Overview of detection methods of Cross site scripting Attack

Pratibha Kamal, Bharti Nagpal, Radhika Murari
Ambedkar Institute of Advanced Communications Technology & Research
Geeta Colony, New Delhi-110031

Abstract— **Recently, attacks against web application, such as SQL injection and cross site scripting, tend to increase. In this paper, we have done the analysis on different approaches of detection techniques against cross site scripting attacks by extracting an attack feature of cross site scripting attacks . We prepared samples for learning , to show the effectiveness of each approach. As the result, our analysis on detection method was successfully detected 99.5% attack test samples and 97.5% normal test samples. Web applications often use cookies for maintaining an authentication state between users and web applications, these cookies are typically sent to the users by the web applications after the users have been successfully authenticated. Every subsequent request that contains the valid cookies will be automatically allowed by the web applications without any further authentication. The cookies are used to both identify and authenticate the users; therefore they are an interesting target for potential attackers. Cross Site Scripting attack (XSS for short) is one of popular attacks which is often used to steal the cookies from a browser's database. In this paper we have shown different features of detection techniques used to protect we applications.**

Index Terms—**Cross-site scripting attacks. Documents objects model, detection techniques, hyper text markup language, security vulnerabilities.**

I. Introduction

'XSS' also known as 'CSS' (Cross Site Scripting, Easily confused with 'Cascading Style Sheets')is a very common vulnerability found in Web Applications, 'XSS' allows the attacker to INSERT malicious code,There are many types of XSS attacks, We have mention three most used types of cross-site scripting attack . in this attack attacker use to attack using scripting tags.

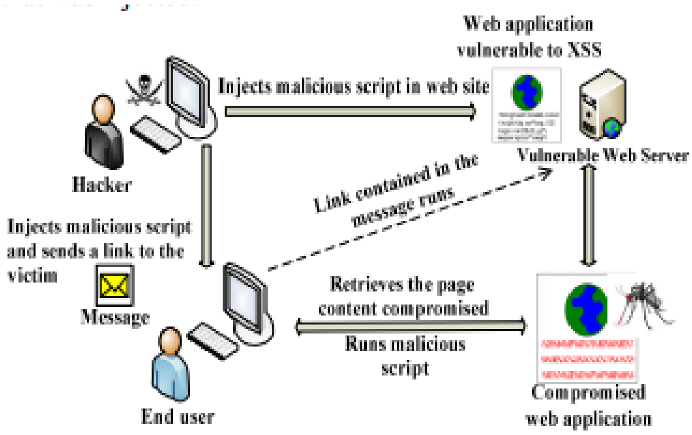

Web application vulnerable to XSS
Injects malicious script in web site
Hacker
Vulnerable Web Server
Link contained in the message runs
Injects malicious script and sends a link to the victim
Message
Retrieves the page content compromised
Runs malicious script
End user
Compromised web application

Figure1: Overview of XXS Attack[1]

The First Attack I want talk about is 'URL XSS' this means that the XSS won t stay on the page it will only get executed if you have the malicious code in the URL and submit the URL we will Talk more on how to use this in our advantages. The Second Attack is input fields, Where ever you can insert data, it is very common, to be XSS vulnerable, for example say we found a site with a search engine, Now in the search box you enter 'hacker' now hit enter, when the page loads, if it says your data like 'Found 100 Results ok now you see its displaying out data on the page, now what if we can execute code? There is no possible way to execute PHP code in this Attack, but certainly is for HTML, JavaScript, but be aware this method, Also won t stay on the server, this is for your eyes only.Dynamic web applications play an important role in providing resources manipulation and interaction between clients and servers. The features currently supported by browsers as HTML tags, scripts, hyperlinks, and advanced functions.

II. Types of XSS

To prevent the script code contained in a document loaded from some Web site accesses documents loaded from some other Web site, browsers do not allow to access between documents loaded from different site that is between client and the server Therefore attackers use another techniques to implement cross site scripting. There are three main categories to explain cross site scripting attack.

