IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Technique to protect cookies using dual Protection Method

Prof. Swapnil Waghela
Department of Computer Science & Engineering, Acropolis Technical Campus, Indore, India

Prof. Nitisha Tiwari
Department of Computer Science & Engineering, SIRT, Sage University, Indore, India

***Abstract*-- Cross site scripting (XSS) is security vulnerability in World Wide Web. Information sharing between computers is increasing day by day. web security is most important topic to be discussed. Web applications often use cookies to maintain authentication between the user and web application where XSS attack is a popular attack to steal cookies between client and server. By using XSS technique, attacker insert malicious script onto the application's output. Some Web Application Vulnerabilities like Cross Site Scripting (XSS or CSS) and SQL Injections are happened because of poor input validation.**

In this paper we are proposing a new and light weight technique for "Cookies Alteration". This technique aims to repay the cookie useless for the attacker. Our technique is to implement a proxy that will recreate the cookie that is sent back and forth between the browser (Client) and server (web application). If attacker steals cookies then these cookies will be worthless for him. Our aim is not only to protect cookies but render the user's data secure.

***Keywords*-- Cookies, Web Proxy, XSS Attack**

I. INTRODUCTION

The Cookies are a mechanism to provide state full communication over HTTP[1]. Cookies are mainly used to store the session Id's or personal information in today's web applications. Cookies are sent by the web application as a part of the response message using Set-Cookie header. The browser stores cookie in its database, and includes the cookies with every subsequent request to the web application. This is shown in the diagram below.

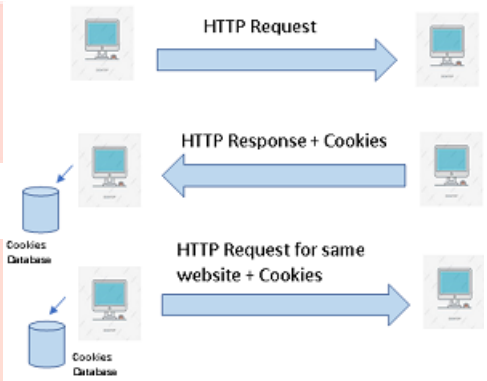

Figure 1.1
Shows how cookies from server are stored in database and also along with every request to subsequent page cookies from database are sent to server

Types of Cookies:

In general cookies can be classified into two types, this depends on a attribute of cookie that is their life time named as "expires" is represented by positive, or negative value [2][3].

Types are:

- Session cookies are temporarily used; they are discarded when the browser is closed. They expire attribute is set with negative value.
- Persistent Cookie can be kept longer until they expire; they are stored on a disk and survive across a computer restart.

II. BACKGROUND WORK

One simple way is to disable cookies, but it may cause the web server denying to work without the cookies. Other techniques to protect cookies from the attackers [4].

IP Mapping:

The web server maps IP addresses of users with cookies and denies any access that comes from invalid IP .This helps to shift problem but it does not work where the users access the Internet through web proxy.

Http Onlyattribute:

Http Only attribute is a Microsoft extension, it can also be included in the cookies before being sent to the browser. With the Http Only attribute, the browser deny scripting language to access those cookies. The Http Only attribute is originally not a part of HTTP; the browsers that are not aware of this attribute will ignore and it consequently remain vulnerable.

Secure cookies:

Secure cookies mean that the clients and the web servers only send the cookies via the SSL connections.

No solution mentioned above guarantee that the cookie will safe from XSS attack. We propose a new approach which aims not to protect the cookies but instead render the cookie unusable for the attackers.

III. PROBLEM DEFINITION

we are facing is to save cookies from XSS attack, we have two type of attacks that can be used to steel cookies from system, the potentiality of these attacks can be judged as even the firewall cannot stop the scripts from being executed and once they are executed there is no chance to save cookies, the cookies will be sent to attackers system whose address is embedded in the script. More over we still have an option, if we close the browser it will not execute, the greater problem is without our permission browser will execute those script and even after our cookies are sent to the attacker's computer we won't know anything about it.

XSS Attack

Cross Site Scripting (also known as XSS or CSS) is generally believed to be one of the most common application layer hacking techniques. In an XSS attack, a Web application is sent with a script that activates when it is read by an unsuspecting users browser or by an application that has not protected itself against cross-site scripting. Because dynamic Web sites rely on user input, a malicious user can input malicious script into the page by hiding it within legitimate requests. Once XSS has been launched, the attacker can change user settings, hijack accounts, poison cookies with malicious code, expose SSL connections, access restricted sites and even launch false advertisements.

