ISSN 2395-1621

Bait Alarm: Anti-Phishing using Visual Similarities

[#1]Pritesh Patil, [#2]Aditya Suryawanshi, [#3]Sanket Deshmukh,
[#4]Prof. Ms. Mansi Bhonsle

[1]priteshpatil08@gmail.com
[2]adityasuryawanshi224@gmail.com
[3]sanketdeshmukh5769@gmail.com

[#123]Student, Department of Computer Engineering
[#4]Prof, Department of Computer Engineering

G.H. Raisoni, College of Engineering and Management, Wagholi Pune.

ABSTRACT

The Phishing websites looks similar to the original one because of their appearances created by attacker or hacker and user can easily trapped into this by submitting their username and password into these fraud sites. In this research paper we developed a new algorithm called CSS algorithm. The CSS algorithm is used to detect URL phishing attacks and also it provides multilayered security to the fraud held on internet. The CSS algorithm can detect the CSS filename, CSS domain, CSS content of the file, URL of trusted site, The domain of the site, The title of the site. Finally we conclude through experiments that our CSS algorithm can effectively found URL Obfuscating phishing attacks.

Keywords: Anti-phishing, Naive Bayesian Classifier, URL Obfuscation, Login interface.

ARTICLE INFO

Article History

Received: 6th May 2016

Received in revised form :

6th May 2016

Accepted: 10th May 2016

Published online :

12th May 2016

I. INTRODUCTION

In today's world phishing is the most popular and tremendously increasingly most lucrative crime activity. Some researches shows looses of several millions every year. Attacker designed such a pages that lure victim to enters his/her private information(such as password, credit card number, bank account details).Ones user enters this information goes to hacker and he misuse user's private information and this cause a greater loss to the user. Phishing is also headache for e-commerce due to user's distrust of the whole e-commerce environment.

Various techniques are used by the attacker to perform attacks such as DNS cache poisoning, Web Server Takeover, Email spoofing etc. Recent research has shown that phishers may in principle be able to determine which banks potential victims use, and target bogus emails accordingly.[1] Social networking sites are now a prime target of phishing, since the personal details in such sites can be used in identity theft; [2] in late 2006 a computer worm took over pages on MySpace and altered links to direct surfers to websites designed to steal login details. [3] Experiments show a success rate of over 70% for phishing attacks on social networks [4]. There are lot of solutions available to detect whether a web page is phishing page or not. But it is very difficult to detect all the attacks done by attacker but the ObURL algorithm which is used in our project as a primary or basic algorithm can detect the maximum number of URL Obfuscation phishing attacks because we follow the given test cases on every URL displayed on our Email.

- DNS Test.
-IP Address Test.
-URL Encode Test.
-Shorten URL Test.
-White List Test.
-Black List Test.
-Pattern Matching Test.

And it also checks the iFrame, source URL of iFrame, Content of iFrame's source URL, input form in Email.

II. RELATED WORK

ObURL algorithm is already developed but it is not provides security up to the mark so we are developed a new algorithm called CSS algorithm.

ObURL detection Algorithm:

Input: Content of Email.

Output: Prevent the User if URL seems Counterfeit.

Alert User:Possible phishing.
Safe User:No phishing.
DB:Database.

```
If input form found in Email Content then
        Alert User;
End
For each iFrame in Email content do
//get the content of iFrame
For each iFrame in Email content's iFrame source do
       if input form found then
        Alert User;
End
For each hyperlink in Emailcontent's iframe source do
        // perform the test 1 to 6
        End
For each hyperlink found in Email content and iFrame source
URL \

do

Test 1: //DNS Test

        if hypertext!= Anchortext
        then
                Alert User;
Test 2: //IP Address Test

        if IP address found in hyperlink
        then
                if IP address found in White list DB then
                                Safe User;
        Else Alert User;
                        //IP address found in blacklist DB
Test 3://Encoded Test

        if hyperlink found encoded
        then
                Decode hyperlink;
                Inform User;
Test 4://Shorten URL Test

        If URL is shorten
        then
                Alert User;

Test 5://hyperlink white list and black list test

        If URL found in white list and blacklist test
        then
                Safe user;
        Else
                Alert User;
                //URL found in Blacklist DB

Test 6://Pattern Matching Test.

        If hypertext and anchortext pattern is matching
        then
                Alert User;
```

III. PROPOSED WORK

After receiving related work on phishing and and some anti-phishing strategies we propose a novel solution, Bait Alarm, to efficiently detect phishing web pages. Note that page layout and contents are fundamental features of web pages appearance. Since the standard way to specify page layouts is through the style sheet(CSS), we develop an algorithm to detect similarities in key element related to CSS.

Phishing Attacks:

- First of all, the phisher have to create a phishing website to lure the victim which seems as legitimate one.
- Then host the site on internet for use of victim secrete information.
- If victim visit phishing website, it convinces the victim to enter some confidential information.
- Phisher then acquire some entered data and later it can be misuse by phisher.

