Enhanced Link Redirection Interface for Secured Browsing using Web Browser Extensions

Mrinal Purohit Y
Dept. Of Computer Science
Amrita University
Coimbatore, India

Kaushik Velusamy
Dept. Of Computer Science
Amrita University
Coimbatore, India

Shriram K Vasudevan
Dept. Of Computer Science
Amrita University
Coimbatore, India

Abstract—**In the present world scenario where data is meant to be protected from intruders and crackers, everyone has the fear to keep their private data safe. As the data is stored on servers accessed through websites by browsers, it's the browsers, which act as a medium between a user and the server to send or receive data. As browsers send data in plain text, any data which is sent could easily be intercepted and used against someone. Hence this led to the use of Transport Layer Security (TLS) and Secure Socket Layer (SSL), which are cryptographic protocols designed to provide communication security over the Internet. A layer on top of SSL/TLS, support an encrypted mode, also known as HTTPS (HTTP Secure). Therefore, one of the main aspect of security lies in the website supporting HTTPS. Most websites have support for this encrypted mode and still we use an unencrypted mode of websites because a common user is unaware of the advancements in the field of technology. So to help us, in browsers, we have extensions or plug-ins to ease our life. This paper proposes the idea to implement the security measures in the web browsers.**

Keywords—Browsers; HTTPS; SSL

I. Introduction

Now-a-days, the browsers are the common application which people use to obtain information and share the same with others. It has become a common sight to the eye. But it's not just the information which matters; it is the person's personal information which is at stake. Browsers are said to be secured when the data is sent from the browser, only based on what type of layer of security the website has used. If there is no layer of security, then the data is sent as a plain text. Few websites use HTTPS which makes the website secure by encrypting data using long term public and secret keys, before sending. Hence it depends on whether a website has this support for HTTPS or not. Currently most websites offer HTTPS versions of their simple websites which are only triggered when there is a data exchange between the user and server.

A browser extension is a computer program which extends a normal browser's functionality. Browsers lack in few functionalities which are complemented by extensions. Extensions can be disabled but as they help a user in providing assistance, the user is forced to be dependent on them to make their work easier and make their browsing experience better. Different browsers have different requirements for an extension to be developed. Each browser have their own set of architecture and API's which requires different code and skills for each extension. Extensions are developed using web technologies like HTML, JavaScript, CSS and XML. Most famous browsers like Chrome and Firefox have their own web store for extensions which can be downloaded and used by anyone. Thus making it a very powerful tool for developers to make use of the browser's robustness.

This paper combines secure web browsing using HTTPS focusing on website redirection using browser extension and spam filtering to save the user's personal data and to make his browsing experience secure. The links which are classified as spam are stored in a vault for future references for spam and secure redirection.

II. Literature Survey

Browsers being one of the main source of information retrieval from the Internet, have many vulnerabilities. Though they have private browsing feature in browsers like Chrome, Firefox and Internet Explorer they are still prone to attacks as stated in [9]. They describe the flaws existing in browsers even in the private mode, which the user imagines is secure. They describe attacks prone to a local attacker and a web attacker who can access personal data even if the user uses private browsing mode. The proposed flaw points the use of extensions leaving trace of websites visited, on the disk which can be accessed by an attacker. Hence even if a user makes use of private browsing, they are still not secure. But this gives an insight that web URL's can be accessed in both normal and private mode of the browsers [9].

Each browser differs in its extensible architecture and working. Extensions depend on the architecture, whether they can be developed or not. For example, from the following list of browsers, Internet Explorer, Firefox, Chrome and Safari, only Safari doesn't support extensions. [9]. An extension called BROWSERSPY was developed, which did not require any special privileges but still it managed to take complete control over the browser and observe all activity performed through the browser staying undetectable. Extensions can be harmful but at the same time helpful. An example is PwdHash [10] which hashes the plaintext data given by the user, data associated with the website and a private salt stored on the client machine. Therefore, there can be both security oriented and security hindering extensions existing in the extensions market [7].

Both the end users and administrators of various services on the Internet such as email systems use different anti-spam techniques. Some of these anti-spam techniques have been embedded in products, services and software to ease the burden on users and administrators, But there is a unique technique which serves as a complete solution to the spam problem for securing the browsing activities on the Internet and each has different trade-offs ranging between incorrectly rejecting legitimate links Vs. not rejecting any spam link.

Table.1 below describes security in Google Chrome, IE and Mozilla Firefox. The comparison considers metrics such as vulnerability report counts and URL blacklists [3]. This paper takes a fundamentally different approach, examining which security metrics are most effective in protecting end users who browse the Internet. The following graph shows the comparison of different browsers based on the analysis of the mentioned factors. It is seen that all the three major browsers don't provide URL Blacklisting service. We have analyzed that neither Google's Safe Browsing service nor Microsoft's URS, appears to provide a fully comprehensive snapshot of all malware and spam web links in the wild at any given point in time [8]. This proves a strong support for the idea proposed in this paper.