- Non persistent (or reflexive XSS) means that malicious code is not persistently stored in a vulnerable server, but it immediately echoed by the vulnerable server back to victim. This can be understood by an example: suppose a person is accessing www.bank.com in order to do an online transaction , at the same time the victim might also be accessing www.attacksite.com, and be persuaded into clicking the below link

<a href ="http://www.bank.com/ <SCRIPT>
Document.location="http://www.attacksite.com/stealcookie.phr?'
+document.cookie;</SCRIPT>">
Click here to win a million Dollars

The script will get executed and the cookies of www.bank.com will be sent to www.attacksite.com. The owner of attack site can use cookies to impersonate www.bank.com with respect to the person.

- Persistent (or stored) XSS means that the malicious code is persistently store in a server's storage , and may later be embedded in an HTML page and send to the victim. This can also be explained by an example. Suppose a Script is posted on an online message board of www.bank.com.

Click Here To See A New Promotion
<SCRIPT>
Documentimage[0]=
http://www.attacksite.com/images.jpg?stealcookies+document.cookies;
</script>

The victim who reads a message will receive the malicious script as a part of message. The victim's browser will then execute the malicious script which will later send the cookies of www.bank.com to www.attacker.com

The main problem with this is we cannot stop script to run once it has enter with html script, browser will run the script and the coding of script will send our cookies to the attacker.

IV. SOLUTION DOMAIN

We propose a new technique for cookie protection named "Preventing XSS attack by cookie Modifying USING A PROXY SERVER", which we will implement as a part of web proxy. With this technique in place, the web proxy will automatically modify the value if the name attribute in the cookie with a randomly generated unique id, before sending the cookies to the browser, so the browser will keep the randomized value in its database instead of the original and the original value along with the randomized value as key will be saved in a hash table. The returned cookies from browser will be rewritten back to the original name/value sent by server and then forwarded to the server. Now as the browsers data base does not store the original value even if XSS attacker steals the cookies from the browser's database. The cookies cannot be used later to impersonate the users.

V. TERMINOLOGY USED

Web Proxy:

The proxy as an intermediary or a middleman that fulfills transactions on behalf of clients , many organizations allow the users only to access the internet via web proxy. It is an important control point for web surfing which is commonly built in with various security capabilities . The web proxy can be a separate device or a part of fire wall. It must sir between client and server, act as both the client to web server and as a server to the client i.e. the browser. All web connections from the client are intercepted at the web proxy, and then the web proxy will initiate new web connections to the web servers on behalf of the clients.

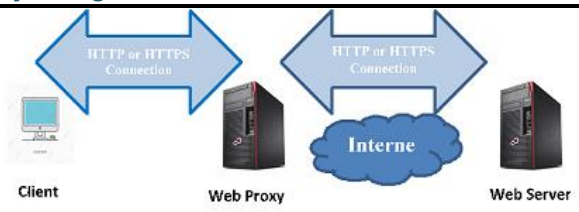

Figure 1.2
Shows position where proxy is to be implemented

- Install Web proxy at every client side.
- The Web proxy as an intermediary or a middleman that fulfills transactions on behalf of clients.
- If Client sent any HTTP request to Server then server create a Cookie for that particular request.
- Server send HTTP response message with generated cookie to web proxy as web proxy is working like an intermediary.
- Here web proxy will start its work and generate an encrypted or modified cookie corresponding to original cookie.
- Then Web proxy sends this modified cookie to client.
- Client store that modified cookie in its database.
- If any attack takes place at this time, only modified cookie will be display which will not be fruitful for attacker.

VI. CONCLUSION

The proposed mechanism for Protecting Cookies from cross site scripting attack has guarantee that the cookie will safe from XSS attack. Cookies will be safe from Persistent XSS attack and Non-Persistent XSS attack.

