Advances in Natural and Applied Sciences, 9(6) Special 2015, Pages: 304-310

AENSI Journals

Advances in Natural and Applied Sciences

ISSN:1995-0772 EISSN: 1998-1090
Journal home page: www.aensiweb.com/ANAS

Detection of URL Based Attacks Using Reduced Feature Set and Modified C4.5 Algorithm

Rakesh, R., Muthurajkumar, S., SaiRamesh, L., Vijayalakshmi, M., Kannan A.

Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai 600025, India.

ARTICLE INFO

Article history:
Received 12 October 2014
Received in revised form 26 December 2014
Accepted 1 January 2015
Available online 25 February 2015

Keywords:
C4.5 algorithm, Classification
URL based attacks, Phishing
CSRF

ABSTRACT

Web Security concerns with the proposal of new security measures to use against attacks performed over the Internet. Phishing is as an activity where confidential information from the user is obtained by luring the user towards an illegitimate URL. Illegitimate web URLs inculcate a variety of features that makes them look as a replica of the legitimate URL. Therefore, Phishers employ such features by means of page content, User Interface (UI), Uniform Resource Locator address (URL) within their illegitimate webpage in order to make them look similar. Cross Site Request Forgery (CSRF), is another URL attack that typically makes use of cookies, browser authentication or client side certificates to perform a harmful action through a victim's normal action. By redirecting the user's browser to the target, the attacker makes use of cookies or currently active session to complete the attack. Many researchers have proposed various solutions to solve these problems, nevertheless, attacks that take place through URL's are on the rise and no single solution exist that could facilitate users to counter such URL threats. In this paper, a new algorithm for detecting URL attacks is proposed. This proposed work uses only the important features that effectively classifies illegitimate URL's. This algorithm is designed using a modified C4.5 classifier and a User Interface implementation that makes use of the knowledge obtained is used for testing the presence of Cross-Site Request Forgery form exploits and when such an exploit is found, the user is alerted.

To Cite This Article: Rakesh, R., Muthurajkumar, S., SaiRamesh, L., Vijayalakshmi, M., Kannan A., Detection of URL Based Attacks Using Reduced Feature Set and Modified C4.5 Algorithm. **Adv. in Nat. Appl. Sci.,** *9(6): 304-310, 2015*

INTRODUCTION

Generally, Phishing activity is usually accomplished by sending e-mails. Such an e-mail typically contains a link that looks like originating from a genuine organization and thereby urging the users to submit and verify their information by following the link they had received. The reasons that are cited for such verifications include, updation of user database, security breach, maintenance of backend tools, lucky draw in competitions etc. The URL that hosts the exact replica of the original webpage would then ask the user for information such as User ID's, credit or debit card numbers, date of birth, passwords etc. These tricksters obtain personal information from users by tricking the users with the help of browser pop-ups, browser session, drive-by-downloads from unknown sources etc.

Such tricksters target organizations which have a large customer base, send e-mails containing the illegitimate page link to thousands of users so as to improve their hit rate and setting up the illegitimate webpage in such a way that it deceives the user into exposing their information.

Thus phishing attacks in many cases, lead to loss of reputation and loss of money. In the year 2012 alone, around *300,000* phishing attacks were recorded worldwide and new online payment services, crypto-currency sites are being targeted more frequently. Though phishers deploy various methods in luring users, a set of characteristic features acts as a base for such illegitimate webpages. These features are predominant for the success of a user falling prey (Mohammed *et al* 2014) and these features help to classify a given webpage as either legitimate or not. Classification methods have been popular in detecting such features but these methods need to be adjusted or calibrated in order to decrease its verified high false positive rates (Zhang *et al* 2006).

Cross Site Request Forgery (CSRF) is among the most exploited web security vulnerabilities as per Open Web Application Security Project (OWASP). In a Cross Site Request Forgery attack, a malign site directs a user

Corresponding Author: Rakesh, R., Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai 600025, India.