- Reflected XSS attacks
- Stored XSS attacks

- DOM XSS attacks

A. Reflected XSS:

The most common type of cross-site scripting exploit is the reflected exploit. It targets to vulnerabilities that occur in some Web sites when data submitted by the client is immediately processed by the server to generate results that are then sent back to the client system. This attack is successful if it can send source code to the server that is included in the Web page results sent back to the client, and when those results are sent the code is not encoded using HTML special character encoding thus being interpreted by the browser rather than being displayed as invert visible text.[7]

B. Stored XSS:

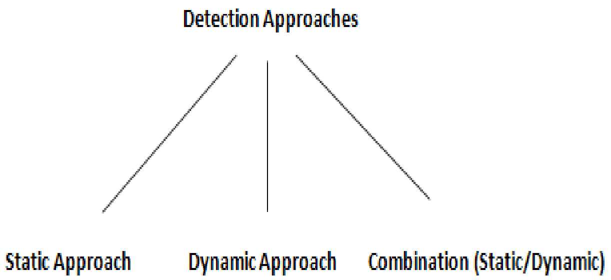
Also known as HTML injection attacks, stored cross-site scripting exploits where some data sent to the server is stored (typically in a database) to be used in the creation of the page that will be served to another user.[7] Stored XSS occurs when a web application gathers input from a user which might be malicious, and then stores that input in a data store for later use. The input that is stored is not correctly filtered. As a consequence, the malicious data will appear to be part of the web site and run within the user's browser under the privileges of the web application. Since this vulnerability typically involves at least two requests to the application, this may also called second-order XSS.

C. DOM-based XSS:

It is the third type of attack, where the logic errors in legitimate Java Script and careless usage of client-side data. In DOM- based XSS, the vulnerability based on the Document Object Model of the page. This type of attack occurs when the java-script in the page access a URL parameter and use this information to write HTML to the page. Developers and site maintainers need to familiarize themselves with techniques to detect DOM Based XSS vulnerabilities**.**

III. DETECTION TECHNIQUES

Cross-site scripting attack have three approaches of detection techniques to secure the data from unwanted attack over internet. They are static, dynamic, combination of static and dynamic.



1. Static approach
2. Dynamic approach
3. Combination of static and dynamic approach

Every technique have its own properties to fortify the malicious code to enter into any web application. As we know in cross site scripting attack the attacker use to add malicious code using scripting tags in the web application code. So we have identified some of the symbols that are used while injecting malicious code in the source code of any page.

Variables	**Symbols**
S1	“ (double quotation mark)
S2	> (greater than sign)
S3	/ (slash)
S4	< (less than sign)
S5	= (equal)
S6	, (single quotation mark)
S7	: (colon)

TABLE 1 DIFFERENT SYMBOLS

A. Static Approach

Bounded Model Checking:
In this approach Huang has been proposed a technique to minimize the number sanitation routines inserted and to identify the reason of error that enhance the error report.

Software Testing Approach:
This approach combines user experience modeling as black box testing and user- behavior simulation but they cannot guar-tee the detection of all the flaws.

Taint Propagation approach:
This kind of analysis is being used in so many dynamic and static approaches, they use data flow analysis to track movements of information from flow origin to end.

- In this approach there are some assumption and it is not considered a good idea to have faith on the user and not perform the sanitation function.
- As there can the some XXS attack script that can bypass many filter considered to be strong.
- Strong mechanism is not provided.

B. Dynamic Approach

Syntactical approach:

- Su and Wasserman suggested that when there is a successful injection attack there **is a change in the syntactical structure** of the exploited entity.
- So, they have presented a syntactical approach to detect the malicious code from string output of that syntactical approach.

Proxy-based approach

- A web proxy could be used to prevent transferring of any sensitive information from a victim's site to any other site.
- Malware is detected and it use to block that page or link by this application- level firewall.
- But it can not sufficient to blacklist a link to prevent cross site scripting attack.

Interpreter-based approach

- This approach was introduced to track un-trusted data at the character level and for identifying vulnerabilities that use context-sensitive string.
- This techniques is good and also able to detect vulnerabilities as security assurance is added by modifying the interpreter.

C. Combination of Static and Dynamic approach

Lattice-based approach

- This uses a tool called WebSSARI, which combines the static as well as dynamic approaches.
- It is used to detect the vulnerability by using taint propagation analysis.
- When this tool knows that taint data has reached sensitive function, it automatically puts runtime guards which are also called as sanitization routine.
- There is a big drawback of ths approach that there is a large number of negative and positive intra-procedural type-based analysis.

