International Journal of Research in IT and Management (IJRIM)
Available online at : http://euroasiapub.org
Vol. 6, Issue 10, October - 2016, pp. 69~77
ISSN(o): 2231-4334 | ISSN(p): 2349-6517,| Impact Factor: 5.96 | Thomson Reuters ID: L-5236-2015

Browser Side Protection of E-Government Website Against Phishing

Oinam Bhopen Singh[1],
Research Scholar,
Department of CSE, University of Science and Technology, Meghalaya
Ri-Bhoi, Meghalaya-793101, India.

Dr. Hitesh Tahbildar[2]
Computer Science Department,
H.O.D, Assam Engineering Institute, Guwahati, India

ABSTRACT

The ability of E-Government to make public services easily available online to the citizen is an attractive prospect. But to create a trust context among the citizens to enable them to use the web for sensitive transactions is very important. As Phishing attacks are not limited only to the e-commerce and banking world, it is also affecting the e-Government world as most of the funds of various schemes or yojanas are directly transferred online to the beneficiaries' account. So in today online environment we need to protect the data from phishing and safeguard our information. In this paper we present a browser side protection mechanism which uses two approaches to prevent the users from entering into a phishing site. The first approach is a white-list which stores the legitimate e-government website and portals and just checks a legitimate site. If a user enters a phishing site or visits a domain name similar to a e-government site then it will use the next approach to identify the page. The second approach does a web page content analysis and computes a risk rating percentage and gives a alert message if the risk rating is greater than a threshold value.

Keywords

E-government services, Phishing, Blacklist, White-list, Heuristics, etc.

1. Introduction

E-Government refers to the use of *Information and Communication Technology*(ICT) by government agencies to transform relations with citizens, businesses and other arms of government[1]. The ability to make public services easily available online is an attractive prospect for governments, not just because this gives citizens easy access, but also because of the potential cost-savings. But e-Government services in their current status are not secure; they bring benefits but also security threats that need to be addressed.

During the delivery of online public services, it is not only important to authenticate the user for her/his identity, but it is also important to authenticate the website that the user is accessing for availing various public services. Considering the number of phishing attacks that take place over the web every day, the user must be able to correctly identify that the website that she/he has open is actually the right website that it is claiming to be. Phishing is a severe attack that deals with social engineering methodology to illegally acquire user's sensitive information.[2] Lack of appropriate protection measures may lead to revealing her/his personal credentials over a fake website, which can amount to severe financial and social losses to not only the user but also to the concerned department whose web interface was imitated for phishing.

If a particular URI emanates from a trusted government website and leads to an untrusted website (or fake website), then the browser should not allow the request to pass through. All the communication that happens between the browser and the server has to be shielded from man-in-the-middle attacks.

Hence PKI/SSL is a must for all government websites. There are various mechanisms available today which are intended to enhance user security and thwart phishing attacks. This is especially common in the banking world, where strong multi-factor authentications are used. Unfortunately, for most of the large e-government web applications, implementing strong two factor authentication like smart cards or hardware tokens is just not cost effective or practical because of the sheer size of the user population which needs to be dealt with that can often be upwards of several millions[3]. As a result these e-government services must resort to other creative ways to strengthen their authentication.

The mechanism that has to employed should address the *User*, the *Browser* or the *Web Server*. For example, the *user* may be educated not to trust (and click on) anything received over the Internet and to use different password for every Website. The *web server* are the e-government web server maintained by the government, so the protection mechanism can be address by them. The *browser* is the application software that is executed on the client's computer on behalf of the user to initiate and perform the transaction. The protection mechanism to address the browser has to be initiated by the user.

So, in this paper we are presenting a *browser-side protection model* to protect against phishing attacks on e-Government websites. This model uses the concept of white-list and heuristic method. It works in a two stage process, in the first stage the entered URL or the link URL is first compared with the white-list. If it is found in the white-list the websites is a legitimate one. In the second stage the URL not found in the list is evaluated on the basis of its content. To compare the user entered URL with the white-listed URL, we use Levenshtein edit distance algorithm[4], which is a metric for measuring the amount of difference between two strings. And for evaluating content based analysis, a risk rating percentage is calculated. The risk rating value determines whether the website is a potential threat to user or not. The website content is analyzed using five different characteristics: Referrer, Password, Encryption, Age, and Country.