Avoiding phishing Attacks:

- A whitelist in the context of phishing detection is simply a list of trusted websites.
- For CSS detection to work properly, the list contains more than just the URL of the trusted website. Each entry in the whitelist database contains six strings: The URL of the trusted site, The domain of the site, The title of the site, The CSS filename, The CSS domain and The CSS content of the file.

a) The URL of the trusted site :

The URL of the trusted site is used to periodically update the CSS information in the database. This is the URL of the site such as "".

b) The domain of the site:

The domain of the trusted site is the domain of the URL such as "signin.ebay.com" and is used to determine whether the current page dispalyed in the browser is on whitelist or not.

c) The title of the site:

The title of the trusted site is the page title of the site such as "Welcome to ebay" and can be used during CSS content detection to speed up detection by matching potential phishing site titles with titles in the whitelist database.

d) The CSS filename:

The CSS filename is the filename of the CSS file such as "paypal.css" and can also be used during CSS content detection to speed up detection by matching potential phishing site CSS filenames with filenames in the whitelist database.

e) The CSS domain:

The CSS domain is the domain of the location of the CSS file such as "secureinclude.ebaystatic.com". Often the domain is the same as the site domain, but in other cases such as eBay, the CSS file is hosted on a different domain. Storring the CSS domain is essential because if a match is found of a website not in whitelist. then it is most likely a

phishing site linking to the actual CSS file location of the legitimate site.

f) The CSS content of the file:

The CSS content is the actual text contained in the CSS file that contains all of the style information. The CSS content is used to compare with the CSS content of possible phishing site in order to determine if there is a match with a legitimate site.

CSS Detection Algorithm

Step 1: Rules set extraction:
When the user opens the web page, browser can capture the CSS structure of the page. The CSS structure is the series of rules with the general representation as

Selector 1 {Property1-1:
Value1-1;Property1-2:Value1-2;...};
Selector 2{Property2-1:
Value2-1;Property2-2:Value2-2;...};

Step 2: Convert CSS rules into comparison-unit:
In order to calculate similarity value more effectively, we convert CSS rule into a new representaion, which we called comparison-unit.

Definition 1: (Comparison-Unit) Given a Web page's CSS rule set,
CSS() = {.....,[selector{.....;[Property:Value;.....],.....}],.....},

the corresponding comparison units set of the web page is represented as

CompUnit()={.....,[Property:[.....;Value:[.....,Selector,.....],.....],.....,}.

Definition 2: (Complexity Score) The complexity Score of the webpage is the fundamental visual layout matrics.

Definition 3: (Match Score) Given the comparison unit of web page A and B.

Definition 4: (Similarity) Given the comparison unit of web page A and B.

Based on Analysis of our phishing pages ,the id and class selectors influence more in visual layouts similarity.Generally,different web pages should have different ID and class Selectors,specially for some unusual name of the ID selector.

IV. ADVANTAGES OVER PREVIOUS METHOD

Generally we use Google Browsing to search particular site but as we know that attacker can uses number of different methods to lure the victim but our CSS algorithm can provide maximum security to the URL attached to the Email. We have update the list of Whitelist database frequently.CSS algorithm provides trustworthy security because we use Comparison Unit, Complexity Score, Match Score, and Similarity test. It is secure from phishing sites.

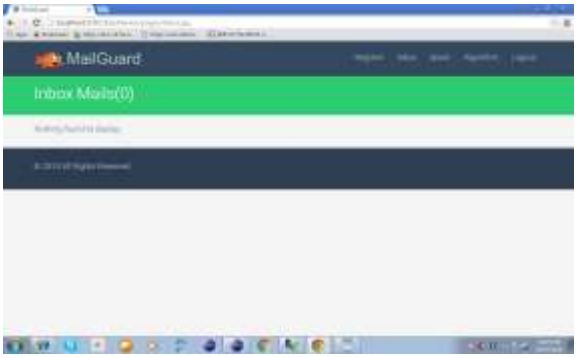
V. IMPLEMENTATION RESULT

1. Registration for new user

2. Login Window

3. Checking User name and Password

4. After Login



5. Algorithm

6. Logout

7. About Project

VI. CONCLUSION

Our CSS Detection algorithm can provide maximum security as well as it will work on both known as unknown phishing attacks. As we use ObURL algorithm also so our both algorithms provides multilayer security to each and every attachment of Email contains URL and provides accurate result to user. Our CSS algorithm increases the level of security to provide the user's confidential data more secure.

REFERENCES

[1] "Phishing for Clues". Indiana University Bloomington. September 15, 2005.

[2] Kirk, Jeremy (June 2, 2006). "Phishing Scam Takes Aim at MySpace.com". IDG Network

[3] "Malicious Website / Malicious Code: MySpace XSS QuickTime Worm". Websense Security Labs. Archived from the original on December 5, 2006. Retrieved December 5, 2006.

[4] Jagatic, Tom; Markus Jakobsson (October 2007). "Social Phishing". Communications of the ACM 50 (10): 94–100.

[5] Anti-phishing Working Group (APWG) Official site, http://www.apwg.org

[6] Phishing: The history of phishing attacks, URL:http://www.phishing.org/history-ofphishing/.

[7] Gaurav, Madhuresh Mishra, Anurag Jain, (March-April 2012) “Anti-phishing techniques: A Review” International Journal of Engineering Research and Application (IJERA), ISSN: 2248-9622, Volume.2 Issue.2, Pages: 350-355.

[8] The Phishing Guide, Understanding & Preventing phishing attacks. By: Gunter Ollmann, Director of Security Strategy IBM Internet Security System.

[9] Jigar Rathod, Prof. Debalina Nandy “URL Obfuscation Phishing and Anti-Phishing: A Review” International Journal of Engineering Research and Application (IJERA), ISSN: 2248-9622, Volume.4, Issue.1 (Version 1), January.2014.