The main concern of a browser extension is to secure the user from malicious links and safe guard the user's private data that are shared on the Internet. Rather than discovering vulnerabilities, it is the need of the hour to protect a user in their browsing experience. Thus, this paper proposes the basic idea to use the browser extensions to prevent spam and make the fullest utilization of the browsers as well as the website's fullest power which supports the HTTPS.

TABLE I. Comparison of factors leading to vulnerabilities in different web browsers.

Criteria	**Chrome**	**Internet Explorer**	**Firefox**
Sandboxing	Yes	Implemented	No
Plug-in Security	Yes	Implemented	No
JIT Hardening	Yes	Yes	No
ASLR	Yes	Yes	Yes
DEP	Yes	Yes	Yes
GS	Yes	Yes	Yes
URL Blacklisting	No	No	No

III. SPAM WEB LINKS

Most spam travels through blog networks. In order to get link redirections back to their sites or their client's sites, members of fake blog owners are paid for posting the spam links for higher hits. Guest blogging and other forms of contributing content to legitimate sites is a much whiter tactic, but considering that as a strategy that relies heavily on low-quality advertisement. Guest blogging looks similar to a blog network spam.

Article marketing is another method to spread spam. This method provides one or two links with the anchor text of the user's choice, and hence the ranking increases in search engines. Such articles are found to be easy, cheap and without creativity or mental effort. Most articles on the Internet are made for the sole purpose of getting huge hits for their links, and essentially all the followed links are self-generated rather than endorsements. Due to its wide spread on the Internet these links with no weights come in and the links with no impact go out. They are persistent because of decent free template which is not filtered by Google.

Most links which the users don't want to visit are embedded in a site wide link where the users are redirected to visit, so as to bring attention to their websites. Creating a piece of link and later replacing the content with something more beneficial and tricking the people to link to their desired content are examples of Link Bait Switching. Social Bookmarking and sharing sites carry many web links which don't have any value. Profile spam and comment spam add to the above [1].

Spam web links are spam links which are spread on the Internet, and which take advantage of link based ranking algorithms which gives websites higher rankings the more highly ranked websites linked to it. It's a trend on social networking sites to spread spam links [5]. Social sites with low spam control, stops getting visitors when being overrun by low quality external links. Handling spam is getting harder day by day as new technology emerges. Spam traps are often email addresses that were never valid or have been invalid for a long time, which were used to collect spams. They are found by pulling addresses off the hidden webpages. Spamcop, a blacklist directory uses spamtraps to catch spammers and blacklist them. Hence it gets tougher to track them out through web services [11].

Google bomb refers to the practice of creating a large number of links that cause the webpages to have high rankings in Google searches. It is mostly done for either business, political or pun purposes. Spam web links are the links which are spread on the Internet, and take advantage of link based ranking algorithms, which gives websites higher rankings. The more highly ranked websites are linked to it. It's a trend on social networking sites to spread spam links [5

IV. PROPOSED METHODOLOGY

An extension for a browser is usually developed using web technologies like HTML, JavaScript, CSS and XML. Using these technologies, an extension to secure a user's browsing experience has been implemented. The browser concentrated for this extension is Google Chrome. The basic idea is to redirect any URL to the HTTPS version based on its domain name and if the redirection falls back then the extension checks if the URL is a spam. So while checking if the link is a spam, the user is then redirected to a safe website with a message and the URL is stored in a safe vault and thus the link is safely redirected to the original URL.

Websites, even if they use HTTPS are a bit unsecure because every website has their own separate domain. And here the redirection is to that secure domain rather HTTPS. This is the reason why a particular rule cannot be used for every website by changing the HTTP in the URL to HTTPS. Because that would only mean the change of protocol, whereas in reality it means that we should redirect to the secure website which sends data in an encrypted mode.

Initially, It is mandatory to first determine the domain of the URL and then based on the predefined set of rules we would redirect the user. The rules are written in an XML document which would be parsed by JavaScript. Only if a rule for the particular domain exists, it is redirected.

Example of a rule set: 1

```
<rules name=”google”>
<target host=”google.com”/>
<rule from="^http://encrypted\.google\.com/" to="https://encrypted.google.com/" />
</rules>
```

The above is an example of a rule set for http://www.google.com. Separate names are given to each rule to describe them uniquely. The 'target' tag is used to determine the domain name, which the extension would use to find out the website to be redirected. Here the redirection is to http://encrypted.google.com. This is a basic redirection from an unencrypted site to its HTTPS version of the original domain. This is just the initial version, as there is no functionality for redirection with parameters in the URL.