The paper is organized as follows. In the first section the problem of phishing in e-government websites and our approach to solve the problem is explained briefly. Next the second section is a literature survey on existing anti-phishing browser extension. The third section presents our anti-phishing model. Finally, the paper ends with the conclusion.

2. Literature Survey

To counter the phishing threats, a number of anti-phishing browser extensions have been proposed, both by industry and academic world. A few of them have been surveyed which had help in developing our model.

2.1 AntiPhish

AntiPhish [5] is a browser extension that aims to protect inexperienced users against spoofed web site-based phishing attacks. It keeps track of a user's sensitive information like password and prevents this information from being passed to a website that is not considered safe. The development of AntiPhish was inspired by automated form-filler applications. It takes the common functionality integrated in most browsers such as Mozilla or the Internet Explorer that allows form contents to be stored and automatically inserted if the user desires. This content is protected by a master password.

2.2 Dynamic Security Skin

Dynamic Security Skin [6] is a browser extension for Mozilla Firefox, that allow a remote web server to prove its identity in a way that is easy for a human user to verify and hard for an attacker to spoof. It uses two interactive techniques to prevent spoofing. First, the browser extension provides the user with a trusted password window. Next, it provides a technique for a user to distinguish authenticated web pages from insecure or spoofed web pages.

2.3 eBay's AccountGuard Toolbar

The eBay Toolbar [7] uses a combination of heuristics and blacklists. The toolbar also gives users the ability to report phishing sites, which will then be verified before being blacklisted.

2.4 Google Safe Browsing

Google provides the source code for the Safe Browsing feature[8] and says that it checks URL against a blacklist.

2.5 iTrustPage

The iTrustPage [9] is a browser extension that does not rely on automation to detect phishing. Instead, iTrustPage relies on user input to decide on the legitimacy of a web form. It prevents users from entering any information into suspicious web form. iTrustPage intercepts the user input and prompts the user for search terms that describe the Web page the user intend to visit. With these search terms, iTrustPage performs a Google search and validates the web form only if it appears among the top search results.

2.6 McAfee SideAdvisor

SiteAdvisor [10] claims to detect not just phishing websites but any sites that send spam, offer downloads containing spyware, or engage in other similar bad practices. The determination is made by a combination of automated heuristics and manual verification.

2.7 Microsoft SmartScreen Filter

Microsoft SmartScreen Filter [11] is a tool for Internet Explorer 9 (IE9) users which uses blacklist and heuristic analysis to determine whether the page is phishing or legitimate. When a user visits a page, the contents of the page are compared against heuristic characteristics. If the page fails to pass the heuristic test, a yellow shield will appear warning the user about the contents of the page and will suggest the user not to enter any confidential information. However, if no suspicious properties are found, the tool will check its URL against a blacklist. If a match is found in the blacklist, a red shield will appear informing users about the blacklisted page.

2.8 Netcraft Anti-Phishing Toolbar

The Netcraft toolbar [12] also uses a blacklist, which consists of fraudulent sites identified by Netcraft as well as sites submitted by users and verified by the company. The toolbar also displays a risk rating between one and ten as well as the hosting location of the site.

2.9 PhishNet

PhishNet [13] is anti-phishing tool to improve the resilience and efficiency of blacklists significantly. It comprises of two major components. The first is the URL prediction component that works in an offline fashion, examines current blacklists and systematically generates new URLs by employing various heuristics. The second is an approximate URL matching component which performs an approximate match of a new URL with the existing blacklist.

2.10 PhishProof

PhishProof [14] is an anti-phishing tool designed to help Firefox users distinguish between phishing and legitimate websites. After the PhishProof toolbar is installed on Firefox, a toolbar appears on the main browser window. The toolbar has two states, idle and active. When a browser or a new tab is opened, no page is loaded in the browser window, the PhishProof toolbar is in idle state. The PhishProof toolbar become active when a page is loaded in the browser window. PhishProof's active state has four components: PhishProof menu button, Risk rating bar, since label and country label.