(prey) browser to send a request to a legitimate site, as if the request was part of the user's (prey) interaction with the legitimate site, gaining control over the user's network connectivity and the browser's state, such as cookies, to interrupt the reliability of the user's session with the legitimate site (Bath *et al* 2008). A CSRF attack involves 3 components. They are a user (prey), a legitimate site, and a malign site. The user (prey) maintains an active session with a legitimate site and parallelly visits a malicious site too. If such a condition doesn't exist, then it is not termed as a CSRF attack.

The malign website injects a HTML or HTTP request for the legitimate site into the user's (prey) session thereby compromising the reliability of the legitimate site. Suppose there is a user who logs in a website A. After logging in to the website, he would work in a valid session. Now without logging out from the website, he also wants to visit a website B. But in the website B, attacker might have posted a malicious link in the website B which is able to send a HTTP request to the website A to request to perform some valid action but require a valid session. When User visits the website B and by mistake click on the malicious link, the HTTP request is sent to the website A which uses the valid session of the user to perform some valid action on the website A.

In this paper, features that are highly effective in detecting phishing URL's are analyzed using a modified version of C4.5 classification algorithm. The structure of the paper is organised as follows: Section 2 discusses about the related works that explores URL based phishing and form based CSRF attacks. Section 3 details about highly effective features that help categorize a phishing page along with examples of how a Cross-Site Request Forgery attack work. In Section 4, dataset preparation followed by system architecture are discussed. Finally, in Section 5, we explain how our proposed technique works better when comparing with other existing methods based on the classification accuracy.

Related Work:

In this section, we look through different methodologies adopted for identifying URL based Phishing and CSRF attacks.

There has been a number of literature work reporting the identification of URL based phishing attacks. Mc. Grath *et al*. (2008) effectively studied and carried out a relative analysis of malign (phish) and legitimate (nonphishing) URL's. The author extensively analysed characteristics like IP addresses, WHOIS records, GIS information lookups, relative features of the URL address such as length of the URL, URL characters used, presence of highly known and targeted organisation names etc and illegitimate use of free webhosting services by perpetrator. Zhang *et al*. (2007) developed a tool called CANTINA, which was used for classifying malign (phish) URL's by studying the content of any given webpage. Zhang *et al*. (2011) assigned a weighted sum to 8 URL features with respect to content, lexical, and WHOIS and built the classifier. Among the relative lexical characteristics, the author looked for URL dots, usage of or presence of special or certain characters, usage of IP addresses in the URL, and domain registration information such as age of the domain. The author further developed eight distinguishable features and proposed a new tool called CANTINA+ which analysed HTML Document Object Model (DOM) to find malign pages. Blum *et al*. (2010) proposed a method based on Support Vector Machine to detect malign or phishing URL's. The author identified and used twenty three characteristics in order to train the Support Vector Machine model which were based on domain feature, path feature and protocol feature of the URL. The author through his SVM model achieved an accuracy rate of 99%. Fette *et al*. (2007) used the concept of machine learning in order to find or classify phishing URL's or messages. The author used the characteristics of URL's present in a webpage or an email message such as the number of hyperlinks found in the message, number of domain info identified, and number of dots present in the URL etc. and identified malign or suspicious URLs with an accuracy rate of 96%. Whittaker *et al*. (2010) studied the URL characteristics and webpage content for any given URLs to determine whether a page is phishing or not. The author identified and used characteristics such as inclusion of IP address, lexical string features of a URL etc and successfully classified phishing pages with more than 90% accuracy. Xiaoli *et al* (2009) presented the result of the review of more than 200 CSRF attacks. The author also demonstrated a real world reflected CSRF attack and designed two types of CSRF threat models. Additionally, attack trees were developed which can be used by future researchers to design the CSRF defenses and mitigation mechanisms. Shahriar *et al* (2010) proposed the detection of CSRF attacks with the conceptual idea of checking the content of suspected requests. The basic idea is to capture a malign or suspicious request containing webpage parameters such as term frequency, form values such as user input fields and correlate these with one of the noticeable form fields present in an open window. If an exact match is found, the author modified the malign or suspicious request to make it look as if it is legitimate, and checked it by directing it to the remote webURL, identifying the type of content it has and matching with the expected content type. Any mismatch if found between the URL request feature values or page content type results in an alert. Their proposed technique does not require storing URLs or tokens that are to be checked and matched later for detecting such attacks.