REFERENCES

1. Persistent dom-based xss in https://help.twitter.com via localstorage, 2018, https://hackerone.com/reports/297968.
2. P. Bisht and V. Venkatakrishnan, "XSS-guard: precise dynamic prevention of cross-site scripting attacks," in Proceedings of the International Conference on Detection of Intrusions and Malware, pp. 23–43, Springer, 2008.View at: Google Scholar
3. N. Jovanovic, C. Kruegel, and E. Kirda, "Pixy: A static analysis tool for detecting web application vulnerabilities," in Proceedings of the 2006 IEEE Symposium on Security and Privacy, p. 6, IEEE, 2006.View at: Google Scholar
4. T. Jim, N. Swamy, and M. Hicks, "Defeating script injection attacks with browser-enforced embedded policies," in Proceedings of the 16th International Conference on World Wide Web, pp. 601–610, ACM, 2007.View at: Google Scholar
5. S. Stamm, B. Sterne, and G. Markham, "Reining in the web with content security policy," in Proceedings of the 19th International Conference on World Wide Web, pp. 921–930, ACM, 2010.View at: Google Scholar
6. M. Van Gundy and H. Chen, "Non spaces: Using randomization to defeat cross-site scripting attacks," Computers & Security, vol. 31, no. 4, pp. 612–628, 2012.View at: Publisher Site | Google Scholar
7. E. Athanasopoulos, V. Pappas, A. Krithinakis, S. Ligouras, E. P. Markatos, and T. Karagiannis, "xjs: practical XSS prevention for web application development," in Proceedings of the 2010 USENIX Conference on Web Application Development, pp. 13–13, USENIX Association, 2010.View at: Google Scholar
8. Binishala, amazon.co Security Vulnerability, 2016, https://www.openbugbounty.org/incidents/152371/.
9. Brute, ebay.com security vulnerability, 2016, https://www.openbugbounty.org/incidents/121171/.
10. L. O'Donnell, https://hackerone.com/reports/297968, 2018.
11. T. Adams, Reflected XSS in Tooltipy (Tooltips for Wp) Could Allow Anybody to Do Almost Anything an Admin Can, 2018, https://advisories.dxw.com/advisories/xss-in-tooltipy/.
12. Akamai, State of The Internet: Security - Web Attack Report Infographic, 2018, https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-summer-2018-web-attack-infographic.pdf.
13. T. Scholte, D. Balzarotti, and E. Kirda, "Quo vadis" A study of the evolution of input validation vulnerabilities in web applications," in International Conference on Financial Cryptography and Data Security, pp. 284–298, Springer, 2011.View at: Google Scholar
14. G. S. Kc, A. D. Keromytis, and V. Prevelakis, "Countering code-injection attacks with instruction-set randomization," in Proceedings of the 10th ACM Conference on Computer And Communications Security, pp. 272–280, 2003.View at: Google Scholar
15. W3C, Content Security Policy Level 2, 2016, https://www.w3.org/TR/CSP2/.
16. M. Bishop, M. Dilger et al., "Checking for race conditions in file accesses," Computing Systems, vol. 2, no. 2, pp. 131–152, 1996.View at: Google Scholar
17. MITRE,"Cwe-367:Time-of-checktime-of-use(toctou)race condition," https://cwe.mitre.org/data/definitions/367.html.View at: Google Scholar
18. S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats, vol. 54, Springer Science & Business Media, 2011.
19. J. H. Jafarian, E. Al-Shaer, and Q. Duan, "An effective address mutation approach for disrupting reconnaissance attacks," IEEE Transactions on Information Forensics and Security, vol. 10, no. 12, pp. 2562–2577, 2015.View at: Publisher Site | Google Scholar
20. P. Team, Pax Address Space Layout Randomization (aslr), 2003.
21. A. Klein, "Dom based cross site scripting or xss of the third kind," Web Application Security Consortium, Articles, vol. 4, pp. 365–372, 2005.View at: Google Scholar
22. T. Pietraszek and C. V. Berghe, "Defending against injection attacks through context-sensitive string evaluation," in Proceedings of the International Workshop on Recent Advances in Intrusion Detection, pp. 124–145, Springer, Berlin, Germany, 2005.View at: Publisher Site | Google Scholar
23. I. Papagiannis, M. Migliavacca, and P. Pietzuch, "PHP aspis: using partial taint tracking to protect against injection attacks," in Web Apps' 11: Proceedings of the 2nd USENIX Conference on Web Application Development, pp. 13–24, USENIX Association, 2011.View at: Google Scholar
24. P. Vogt, F. Nentwich, N. Jovanovic, E. Kirda, C. Kruegel, and G. Vigna, "Cross site scripting prevention with dynamic data tainting and static analysis," in Proceedings of the NDSS, vol. 2007, p. 12, 2007.View at: Google Scholar
25. P. Wurzinger, C. Platzer, C. Ludl, E. Kirda, and C. Kruegel, "SWAP: Mitigating XSS attacks using a reverse proxy," in Proceedings of the 2009 ICSE Workshop on Software Engineering for Secure Systems, SESS 2009, pp. 33–39, Canada, May 2009.View at: Google Scholar
26. E. Kirda, C. Kruegel, G. Vigna, and N. Jovanovic, "Noxes: A client-side solution for mitigating cross-site scripting attacks," in Proceedings of the 2006 ACM Symposium on Applied Computing, pp. 330–337, France, April 2006.View at: Google Scholar
27. G. Wassermann and Z. Su, "Static detection of cross-site scripting vulnerabilities," in Proceedings of the 30th International Conference on Software Engineering, pp. 171–180, ACM, 2008.View at: Google Scholar
28. M. Ter Louw and V. Venkatakrishnan, "Blueprint: Robust prevention of cross-site scripting attacks for existing browsers," in Proceedings of the 2009 30th IEEE Symposium on Security and Privacy, pp. 331–346, IEEE, 2009.View at: Google Scholar