2.11 PwdHash

PwdHash [15] is a browser extension that transparently produces a different password for each site, improving web password security and defending against password phishing and other attacks. It applies a cryptographic hash function to a combination of the plaintext password entered by the user. In essence, the password hashing method is extremely simple; rather than send the user's cleartext password to a remote site, a hash value derived from the user's password, *pwd,* and the site domain

name. Specifically, PwdHash captures all user input to a password field and sends *hash (pwd, dom)* to the remote site, where *dom* is derived from the domain name of the remote site.

2.12 Spoofguard

Spoofguard [16] does not use white-lists or blacklists. Instead, the toolbar employs a series of heuristics to identify phishing page. Spoofguard monitors a user's Internet activity, compute a spoof index, and warns the user if the index exceeds a level selected by the user. It then translate this index into a traffic light: red for spoof index above a threshold, indicating the page is probably hostile; yellow for index in the middle; and green for low index, indicating the page is probably safe.

2.13 TrustBar

TrustBar [17] is a browser extension for improve secure identification indicators. Users can assign a name/logo to a secure site, presented by TrustBar when the browser presents that secure site; otherwise, TrustBar presents the certified site's owner name, and the name/logo of the Certificate Authority (CA) who identified the owner.

2.14 TrustWatch

ThrustWatch [18] is a domain verification toolbar and search site developed by GeoTrust. It is a browser extension that displays information to help consumers instantly verify the identity and check security standards of web sites that are conducting e-business or requesting confidential information. TrustWatch provides a web site verification rating by displaying a green, yellow or red light.

3. Browser Side Protection Model

Most client side solutions are browser plug-ins or extensions that are installed on user's machine. These extensions monitor websites visited by users and informs them if they are about to enter a fraudulent page. There are many different solutions present to help users distinguish a fraudulent page from a legitimate one but each technique has some limitations. As we have seen from the literature survey, most of the techniques for phishing detection are based on blacklist, heuristics or a combination of both. In the blacklist approaches, when the user visits a website that is in the blacklist, she/he will be warned. But maintaining a blacklist requires a great deal of resources for reporting and verification of the suspicious websites. In addition, phishing sites emerge endlessly, so it is difficult to keep a global blacklist up to date. In the heuristic method, a risk rating score is computed for each webpage based on the weight of results of each sets of heuristic characteristics. If the risk rating score is greater than a threshold value different warning message are given.

For our case at hand we are considering a particular case in which only the e-Government websites are verified and protect the users from Phishing attacks. In this browser side protection model we are going to use the white-list approach and the heuristic approach. In the white-list, we can easily maintain a list of legitimate e-Government websites and portals along with some few trusted websites like, Google.com, Facebook.com, etc., which is very limited in number. Using the URL entered by the user a comparison is done with the white-list and verifies only the legitimate e-government site. In the comparison if a match is not found then we use the heuristic method to identify a phishing site or a legitimate site.

In the heuristic approach, webpage content analysis is done to counter phishing attack. Five heuristic characteristics: referrer, password, encryption, age and country are used. Their individual characteristic score is computed and finally the risk rating is used to classify the webpage is a phishing or not.

3.1 White-list Approach

A white-list[19] or approved list is a list of entities that for one reason or the other are provided a particular privilege, service, mobility, access or recognition. A white-list is the best way to validate input. One will know exactly what one's desire is and that there is not any wrong types accepted. When compared to blacklist method, white-list data is short and precise. White-list can contain data as per the comforts of the user, if the user is specific about the kind of sites he wishes to visit. On the other hand, blacklist needs dynamics update of list in order to warn the user about the site visited. We implemented

white-listing concept. We use Levenshtein edit distance algorithm [20] to compare the user selected URL and with the white-listed URLs.