Threat Model:

In this section, a basic overview of how the structure of a URL looks as well as the different URL features that are modified and maliciously used by phishers to lure users are discussed.

3.1 URL Structure:

Uniform Resource Locator is used to traverse or identify a webpage. An example for URL is, http://www.ebay.account.com/login/index.php.

The structure of an URL is as follows,

<protocol>://<subdomain>.<primarydomain>.<TLD>/<pathdomain>

For the above example, Protocol is http, Sub-domain is ebay, Primary-domain is account, Top-Level-Domain is com, Path-domain is login/index.php. Phishers usually make changes in URL or address of a webpage and create phish pages to lure users. Mohammad *et al*. [3] did a vast study on the different features that helps in classifying a phish webpage and concluded that large number of features identified only leads to increase in the classification process and hence identified 17 features that can effectively classify a fake URL.

3.2 CSRF Threat Modelling:

CSRF attacks are classified into either stored CSRF or reflected CSRF attacks. Let us see stored CSRF attack with an example. Let us assume that a user is logged on to a site (*www.abc.com*) that stores the user profile. The user profile includes a contact email address which has an initial value *user@abc.com*. The client side user interface provides a form (*emailchange.html*) to change the email address of a logged on user legitimately. A new email address provided by a user (Line 5) is updated by the server side script (*i.e.*, *profilechange.php* at Line 3). The request of the email address change is sent to *profilechange.php* by a hidden field (*newmailid*) at Line 4. The following code snippet explains the same,

Client side Code:

```
1. <HTML>
2. <BODY>
3. <FORM action = “profilechange.php” method = “POST”>
4. <INPUT type = “hidden” name = “action” value = “newmailid”>
5. <INPUT type = “text” name = “email” value = “”>
6. <INPUT type = “submit” value = “Change Email Address”>
7. </FORM>
8. <BODY>
9. </HTML>
```

Server side Code:

```
1. if (isset($_COOKIE[‘user’]))
2. {
3. if (! session_is_valid ($_COOKIE [‘user’] ))
4. {
5. echo “session is invalid.!”;
6. exit;
7. }
8. }
9. if ($_POST[‘action’] == ‘newmailid’)
10. {
11. change_profile($_POST[‘email’]);
12. }
```

The server side code snippet checks if a cookie value has already been set for the user (Line 1) and the session is valid or not (Line 2). If the session is not valid, then the program shows an error message and terminates (Line 3-4). Otherwise, the session is valid (Line 7), and the request is performed by calling the *change_profile()* at Line 8 with the new email address (*$POST[‘email’]*). If a user supplies the new email address as *newuser@abc.com*, the legitimate HTTP request becomes *http://www.abc.com/ editprofile?action=newemailid&email=newuser@abc.com*. Let us assume that the user is logged on to *www.abc.com* as well as visiting another site that contains a hyperlink *http://www.abc.com/editprofile?action=newemailid&email=attacker@abc.com*. If the user clicks on the link, the contact email address is changed to *attacker@abc.com*. The user becomes a victim of a reflected CSRF attack. To become a victim of a stored CSRF attack, the malicious link needs to be present in the webpage downloaded from the trusted website.

Proposed System:

In this section, a basic overview of how the structure of our proposed system works along with the function of different components are discussed. Our approach considers CSRF attacks through HTML form submissions alone.

A dataset consisting of 200 URL's were collected that included 100 illegitimate URL's and 100 legitimate URL's. In-order to provide an unbiased experimentation, the dataset consisted equal proportion of URL's.

