 4:  if a_dns is dotted decimal
            Step 4.1.return POSSIBLE_PHISHING and goto next phase image based webpage matching
Step 5: if a_linkor v_linkis encoded
    Step 5. 1.  v_link2= decode (v_link);
    Step 5. 2. a_link2= decode (a_link);
    Step 5. 3.  returnLinkGuard(v_link2, a_link2);
 /* Analyze the domain name for possible phishing */
Step 6:  if v_dnsis NULL
    Step 6. 1. returnAnalyzeDNS(a_link);
Step 7: end
```

The *LinkGuard*algorithm works as follows. In its main routine *LinkGuard*, it first extracts the DNS names from the actual and the visual links (steps 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 (step 3). If dotted decimal IP address is directly used in actual dns, it is then a possible phishing attack of category 2 (step 4). If the actual link or the visual link is encoded (category 3), we first decode the links, then recursively call *LinkGuard*to return a result (step 5). When there is no destination information (DNS name or dotted IP address) in the visual link (categories 4 and 5), LinkGuard calls *AnalyzeDNS*to analyze the actual dns (step 6). LinkGuard therefore handles all the 5 categories of phishing attacks.

```
intAnalyzeDNS (actual_link)
/* Analyze the actual DNS name according to the blacklist and whitelist*/
Step 1: if actual_dns in blacklist
    Step 1. 1. return PHISHING goto end
Step 2: if actual_dns in whitelist
    Step 2. 1. returnNOTPHISHING and goto next phase image based webpage matching
Step 3:  return PatternMatching(actual_link);
Step 4: end
```

```
intPatternMatching(actual_link)
Step 1: for each item prev_dns in seed_set
    Step 1. 1. bv = Similarity(prev_dns, actual_link);
    Step 1. 2. if (bv == true)
        Step 1. 1. 1. return POSSIBLE_PHISHING and goto next phase image based webpage   matching
    Step 1. 3. return NO_PHISHING and goto end
Step 2: end
```