The Levenshtein distance is a string metric for measuring the amount of difference between two sequences. The Levenshtein distance between two strings is defined as the minimum number of single-character edit (i.e. insertion, deletion, or substitution) required to change one string into the other. The Levenshtein distance may also be referred to as edit distances. The most common way of calculating [21] this is by the dynamic programming approach. A matrix is initialized measuring in the (m,n)-cell the Levenshtein distance between the m-character prefix of one with the n-prefix of the other string. The matrix can be filled from the upper left to the lower right corner. Each jump horizontally or vertically corresponds to a insert or a delete respectively. The cost is normally set to 1 for each of the operations. The diagonal jump can cost either 1, if the two characters in the row and column do not match or 0, if they do. Each cell always minimizes the cost locally. This way the number in the lower right corner is the Levenshtein distance between both strings.

For example, here the Levenshtein edit distance between two strings – "india.gov" and "Indian.com". The first string is the host name of genuine Indian government web portal and the second is not a government site.

		I	N	D	i	a	.	g	o	v
	0	1	2	3	4	5	6	7	8	9
i	1	0	1	2	3	4	5	6	7	8
n	2	1	0	1	2	3	4	5	6	7
d	3	2	1	0	1	2	3	4	5	6
i	4	3	2	1	0	1	2	3	4	5
a	5	4	3	2	1	0	1	2	3	4
n	6	5	4	3	2	1	1	2	3	4
.	7	6	5	4	3	2	1	2	3	4
c	8	7	6	5	4	3	2	1	2	3
o	9	8	7	6	5	4	3	3	1	2
m	10	9	8	7	6	5	4	3	2	1

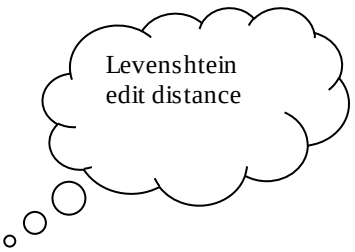

Fig. 1 Example of Levenshtein distance matrix

The algorithm [22] to check the URL with the white list is given below:

3.1.1 Algorithm for White-list Check

Step 1

The URL of the site entered by the user is taken as input string. This string is edited such that the host name of the site is retained and the path for the resource is eliminated.

Step 2

The edited URL is compared with all the URL's present in the white-list. The comparison is carried out by using Levenshtein edit distance algorithm.

For example refer to fig. 1 above.

Step 3

The minimum of all the edit distances calculated is taken and compared with the threshold value. The threshold value is given such that the e-Government sites in the white list are distinguished from the non e-governmental sites i.e., if the minimum edit distance value is greater than the threshold value then it is not a e-government site. If that value is less than threshold then the URL will go for IP check, the next step.

Step 4

As the phishing sites uses the host name which is very near to the legitimate site, the edit distance value will obviously be low. So the IP address of the entered site is compared with the IP address of the site in the white-list which encountered the minimum edit distance. If both the addresses are same, then it is a legitimate URL. If not, go for heuristic approach.

A phishing site can be caught at step 2 and 3, but we are using it only to find a legitimate e-government website. To detect a phishing site we will analyze the heuristic characteristic in the next approach.

3.2. Heuristics Approach

In this approach[16] to protect against a phishing attack, it involves web page content analysis and calculating a risk rating percentage which determines whether the web page is a threat or not. If risk rating calculated is above a threshold value, the current page is classified as a potential threat and users are notified through alert message. Five characteristics *referrer, password, encryption, age and country* contribute to the risk rating percentage. Each of these characteristics are checked individually and their individual scores are assigned. All characteristics(C_i) are assigned score, either 0 or 1.C_i=0 means it is a legitimate page whereas C_i=1 means the current page maybe a phishing page. To analyse these characteristics we need to include two more list i.e. the Mail list and the Country list, which is implemented. The Mail list contains the domain names of email providers like Gmail, Hotmail, Yahoo, rediffmail, etc. The Country list contains the name of the countries which have a history of hosting phishing websites.

3.2.1 Heuristic Characteristics

The five heuristic characteristics [14] used in calculating the risk rating are discussed below:

3.2.1.1 Referrer Characteristics

Referrer characteristic checks how the user has reached current page. Most phishing attacks are initiated by sending fake emails to a large number of internet users.[22] Therefore pages referred from emails are considered a higher potential threat as compare to pages referred from other websites.