Fake URL's were collected from PhishTank and legitimate URL's were collected from DMOZ repository. A modified version of C4.5 algorithm is used as the classifier. C4.5 classifier variant is chosen, as the authors of [3] have justified that C4.5 implementation classifies better when compared to other classification algorithms such as RandomTree, PRISM, Ripper, CBA etc. The modified algorithm is given below.

4.2 Algorithm to generate a decision tree from the given training data:

Input: set of candidate attributes**,** attribute list
Output: A decision tree
1) Create a node N;
2) if attribute-list is empty, then
3) return N as the leaf node;
4) if the sample set are all of the same class, C, then
5) return N as the leaf node labeled with the class C;
6) else, select gain-attribute, the attribute among attribute- list with the highest information gain ratio;
7) Label node N with gain-attribute;
8) for each known value of the gain-attribute
9) Grow a branch from node N for different test conditions;
10) Let si be the set of probable next gain attributes for the given test condition
11) repeat step 6 again
11) if si is empty, then select the next highest information gain attribute from step 6 and goto 7.
12) else if, no outcome is possible, attach a leaf labeled with the most common class from samples;

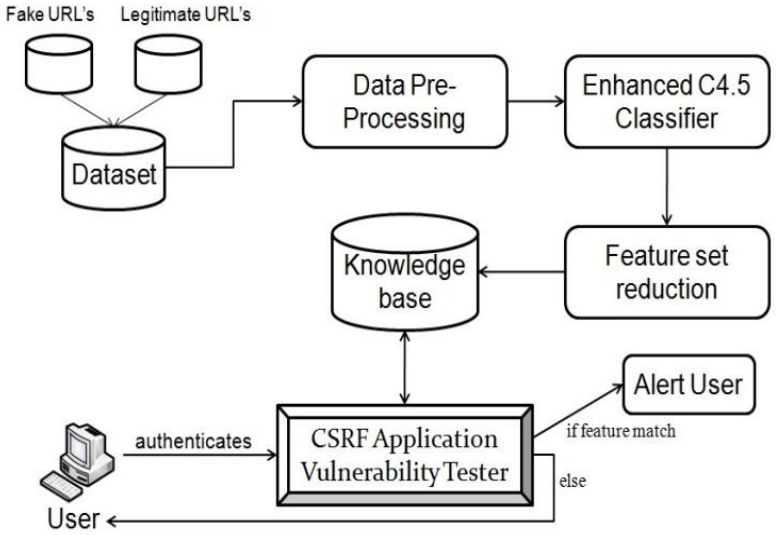

Fig. 1: System Architecture.

After the classifier ouputs a decision tree, rules are generated. These rules are then analyzed for their impact on the training dataset and a reduced feature set is derived. The reduced feature set that was derived from the classifier is as follows.

i. Usage of IP address
ii. request URL
iii. domain age
iv. usage of HTTPS and SSL
v. website traffic
vi. URL length
vii. subdomain and multi subdomain,
viii. adding prefix or suffix separated by hyphen to domain,
ix. URL of anchor

This reduced feature set is then inducted into the *Knowledge Base* which is used as a reference while testing for CSRF URL vulnerability. As given in the architecture diagram, the user who logins into the user interface and browses an example page, say, www.abc.com, the *CSRF Application Vulnerability Tester* component, analyses the URL of the landing page and identifies whether the URL of the landing webpage is a probable phish site and examines the form fields available with various instances of inputs so as to determine whether there is any change in the content type of the HTML page format thereby indicating a CSRF attack. When such an exploit or feature match is found, the user is alerted. Examination of form fields in CSRF Application Vulnerability Tester Component is carried out using the following steps.

4.2 Algorithm to test a site for malign URL:
Input: Login process
Output: Alert the user whether it is malign or not
1. Log into the site.
2. in the following site, if there is request or popup, eg. a hyperlink, check for URL feature match.
3. else-if the site has a form, test it with correct as well as with incorrect input values.
4. if both correct and incorrect values are accepted, alert the user and add the malign URL to the *Knowledge Base.*
5. else, repeat Step 2.
6. continue with site or log out.

