A Novel Approach for XSS Vulnerability in Web Threats

Jenish R. Shah[1] Vishal R. Andodatiya[2] Sohil Gadhiya[3]
[1]P G Student [2,3]Assistant Professor
[1,2,3]Department of Computer Engineering
[1,2]Shree Pandit Nathulalji Vyas Technical Campus, India [3]C U Shah College Of Engineering and Technology, India

Abstract— Cross-Site Scripting is one of the major's attacks described by OWASP. The Cross Site Scripting attack is possible by inserting or changing the programming logic, changing and syntax of HTML elements by code injection attacks. The Web application is XSS Vulnerable when there is no proper input validation. The many web applications like social networking sites are the victims of this attack. This paper describes the various attack technique of cross site scripting and various mitigation involved in cross site scripting (XSS) Vulnerability. As new technology arrived, it comes with lot of new features and possibly attacks. In the today's trends of social web application, Web Forum and other user content driven sites, the SQL injection, cross site scripting (XSS) attack and cross site forgery attack are major challenges for web application. The paper also describes various research perspective involved with cross site scripting. In the cyber world the security is now main issues for the user. The paper also shows demonstration of various cross site scripting (XSS) attack.

Key words: Cross Site Scripting, Boundary injection, XSS server side detection, XSS Vulnerability , XSS detection using Jericho Parser.

I. INTRODUCTION

Today Web Application seen become more popular. Using web application we can done all things likes money transfer, online shopping, Social network, Blogging, Wiki Reviews, Study related information and many more. As an increase the use of Web Application, We required more Security against web threats. Security in the terms of Protect a sensitive data of any user against unauthorized user. Web threats [6] is any threats that uses the World Wide Web to facilitate cybercrime. It is use multiple types of malware and fraud, all which utilized HTTP and HTTPS protocols, but may also employ other protocols and components, such as links in email or IM, or malware attachments or on servers that access the Web. With the increase in the trend web application are becoming vulnerable for attacks. Web application threats like Spam, malware, hacking, phishing, denial of service attacks, invasion of privacy, frauds [2]. Mainly attack on web server mainly performed on server side code and supporting library. Attacks are basically buffer over flow, input validation attacks, format string attack, canonicalization attack, encoding attacks, privilege escalation, form tempering and user generated content. Another new attack like Cross Site Scripting (XSS), SQL injection, insecure direct object reference [1]. XSS cause danger for insertion of a piece of script on client side. Here mainly used JavaScript and this attack can also deployed through a link in an email or on a web page that appears to be originated from hacker's site.

Cross Site Scripting is a type of computer security vulnerability typically found in Web Applications. XSS enables attackers to inject client-side script into Web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy [9].

Cross site scripting is the most basic attack on web application. It provide the surface for other type of attack like Cross site Request Forgery, Session Hijacking etc.
Here mainly three type of XSS which is described below.

- Stored or Persistent XSS
- Reflected or Non- Persistent XSS
- DOM Based XSS

In stored or persistent XSS [5] attacks the injected malicious code is permanently stored on the target server. In this type of attack, attacker first tries to find vulnerability in web application. If such vulnerability is present in web application, he injects a malicious script that will be able to getting user information without Authentication. They get user confidential information or cause other damage. This script reside permanently on the server side. When user access the information via web application, the malicious script gets executed and the confidential information becomes accessible to attacker. Mainly stored XSS perform on web application that take input form of text and store it in the database of web application. Ex. Blogs, forms, comments or profile.

As opposed to Stored XSS attacks [5] in reflected XSS attacks the injected code doesn't reside on the Web server. In reflected attacks malicious link sent to the user using via email or embedding the link in a web page residing on anther server. When user click on the link, the injected code goes to attacker's web server, which sends the attack back to victim's browser. Now browser execute the code because it come from a trusted server. In this way on attacker bypass the some origin policy. When this code execute on the browser, it perform the malicious work like staling the confidential information of victims.

DOM [4] based XSS is an XSS attack where DOM environment in the victim's browser is modified by the original client side script. So that the client side code gets execute in a different manner because if the modification of DOM environment. It is different from the other two XSS attack is executed at the client side.