29. J. Weinberger, A. Barth, and D. Song, "Towards client-side HTML security policies," in HotSec, 2011.View at: Google Scholar

30. S. W. Boyd, G. S. Kc, M. E. Locasto, A. D. Keromytis, and V. Prevelakis, "On the general applicability of instruction-set randomization," IEEE Transactions on Dependable and Secure Computing, vol. 7, no. 3, pp. 255–270, 2010.View at: Publisher Site | Google Scholar

31. W. W. W. C. (W3C), Html5 a vocabulary and associated apis for html and xhtml. W3C recommendation 28 October 2014, 2014, https://www.w3.org/TR/html5/scripting-1.html.

32. Enhancesoft, OS ticket - Support Ticket System, 2016, http://osticket.com/.

33. Os Commerce, oscommerce, 2016, http://oscommerce.com/.

34. I. Automattic, Wordpress, 2016, http://wordpress.com/.

35. Rochen, Joomla!, 2016, http://joomla.com/.

36. R. Hansen, Xss filter evasion cheat sheet, 2018, https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet.

37. Akamai, *Akamais[StateOf the internet]q32016Report*, 2017, https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q3-2016-state-of-the-internet-connectivity-report.pdf.

38. Pylo, *Reloadmaticadd-on*,2018, https://addons.mozilla.org/enUS/firefox/addon/reloadmatic/.

39. M. Weissbacher, T. Lauinger, and W. Robertson, "Why is CSP failing? Trends and challenges in CSP adoption," in Proceedings of the International Workshop on Recent Advances in Intrusion Detection, vol. 8688, pp. 212–233, Springer, Berlin, Germany, 2014.View at: Google Scholar

40. Binishala, amazon.com Security Vulnerability, 2016, https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/script-src.

41. R. Ausbrooks, S. Buswell, D. Carlisle et al., "Mathematical markup language (mathml) version 2.0. w3c recommendation," in Proceedings of the World Wide Web Consortium, vol. 2003, 2003.View at: Google Scholar

42. J. Ferraiolo, F. Jun, and D. Jackson, Scalable Vector Graphics *(SVG) 1.0 Specification*, Iuniverse, 2000.

43. D Gourley , B Totty M Sayer, S Reddy and A Aggrawal, HTTP the definitive Guide 1st elderly media US 2002

44. J. Gareio-Alfaro and G Navarro-Arribas "Prevention of Cross site scripting attacks on current web application" Available :

45. http://hacks-galore.org/guille/pubs/is-otm-07.pdf

46. Suman Saha, Dept. of Computer Science and Engineering, Hanyang University, Ansan, South Korea, Consideration Points: Detecting Cross-Site Scripting, IJCSIS Vol. 4 2009

47. D.Kristol "HTTP State Management Mechanism" in The Internet Society 2000 Available at :http://www.ietf.org/rfc/rfc2965.txt

48. A. Krithinakis, S. Ligouras, E. P. Markatos T. Karagiannis a n d E. Athanasopoulos, "xjs: Practical XSS prevention for web application development," In Proceedings of the USENIX

49. Conference on Web Application Development (Web Apps), 2010

50. Y. Sun and D. He, \Model checking for the defense against cross-site scripting attacks," in Computer Science Service System (CSSS), 2012 International Conference on, Aug 2012, pp. 2161{2164}.

51. Wenjun Wang, Jianmeng Li, The Security Threats and Prevention of Web Applications: Base on the OWASP Top 10 and ESAPI, Beijing,2013, pp. 232-261.

52. M. Van Gundy and H. Chen, "Non spaces: using randomization to defeat cross-site scripting attacks," International Journal of Computer Security,vol.31,no.4,pp.612–628,2012

53. P. Sharma, R. Johari, and S. S. Sarma, "Integrated approach to prevent SQL injection attack and reflected cross site scripting attack," International Journal of System Assurance Engineering and Management, vol. 3, no. 4, pp. 343–351,2012.

54. Y. Sun and D. He, "Model checking for the defence against cross-site scripting attacks," in Proceedings of the 2012 International Conference on Computer Science and Service System, pp. 2161–2164, Maui, Hawaii, January 2012.

55. H. Isatou, S. Abubakr, Z. Hazura, and A. Novia, "An approach for cross site scripting detection and removal based on genetic algorithms," in Proceedings of the Ninth International Conference on Software Engineering Advances: France, pp. 227–232, Nice, France, October 2014.

56. Nithya, P. Lakshmana, and C. Malarvizhi, "A survey on detection and prevention of cross-site scripting attack," International Journal of Security and its Applications, vol. 9,no. 3, pp. 139–152, 2015.