Thus the proposed system works effectively in identifying malign URL's. Effective in the sense, URL's that possess malign features need not be assessed every time a request arise instead Knowledge base is used as a reference thereby reducing the computation needed to arrive at a solution.

Experimental Results:

In this section, we give a summary of how we conducted experiments and evaluated our proposed system against such threats. As mentioned earlier, the training dataset contains 200 URL's. A sample login page was developed where a user is allowed to log in, using his credentials. This training dataset contained URL's based on social networking, sports, entertainment, education and career sites which are shown in the user's login page. The homepage contains these list of sites and the user is given the option of clicking and viewing these sites. Malign requests from the URL's with eye-catching advertisements and phrases such as "*click me*", *"you have won a prize"* were taken into account while testing our proposed algorithm. Such eye catchy URL requests, linked the authenticated user with the malign sites that are active *[online]* and thus, such URL page requests were reported as malign by our proposed algorithm. Also, some of these malign requests were designed to redirect the user to another page that had form fields which requested the user to re-enter the login detail once again, which includes username and password. This form field accepts any such login information, even random login information and hence the URL was classified as malign and it was added to the *Knowledge Base*.

Table 1: Detection rate of the proposed algorithm.

S. No	Type of Site	Number of requests generated	Number of Malign requests detected
1	Social Networking	82	18
2	Entertainment	373	184
3	Sports	178	25
4	Career	220	111
5	Education	195	67

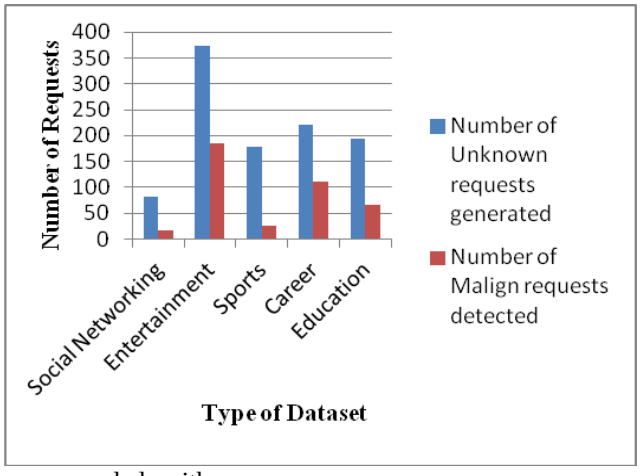

Fig. 2: Detection rate of the proposed algorithm.

Advances in Natural and Applied Sciences, 9(6) Special 2015, Pages: 304-310

Table 1 and Figure 2 shows the number of URL requests generated from the different category of sites used in our training dataset versus the number of malign URL requests detected by our proposed architecture.

Table 2: Success rate of the proposed algorithm.

S. No	Total requests	Malign requests	True Positives	False Positives
1	82	18	16	4
2	373	184	161	11
3	178	25	23	6
4	220	111	100	5
5	195	67	62	4

It is to be noted that, the total URL requests generated includes advertisements and popups from both legitimate as well as malign URL's. The variation in the number of malign requests detected with respect to different domains can be attributed mainly because of the number and type of URL request the page generated. For eg. A user browsing career oriented sites such as www.monsterjobs.com may be asked to enter his LinkedIn social networking details so as to successfully connect with employers. Similarly, sites such as www.rottentomatoes.com (entertainment) doesn't require any need for requesting social networking details and hence the variation in identification of malign requests. Also, the proposed algorithm is found to be consistent with respect to true positive rate across all domains while identifying malign requests. Table 2 shows the True Positives and False Positives detected by the proposed system.