```
float Similarity (str, actual_link)
Step 1:  (str is part of actual_link)
    Step 1. 1. return true;
Step 2: intmaxlen = the maximum string lengths of str and actual_dns;
Step 3: intminchange = the minimum number of changes needed to transform str to actual_dns (or vice versa);
Step 4: if (thresh<(maxlen-minchange)/maxlen<1)
    Step 4. 1.  return true
Step 5: return false;
```

Step 6: end

In *AnalyzeDNS*, if the actual dns name is contained in the blacklist, then we are sure that it is a phishing attack (step 1). Similarly, if the actual dns is contained in the whitelist, it is therefore not a phishing attack (step 2). If the actual dns is not contained in either whitelist or blacklist, *PatternMatching*is then invoked (step 3).

*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 proactively collect DNS names that are manually input by the user when user 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 it is quite similar (but not identical) with one or more names in the *seed set* by invoking the *Similarity* procedure.

Similarity checks the maximum likelihood of actual dns and the DNS names in *seed set*. The similarity index between two strings is 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).

Thus, LinkGuard algorithm analyzes the differences between the visual link and the actual link and also calculates the similarities of a URI with a known trusted site. If this algorithm detects possible or no phishing, then we perform the next phase i.e. image based webpage matching.

5.2. Image-based web page matching

We present a novel technique for comparing a suspicious web page snapshot with the legitimate one. The objective is to compute how similar the two web pages are.

An important feature of a phishing webpage is its visual similarity to its target (true) webpage. Hence, a legitimate webpage owner or its agent can detect suspicious URLs and compare the corresponding web pages with the true one in visual aspects. If the visual similarity of a webpage to the true webpage is high, the owner will be alerted and can then take whatever actions to immediately prevent potential phishing attacks and hence protect its brand and reputation. Image based webpage matching is done as follows:

We identify and consider the overall appearance of the entire page of the original website as well as the suspicious website in order to determine the similarity between them.

5.2.1. Take the snapshot of the suspicious page and the original page

This will either be done by using a browser plug-in for webpage snapshot or a code developed by us. This piece of code will take snapshot of the entire webpage.

5.2.2. Compute Discrete Cosine Transform (DCT)

To compare these two snapshots, first we need to transform the image so that fewer comparisons are required. There are various transforms like DCT (Discrete Cosine Transform), DFT (Discrete Fourier Transform) and other techniques like Cross-correlation which can be used. We have chosen DCT as the initial transform to be performed on the images because of the following advantages over other transforms:

- DCT is an accurate technique for image transformation.
- Time complexity of DFT is $O(N^2)$ while Time Complexity of DCT is $O(N \log_2 N)$.
- Cross-correlation is computationally complex and will take more number of operations to come to a result.

5.2.2.1. DCT

The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality)[6] i.e. the more important parts appear in the upper left corner of DCT matrix.

The 2D DCT can be computed using the one dimensional DCT horizontally and then vertically across the signal because DCT is a separable function. The 2D DCT of an image of M*N samples of a 2D signal f (x, y) is formulated as:

$$F(u,v)=2/(MN)^{1/2}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)u}{2M}\right](2)$$

For u=0, 1, …, N-1 and v=0, 1, …, M-1, where

C(k) = 1/√2for k=0

= 1 otherwise

To compute DCT of a color image, we need to separate the RGB components of the images. We compute DCT for R channel, B channel and G channel separately.

5.2.3. Compute RMS error

DCT of the individual R-channel, G-channel and B-channel of original webpage snapshot is compared with that of DCT of the corresponding RGB channels of the suspicious webpage to obtain Root Mean Square (RMS) error. If the RMS error is 0, the webpage is a legitimate one otherwise if it is below a certain pre-defined threshold then the webpage is a malicious one and warning is displayed to the user.

Thus, to summarize, we have proposed a phishing detection and prevention approach combining URL-based and Webpage similarity based detection. URL-based phishing detection is done using Linkguard Algorithm which is used to compare the two URLs. If phishing is not detected or Phishing possibility is predicted, the algorithm proceeds to visual similarity based detection where a novel technique to visually compare a suspicious page with the legitimate one is presented.

6. Performance analysis based on threshold

We have tested our Google Chrome extension for Gmail account users for 10 phishing websites and analysed its performance on the basis of false negatives for various thresholds selected and various changes made on those phishing websites. On the basis of various RMS (Root mean square) values obtained during testing we have plotted following graph of *Probability of False Negatives versus Threshold values.*

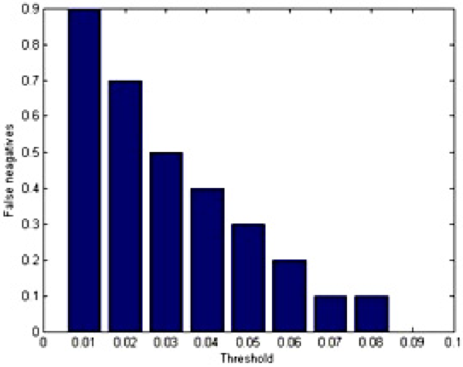

Fig. 2. Probability of false negatives versus threshold

7. Conclusion

In this paper, we have proposed a phishing detection approach that increases the webpage security by checking the hyperlinks in the source code of the email webpage and the overall appearance of the website. Our method serves as the robust alternative to many high cost web security applications.We expect the detection results to be comparableto previously published work which would allow for newkinds of phishing warnings with better coverage, less falsepositives and explicit user recommendations how to avoidthese critical situations.

Acknowledgements

We take immense pleasure in thanking respected Principal, Dr.HariVasudevan for having permitted us to carry out this project work and use college resources. We wish to express our deep sense of gratitude to teacher-in-charge Dr. N.M.Shekokar for allowing us to do the BE Project on this topic. His guidance and useful suggestions helped us in completing the project work on time. Finally, yet importantly, we would like to express our heartfelt thanks to our beloved parents for their blessings, our friends/classmates for their help and wishes for the successful completion of this project.

References

1. Eric Medvet, EnginKirda, EurecomFrance, “Visual-Similarity-Based Phishing Detection”, *SecureComm*, 2008.
2. R. Dhamija, J. D. Tygar, and M. Hearst, “Why Phishing Works”, In *Proceedings of the Conference on Human FactorsIn Computing Systems (CHI) 2006,* Montreal*, Canada*. ACM Press, 2006.
3. Juan Chen, ChuanxiongGuo, “Online Detection and Prevention of Phishing Attacks (Invited Paper)”, *National Natural Science Foundation of China (NSFC).*
4. Mona GhotaishAlkhozae, Omar Abdullah Batarfi, “Phishing Websites Detection based on Phishing Characteristics in the Webpage Source Code”, *International Journal of Information and Communication Technology Research*, Volume 1 No. 6, October 2011.
5. RadhaDamodaram, Dr. M. L. Valarmathi, “Phishing Detection based on Web Page Similarity”, *IJCST* Vol. 2, Issue 3, September 2011.
6. Ashok Banerji, Ananda Mohan Ghosh, ”Compression Technologies for Multimedia”, *Multimedia Technologies*, Tata McGraw Hill, p.76.