3.2.1.2 Password Characteristics

Since phishing is about getting personal details of users, therefore almost all phishing pages have a password field. Password characteristic check scans the current webpage to find any input field of type password. If no input field of type password is found, password characteristics score is set to low. If any single input field with type password is found, the domain is also checked for encryption protocol SSL (Secure Sockets Layer). If SSL is not used by a page requesting password, there is a high probability of website being a phishing websites. Hence password characteristic score is increased.

3.2.1.3 Encryption Characteristics

Majority of phishing pages don't use SSL authentication protocol to transfer user information. Attackers avoid installing this service because some cost is involved. Phishing websites are active only for a short time and attackers avoid paying anything for such short time. Therefore SSL is considered as an important property in determining risk rating of a page. If a webpage request password and does not use SSL to encrypt user data, encryption characteristics score is set to high risk.

3.2.1.4 Age Characteristics

Phishing websites have a very short life time. They have an average age of 3.1 days and many disappear within hours. So the age of each website encountered is calculated, if the age of the website is less than a threshold value, age characteristics score is set to high risk.

3.2.1.5 Country Characteristics

There are countries that have a higher probability of hosting phishing websites as compared to other countries[23]. Countries are classified as potential threat if they have a high ratio of hosting phishing websites to legitimate ones. If the hosting country has a history of hosting phishing websites the country characteristics score is set to high risk.

3.2.2 Calculating Individual Characteristics Score

The webpage characteristics required to compute are extracted using DOM[24] and analyzed to find any characteristics identical to those of a phishing website. And accordingly a characteristics score value (C_i) is assigned. This is achieved through the algorithm given below:

3.2.2.1 Algorithm for Calculating Characteristics Score

Step 1

Load the DOM of URL which is not in the white-list.

Step 2

Get the domain name. Compare the domain name with the Mail list. Check if it exist or not. Set referrer characteristics score. Represent it by C_1.

Step 3

Get all elements of a website that are input fields of a form. Check if password field exist. If yes go to step 4, otherwise set password characteristic score. Represent it by C_2.

Step 4

Get protocols that are available. Check if HTTPS exists or not. Set encryption characteristics score. Represent it by C_3. Also set password characteristics score if step 3 result a 'yes', represent it by C_2.

Step 5

Get age of domain. Check if age is less than threshold value or not. Set age characteristics score. Represent it by C_4.

Step 6

Get country hosting the website. Compare the hosting country with the country list. Check if hosting country is in the list or not. Set country characteristics score. Represent it by C_5.

Finally, all individual characteristics score (C_i) are combined with preset weight (W_i) to compute risk rating. Weights are assigned to give each characteristics different weightage as some characteristics are more important in categorizing a page as compared to others. For example, a website requesting password and not using authentic encryption will be considered more of a threat as compare to a website hosted in any country in the country list.

3.2.3 Calculating Risk Rating Percentage

Risk rating is the ratio of the weighted scores to total score as shown in the Equation 3.1 below:

$$\text{Risk Rating} = \frac{\sum_{i=0}^{n}[(Wi)*(Ci)]}{\sum_{i=0}^{n}(Wi)}$$

Equation 3.1: Total Risk Rating Calculation

Where, W_i and C_i represents the preset weight and individual score of i^{th} characteristics. The values for C_i are set by the algorithm, where C_i is 0 for legitimate pages and C_i is 1 for phishing pages. The value of risk rating obtained is between 0 and 1. A web page is considered as a phishing threat if the risk rating value, as defined in equation 3.1 exceeds a certain threshold δ, i.e.

$$\frac{\sum_{i=0}^{n}[(Wi)*(Ci)]}{\sum_{i=0}^{n}(Wi)} > \delta$$

This equation is adopted from a journal on detecting phishing pages by Angelo P.E. Rosiello et al.,[24]. It is used to compute risk rating because when certain features are present, the probability of webpage being phishing increases. The risk rating computed is multiplied by 100 to get risk rating percentage.