When the extension encounters a URL with parameters it is sliced and anything other than the main domain is saved for future parsing. Now the process is the same, but in addition to it when redirecting the URL, the parameters are concatenated to the redirecting secure URL. Hence JavaScript would help in parsing the URL's to secure versions with the parameters initially received from the basic version of the website. Following is an example for a rule defined for URL's with a parameter.

Example of a rule set: 2

```
<rules name="Wikipedia">
<target host="*.wikipedia.org" />
<rule
from="^http://([^@:/][^/:@])\.wikipedia\.org/wiki/"
to="https://secure.wikimedia.org/wikipedia/$1/wiki/"/>
</rules>
```

The above rule is for the domain Wikipedia where a URL http://fr.wikipedia.org/wiki/Chose is redirected to https://secure.wikimedia.org/wikipedia/fr/wiki/Chose. The 'from' and 'to' attribute in each rule are JavaScript regular expressions. They are used to rewrite URL's in a more complicated way. The parameter part of the URL is parsed and substituted by the JavaScript regular expressions if it matches the given wild card or expression. Hence now parameterized URL's are also redirected. So now a method to solve a fallback to the extension is required, the case where if there is no rules defined.

V. SPAM FILTERING PROCESS IN EXTENSIONS

If there exists no rules for a particular domain, then it might be a spam link. Whether the link is a spam or not is to be detected and then it is to be handled appropriately. If it is a spam, then the user is redirected to a safe page else the user is redirected to the original link, as it does not have a HTTPS support over the domain. As described in [4] (spam detection url.pdf), there are various factors which can be used to determine whether a link is a spam or not. For example, the factors include determining the initial and landing URL, the number of redirects, HTML redirects and page links, etc.

To minimize the network delay, a list of good domains is whitelisted which can be used to classify the good URL's from the spammed ones. A method to overcome the detection is to use a DNS resolver. Every URL is looked up for their hostnames, IP addresses, name servers and mail servers related with each and every domain. Each of the above features help in determining common infrastructure of spam links.

Following is a flow chart illustrating the process a URL undergoes, once it is encountered by the extension. This is a step by step process, from determining the spam URL to checking the rules for a domain. The flow chart is self-explanatory.

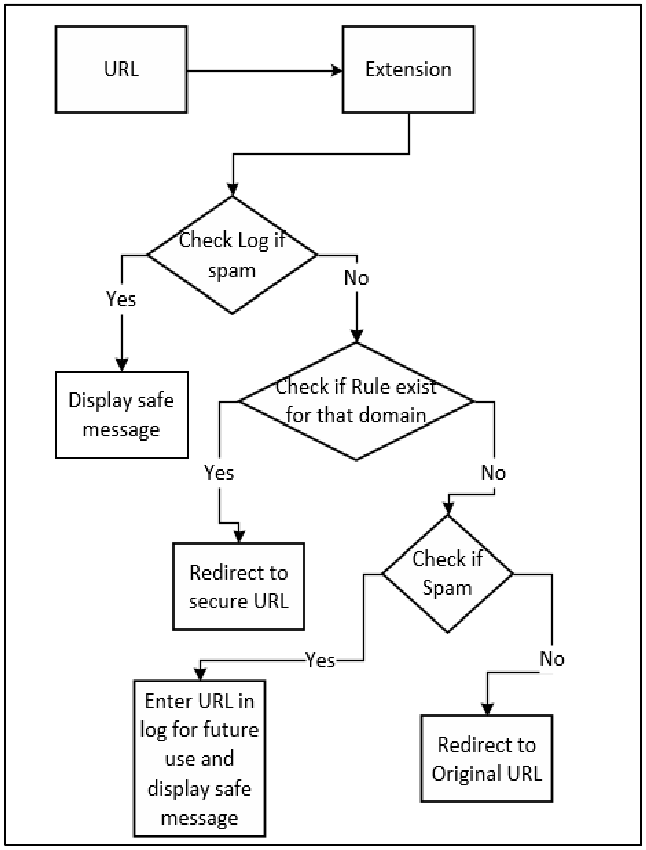

Fig. 1. Flowchart for Enhanced Link Redirection Interface for Secured Browsing.

The proposed idea plans on using a ranking procedure to determine whether a URL is spam or not. After having a look on different aspects of a web page, the page is given a score. Based on the scores, the URL is concluded to be legitimate or not.

Firstly, the URL takes Google PageRank into consideration. Secondly, whether the page exists or not is checked, that is the existence of PageRank, number of links on the page, links on the page are spam or of quality content, whether the page is indexed, the site is indexed and is the page dynamically loading and additionally if it is a RFC complied. Based on the characteristics, the scores are increased if there is a positive response from the web page and negative for any negative response. In the end if the scores lie between a particular low ranges, the link is declared spam and if it is not then it is declared legitimate.