II. LITERATURE SURVEY

For this currently top ten list out of Web Threats published by CISCO-2014. When Analysis of Cisco Report between 2011 to 2014.We Found Many web threats but Cross Site Scripting Maintain its Position in Top ten Web Threats which is Given below in CISCO-2014 Report. In Perl based Method, user select critical system file or any other files whose integrity is important, and computing a hash of those file system file to be established base file. Once the base file has been established, at the same point in time, hashes of those selected file can be recomputed. If file was not modified in any way the hash value remain same. Here the

process of identifying all <script> and <img> tagged content will then be replaced and new hash values for all potential client side executable content. If hash value matched, it means no new element of client side executable content have been detected, which is indicative that the web page not likely contain any XSS. If hash value not matched it means client side value becomes change. Hash based IDE System, it is important to keep the baseline values up to date ensure that rightfully modified content does not trigger false alarms. The one XSS attack vector that want undetected contain null character (\0) in the script tag which made the tag unrecognizable to the IDS and can be accounting for the potential insertion of null characters in XSS attack vectors [15].

S2Xs2 they derived automated framework to detect XSS attacks at the server side based on the notation of boundary injection and policy generation. They develop a prototype tool to automatically insert boundary and generate policy for JSP Page. The limitation of this paper is 1. Modified both server interpreter and 2. Java-script interpreter. Here boundary is an HTML or Java-Script comment that does not alter expected program output or behavior. They implement a prototype tool in java. They pares both JSP and HTML source code. They use Jerico [12] HTML parser has been get sources and DOMs of the parse pages. Jerico provide API to access and manipulate DOM nodes that related JSP features.

Rhino [13] parser is used to identify feature of Java-Script Code blocks. And they policy of attack detection in a web container which is stored on the server side filter. Here they defined mainly constant string [18], full tag [18], partial tag [18], unknown tag [18] etc.

III. Related Work

All above method of Cross site scripting that derives for the detection of the XSS attack in defiant manner. As consider base paper as X2SX2 [18] we will developed a novel approach for XSS Detection.In X2SX2 [18] they work on boundary detection for html comment (<!-- . -->) and JavaScript comment (/*….. */) and they add boundary with respect to <% … %> tags. X2SX2 [18] they detect only XHTML tags. It means they follow the fully tag structure, like opening tag and closing tag. Their approach mainly detect Constant String, Full tag, Partial tag, Unknown tag which the form in the XHTML tagged Structure. In a proposed work we found that this X2SX2 [18] approach not working for encoded data which is enter by the attacker. And use of HTML5 Tag <EMBED> and <SVG> attacker can easily pass the malicious code.

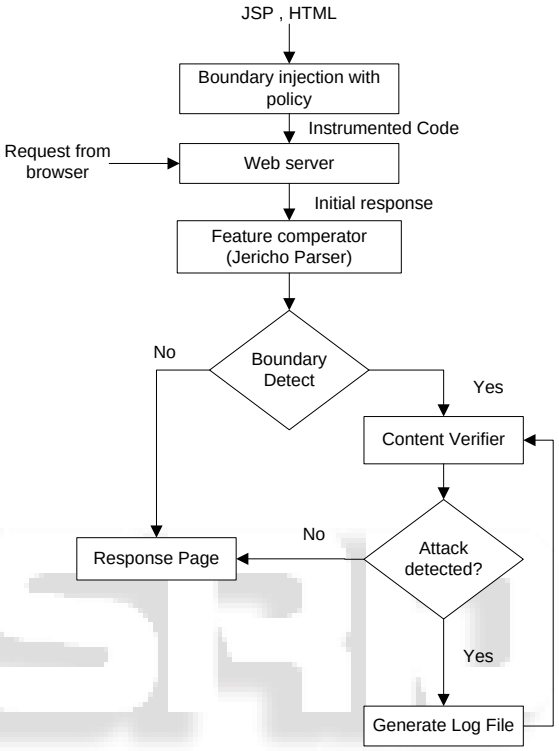
<embed
src="
c9Imh0dH0A6Ly93d3cudzMub3JnLzIwMDAvc3ZnIiB4b
Wxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciI
HhtbG5zOnhsaW5rPSJodHRwOi8vd3d3LnczLm9yZy8xO
Tk5L3hsaW5rIiB2ZXJzaW9uPSIxLjAiIHg9IjAiI
HdpZHRoPSIxOTQiIGhlaWdodD0iMjAwIiBpZD0ieHNzIj
48c2NyaXB0IHR5cGU9InRleHQvZWNtYXNjcmlwdCI+Y
WxlcnQoIlhTUyIpOzwvc2NyaXB0Pjwvc3ZnPg==">type="i
mage/svg+xml"AllowScriptAccess="always">
</embed> [17]

<script>alert(“hello XSS”)<script>

So, we reduce this type of attack with respect to flow diagram which is described below.