Sensitivity for Type I sites (Social Networking)
= total true positives / total malign requests
= 16 / 18
=>0.88
Sensitivity for Type II sites (Entertainment)
= total true positives / total malign requests
= 161 / 184
=>0.875
Sensitivity for Type III sites (Sports)
= total true positives / total malign requests
= 23 / 25
=>0.92
Sensitivity for Type IV sites (Career)
= total true positives / total malign requests
= 100 / 111
=>0.90
Sensitivity for Type IV sites (Education)
= total true positives / total malign requests
= 62 / 67
=>0.925
Average sensitivity score is given by,
∑ Sensitivity for all sites ÷ Total number of different types,
= (0.88+0.875+0.92+0.90+0.925) / 5
= 0.90
Similarly, Specificity for each site is calculated as follows,
Specificity for Type I sites (Social Networking)
= total true negatives / total legitimate requests
= 60 / 64
=>0.937
Specificity for Type II sites (Entertainment)
= total true negatives / total legitimate requests
= 178 / 189
=>0.94
Specificity for Type III sites (Sports)
= total true negatives / total legitimate requests
= 147 / 153
=>0.96
Specificity for Type IV sites (Career)
= total true negatives / total legitimate requests
= 104 / 109
=>0.954

Specificity for Type IV sites (Education)

= total true negatives / total legitimate requests

= 124 / 128

=>0.968

Average specificity score is given by,

∑ Specificity for all sites ÷ Total number of different types,

= (0.937+0.94+0.96+0.954+0.968) / 5

= 0.95

Accuracy of our proposed system is calculated as,

= { ∑ (TP + TN / Population) / 5 }* 100

= 93.38 %

Conclusion:

In this work, a new URL classification technique is proposed to detect both phish and CSRF URLs. Also, this work analysed the different research works carried out by various researchers for providing effective counter measures against URL attacks such as malign URL's and CSRF attacks. From our analysis, it is observed that, the proposed system provides an accuracy rate of 93.38 %.

The performance analysis carried in this work shows that the proposed scheme is highly efficient. As part of the future work, the scope of this research can be extended to large datasets and the work can be developed into a standalone browser, providing reputation scoring using the *Knowledge Base* and can be extended for other types of CSRF attacks also.

REFERENCES

OWASP, Open Web Application Security Project [Online]. Available: https://www.owasp.org.

Mohammad, Thabtah, McCluskey, 2014. Intelligent Rule-based Phishing Websites Classification, IET Information Security, 8: 153-160.

Zhang, J., M. Zulkernine, 2006. A hybrid network intrusion detection technique using random forests, Proceedings of the First International Conference on Availability, Reliability and Security, Vienna, 121-132.

Bath, A., C. Jackson and J. Mitchell, 2008. Robut Defenses for Cross-Site Request Forgery, Proceedings of the 15th ACM Conference on Computer and Communications Security, 75-88.

Zhang, Y., J. Hong, L. Cranor, 2007. CANTINA: a content-based approach to detect phishing web sites, Proceedings of 16th World Wide Web Conference, 639-648.

McGrath, D.K., M. Gupta, 2008. Behind phishing: An examination of phisher modi operandi, Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, 1-8.

Xiang, G., J. Hong, P. Rose and L. Cranor, 2011. Cantina+: a feature-rich machine learning framework for detecting phishing web sites, ACM Transactions on Information and System Security, 14: 1-28.

Blum, A., B. Wardman, T. Solorio and G. Warner, 2010. Lexical feature based phishing url detection using online learning, Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security, 54-60.

Fette, I., N. Sadeh and A. Tomasic, 2007. Learning to detect phishing emails, Proceedings of the 16th international conference on World Wide Web, 649-656.

Whittaker, C., B. Ryner and M. Nazif, 2010. Large-scale automatic classification of phishing pages, Proceedings of the Network and Distributed System Security Symposium.

Xiaoli, L., P. Zavarsky, R. Ruhl, D. Lindskog, 2009. Threat Modeling for CSRF Attacks, International Conference on Computational Science and Engineering, 3: 486-491.

Shahriar, H., M. Zulkernine, 2010. Client-Side Detection of Cross-Site Request Forgery Attacks, International Symposium on Software Reliability Engineering, 358-367.