The extension now determines the URL to be spam or not based on a set of scores. So if it is spam, the user is redirected to a safe website with a message conveyed that it was an unsafe website. Later, this link is added to the vault which is a log file for future references. Again if the same link is intercepted, there is no requirement to calculate the scores again but just to check the log file. And finally, if the link is not a spam but is a safe link with no support for HTTPS, the user is then redirected to the original link with no restrictions.

Very few websites don’t have support for HTTPS. And of the small set, are websites which don’t share user data or require user data. Hence these types of URL’s are redirected to the initial URL phase. These websites don’t have set of rules; hence they are tested for spam. Therefore, if they are legitimate links they are just redirected to the website.

Finally, among the many potential attacks that target Internet with spams or vulnerabilities in browsers, browsers which failed to protect the user from spams have received relatively little attention.

Hence using an extension to enhance the security of user data and their browsing experience would make a great impact in simplifying a user’s life rather than managing their data continuously [2].

VI. Future Works

This extension looks only into the link-redirection and spam detection. But whenever there is a URL redirection, the domains for which the cookies are stored are lost. Hence before the user is redirected, the cookies have to be analysed. Based on the new domain, the old cookies should be deleted and new set of cookies have to be created. This cookie exchange is necessary because the usual cookies are stored on the HTTP version of the website whereas the secure version is HTTPS.

There is a change in protocols, hence in the cookie exchange. This could be a future enhancement. Installing extensions can be cumbersome for every user as they have to go to the Chrome Store every time.

So implementing the extension functionality directly into a browser is a possibility. This feature can be useful for users who possess very less knowledge about securing personal and private data. This can be implemented on open source browsers like Chromium.

VII. Conclusion

Extensions add specific abilities into browsers which helps the user in solving many problems which the user cannot solve on their own. As extensions are just simple programs complementing functions of browsers, they take a very small amount of space, and still cover various aspects of data storage and data security. This extension is one way, of how simple programs can secure a user from malicious links and web crackers.

Acknowledgment

We thank Amrita University, India for providing the opportunity and facilities in all possible ways for this research. We thank all the reviewers of this paper for their helpful comments and suggestions.

References

[1] Ter Louw, Mike, Lim, JinSoon , Venkatakrishnan, V.N, “Extensible Web Browser Security”, detection of Intrusions and Malware, and Vulnerability Assessment, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007-01-01, Pages 1-19, http://dx.doi.org/10.1007/978-3-540-73614-1_1, DOI: 10.1007/978-3-540-73614-1_1

[2] N. Jovanovic, C. Kruegel, and E. Kirda. Pixy: A static analysis tool for detecting web application vulnerabilities (short paper). In Proceesings of the 2006 IEEE Symposium on Security and Privacy, pages 258–263, 2006.

[3] N. Freeman and R. S. Liverani. Exploiting cross context scripting vulnerabilities in Firefox, April 2010. http://www.security-assessment.com/files/whitepapers/Exploiting_Cross_ Context_Scripting_vulnerabilities_in_Firefox.pdf.

[4] Kurt Thomas, Chris Grier, , Justin Ma, Vern Paxson Y, Dawn Song, “Design and Evaluation of a Real-Time URL Spam Filtering Service”, University of California, Berkeley, International Computer Science Institute. Proceedings of the IEEE Symposium on Security and Privacy, May 2011.

[5] Devdatta Akhawe and Adrienne Porter Felt, “Alice inWarningland:A Large-Scale Field Study of Browser SecurityWarning Effectiveness”, University of California, Google, Inc. Usenix Security Symposium, Washington DC, 2013.

[6] Mike Ter Louw, Jin Soon Lim, V. N. Venkatakrishnan, “Enhancing web browser security against malware extensions”. Journal in Computer Virology 4(3): 179-195 (2008).

[7] ACCUVANT LABS security assessment and research, “Browser Security Comparison – A Quantitative Approach”, Pages 140, Version 0.0, Revision Date: 12/6/2011.

[8] Nikhil Swamy, Benjamin Livshits, Arjun Guha, Matthew Fredrikson, "Verified Security for Browser Extensions", Microsoft Research Technical Report, MSR-TR-2010-157.

[9] AGGARWAL, G., BURSZTEIN, E., JACKSON, C., AND BONEH, D. An analysis of private browsing modes in modern browsers. In Proceedings of the 19th USENIX conference on Security (2010), USENIX Security’10.

[10] Ross, B., et al.: Stronger password authentication using browser extensions. In: 14th USENIX Security Symposium (August 2005).

[11] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. On the potential of proactive domain blacklisting. In Proceedings of the Third USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET), San Jose, CA, USA, April 2010.