IV. Proposed Work

In this Proposed Work, if we detect attack as per X2SX2 [18] guideline control moves to attack handler but if attack is not detected than we check the content of Data. In a contain we found the attack than move to attack handler Otherwise generate Log File And Again Check the Next Content Using Content Verifier.



Procedure:

- Inject Static boundary on <%= …… %> or <% …… %> and Println(“ ”) method When Page is developed by Developer .
- As a Boundary use HTML Comment <!-- …. -- > For Jsp page.
- Generate response page.
- Give URL to the Parser.
- Parser will got the Source.
- Retrieve Specific Tag Detail.
- Check with boundary is Exists than Get Tag Detail and Count Total number of Tag between.
- Display log in the Terms of (0,0) to (NUMBER_OF_TAG,ATTRIBUTED_VALUE).
- Classify the newly arrived INPUT VALUE to check whether the data is legal OR Malicious.

V. Conclusion

From studying various paper regarding Cross Site Scripting – XSS attack and increasing the vulnerability on the web threats to access confidential information without authenticated user they retrieve information. So we will try to reduce vulnerability of XSS based on server side detection technique. Implementation of boundary injection and policy checking using Jericho parser with the language Java. We found some issues in boundary injection

technique with persistent data on server side. And its check policy every time when boundary is detect. And at that time it's detect pure Html tag.it is not also detect encoded form Tag. And if we inject data with bounded specific boundary it is fail to check which type of content between two boundaries. For reduce this type of issue and reduce number of time policy checking we need to implement policy at the time of insertion.

REFERENCES

[1] D. Watson, "Web application attacks," Network Security, vol. 2007 issue 11, pp. 7-12, November 2007
[2] W. Kim, O. Jeong,C. Kim, and J. So, "The dark side of the internet:attacks, costs and responses", Information Systems, vol. 36, pp. 675-705, 2011
[3] http://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf
[4] https://www.owaspp.org/index.php/DOM_Based_XSS
[5] https://www.owaspp.org/index.php/Cross-site_Scripting_(XSS)
[6] en.wikipedia.org/wiki/Web_threat
[7] en.wikipedia.org/wiki/SQL_injection
[8] code.tutplus.com
[9] en.wikipedia.org/wiki/Cross-site_scripting-15/12/2014
[10] www.excess-xss.com
[11] http://www.oracle.com/technetwork/java/javamail/index.html
[12] Jerico HTML parser http://jerico.htmlparser.net
[13] Rhino, accessed from http://www.mozila.org/rhino
[14] Hiroya Takahashi, Kenji Yasunaga, Masahiro Mambo, Kwangjo Kim, Heung Youl Youm -"preventing abuse of cookie stolen by XSS"- 2013 Eight Asia joint Conference on Information Security, © 2013 IEEE, DOI 10.1109/ASIAJCIS.2013.20.
[15] Christoper M. Frenz, Jong P. Yoon- "XSSmon: A perl Based IDE for the detection of potential XSS Attacks" - 978-1-4577-1343-9/12 © 2012 IEEE.
[16] Vikas K. Malviya, Sanket Saurav,Atul Gupta- "On Security Issues in Web Application through Cross Site Scripting (XSS)"- 2013 20th Asia-Pacific Software Engineering Conference, 1530-1362/13 2013 © IEEE DOI 10.1109/APSEC.2013.85 .
[17] Guowei Dong, Yanzhang, Xin Wang, Peng Wang- "Detecting Cross site scripting vulnerabilities introduced in HTML 5". - 2014-11th International joint conference on Computer Science and software Engineering (JCSSE), 978-1-4799-5822-1/14 © 2014 IEEE.
[18] Hossain Shahriar and Mohammad Zulkernine - "S2XS2 : A Server Side Approch to Automatically Detect XSS Attacks"- 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, 978-0-7695-4621-4/11 © 2011 IEEE.
[19] http://www.phloxblog.in/jericho-html-parser-simple-html-parsing/#.VO9R-_mUem0
[20] http://www.java2s.com/Tutorial/Java/0120__Development/ParseHTML.htm
[21] http://www.acunetix.com/websitesecurity/cross-site-scripting/
[22] http://www.tutorialsavvy.com/2012/12/rhino-javascript-library-for-java.html/
[23] http://character-code.com/
[24] http://jericho.htmlparser.net/samples/console/src/FindSpecificTags.java
[25] http://www.base64-image.de/step-2.php