As web-based applications have become more sophisticated, the types of vulnerabilities are capable of exploiting has rapidly increased. A particular class of attacks commonly referred to as "code insertion" and often "Cross-Site Scripting" has become increasingly popular. Unfortunately, the number of applications vulnerable to these attacks is staggering, and the varieties of ways attackers are finding to successfully exploit them is on the increase. Analysis of many sites has indicated that not only are the majority of sites vulnerable, but they are vulnerable to many different methods and much of their content is affected.

Detection Approaches	Types	Featues	Advantages
Static	Bounded Model Checking	Use to minimize the number sanitation routines.	It use to enhance the error report
	Software Testing	It uses black box testing and user-behavior simulation	Use to black list the anonymous user using the website
	Taint Propagation	It uses data flow analysis to track movements of information from flow origin to end.	Data flow approache is used to track the malicious attacker
Dyamic	Syntactical	If there is a successful injection attack there is a change in the syntactical structure	they have presented a syntactical approach to detect the malicious code from string output
	Proxy-based	It used to prevent transferring of any sensitive information from a victim's site to any other site	Malware is detected and blocked by this application-level firewall.
	Interpreter	Use to track un-trusted data at the character level and for identifying vulnerabilities that use context-sensitive string	This techniques is good and also able to detect vulnerabilities as security assurance is added by modifying the interpreter
Combination(static/dynamic)	Lattice-based approach	It detect the vulnerability by using taint propagation analysis	When this tool knows that taint data has reached sensitive function, it automatically puts runtime guards which are also called as sanitization routine

Table 2: Analysis of different detection approach

Above are all the features and advantages of all the detection approach use to protect web application from cross site scripting attack.

I. CONCLUSION

From the above analysis we have identified that proxy-based detection technique is good and the best approach to secure web application from injecting malicious code in the system, it use to detect malware and blocked by this application- level firewall.

REFERENCES

[1] Angelo Eduardo Nunan, Eduardo Souto, Eulanda M. dos Santos, Eduardo Feitosa "Automatic classification of cross-site scripting in web pages using Documents-baesd and URL-based Features,"*institute of computing*, 702-707 , 2012.

[2] Xiaowei Li and Yuan Xue"A Survey on Web Application Security-Department of Electrical Engineering and Computer Science, ACM Transactions on Computing Surveys,November 2013

[3] Dr.JayamsakthiShanmugam, Dr. M. Ponnavaikko,"Cross Site Scripting-Latest developments and solutions: A survey",Int. J. Open Problems Compt. Math., Vol. 1, September 2008

[4] Seny Kamara, Sonia Fahmy, Eugene Schultz, Florian Kerschbaum, and Michael Frantzen"Analysis of Vulnerabilities in Internet Firewalls" Center for Education and Research in Information Assurance and Security (CERIAS) Purdue University, IN 47907–2039, USA

[5] IndraniBalasundaram,Dr.E. Ramaraj "An Approach to Detect and Prevent SQL Injection Attacks in Database Using Web Service", International Journal of Computer Science and Network Security, VOL.11 No.1, January 2011

[6] Jane Jaleel Stephan, SahabDheya Mohammed, and Mohammed Khudhair Abbas, "Neural Network Approach to Web Application Protection" International Journal of Information and Education Technology, Vol. 5, No. 2, February 2015.

[7] S.SHALINI, S.USHA"Prevention Of Cross-Site Scripting Attacks (XSS) On Web Applications In The Client Side", International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011.

[8] Sayyed Mohammad SadeghSajjadi and BaharTajalli Pour,"Study of SQL Injection Attacks and Countermeasures"International Journal of Computer and Communication Engineering, Vol. 2, No. 5, September 2013

[9] Vaishali Malekar,Prof. J.M Waghmare"WebApplication Firewall to Protect Against Web Application Vulnerabilities: A survey and comparison International Journal of Computer Technology & Applications, Volume 4 (1), Jan-Feb. 2013.

[10]Nilesh Kochare , Satish Chalurkar, B.B.Meshram " web application vulnerabilities detection techniques survey", International Journal of Computer Science and Network Security, VOL.13, June 2013.

.