4. Conclusion

We look at various anti-phishing browser extensions to mitigate phishing in the literature survey and we presented a new one. We have seen that with the use of white-list, false positive alert especially for e-Government websites are reduced. Even though this protection is meant for e-Government websites, it can be used to detect any phishing attacks through the heuristic approach applied in this model. So only when the browser technology advances to incorporate trust notions, is when people are going to feel safe about using the browser for sensitive transactions.

References

[1] S. Basu, E-government and developing countries: An overview, International review of Law, Computer and Technology, vol. 18 of pg. 109-132, 2004.

[2] Anti-Phishing Working Group. http://www.antiphishing.org/.

[3] Department of Information Technology, Government of India. Draft National e-Authentication Framework, version:1.0, 2011.

[4] https://en.wikipedia.org/wiki/Levenshtein_distance, Accessed on 7th July 2015.

[5] E. Kirda and C. Kruegel. Protecting Users Against Phishing Attacks with AntiPhish. In COMPSAC 2005: proceedings of the 29th Annual International Computer Software and Applications Conference (COMSAC'05) Vol. 1. Pages 517-524, 2005

[6] Rachna Dhamija and J.D. Tygar, The battle against phishing: Dynamic Security Skins. Proceedings of 2005 ACM Symposium on Usable Security and Privacy, pp 77-88, ACM Press, July 2005.

[7] eBay Toolbar and Account Guard, Accessed 22nd June 2015. http://pages.ebay.in/help/account/toolbar-account-guard.html.

[8] Google Safe Browsing for Firefox. www.google.com/tools/firefox/safebrowsing.

[9] Ronda T, Saroiu S and Wolman A, "iTrustPage: A User-Assisted Anti-Phishing Tool", Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems, pp.261-272, 2008.

[10] B. Ross, C. Jackson, N.Miyake, D. Boneh, and J. C. Mitchell. Stronger Password Authentication Using Browser Extensions. In 14th Usenix Security Symposium, 2005.

[11] Microsoft SmartScreen Filter. Accessed 20th July 2014. http://windows.microsoft.com/en-in/internet-explorer/products/ie-9/features/smartscreen-filter .

[12] Netcraft extension accessed 18th July, 2014. http://toolbar.netcraft.com/.

[13] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, Minaxi Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks", Purdue University, Indiana University.

[14] Taimoor Zahid, "An Anti-Phishing tool: Phishproof", A dissertation submitted to the University of Manchester for the degree of Master of Science in the faculty of Engineering and Physical science, 2012.

[15] Ross B, Jackson C, Miyaki N *el al.* "Stronger Password Authentication Using Browser Extensions". Proceedings of USENIX Security Symposium, pp. 17-32, 2005.

[16] Chou N, Ledesma R, Teraguchi Y and Mitchell J C, Client –side Defense against web-based identity theft. Proceedings of 11th annual Network and Distributed System Security Symposium, California, USA, 2004.

[17] Herzberg A and Jbara A, "Security and Identification Indicators for Browsers Against Spoofing and Phishing Attacks", ACM *Transactions on Internet Technology*, Vol. 8, No. 4, Article 16, pp.1-36, 2008.

[18] TrustWatch accessed 18th July,2014. https://www.trustico.co.in/material/DS_TrustWatch.pdf

[19] Ye Cao, Weili Han and Yueran Le. Anti-phishing Based on Automated Individual White-List. InDIM '08, 2008, Fairfax, Virginia, USA.

[20] Levenstein, A., Binary codes capable of correction, deletions, insertions, and reversals. Soviet Physics doklady 10(1966), pg 707-710.

[21] Efficient Implementation of the Levenshtein-Algorithm, Fault-tolerant Search Technology, Error-tolerant Search Technologies. Access 10th July 2015. http://www.levenshtein.net/.

[22] Phishing. http:// en.wikipedia.org/wiki/Phishing

[23] Netcraft, Phishiest Countries. Accessed 7th July 2015 http://toobal.netwraft.com/stats/countries.

[24] Angelo P.E. Rosiello, Engin Kirda, Christoper Kruegel and Fabrizio Ferrandi, "A Layout-Similarity-Based Approach for Detecting Phishing Pages", Secure Systems Lab, Technical University Vienna.