ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (299-302), Month: January - March 2015, Available at: www.researchpublish.com

A Review On Web Application Attacks

Mani Sharma

Department Of Computer Science & Engineering
Mahatama Gandhi Mission's Engineering College, Noida, India

***Abstract*: Web applications are the vast source of exchanging information for the users.In order to provide more service to the users,it has expand its usage.It acts as a channel among users and service providers.With the progressing of this world,web application has become more complex.Hence ,nowdays web applications are facing serious challenges. Content Security Policy is the solution of content injection attacks.It helps to identify and mitigate these attacks.Hence,many websites use this policy to secure themselves from attack. The main objective of our paper is to analyze the problems faced by websites caused by code injection attacks.This paper describes the advantages and limitations of existing techniques.This paper also describes comparison between existing techniques. We believe that analysis of web application attacks done by us will help to secure the content of websites in future.**

***Keywords*: : Web applications, Security, web attacks.**

I. INTRODUCTION

Web application provides a wide source of shared files and data to the users.Hence they support platform to the users.It provides services to the user on time.Web application provides various facilities to user such as online shopping,bill payment,registration forms,shared files,etc.as the world is progressing,web applications no longer remains simpler and has grown more complex nowadays.As a result web applications are facing problems of code injection attacks.HSBC,Google Search Engine,Facebook,Myspace,Twitter and Vodafone are some sites which have been reported to have content injection attacks.These attacks caused by the use of client side langauges such as javascript,vb script,activex,etc.Attackers exploit websites to perform cookie stealing,session hijecking,malware-spreading.Code injection attacks include XSS,CSRF,SQL INJECTION etc.Out of these attacks,XSS is the most popular and dangerous attack found in the websites.

Content Security Policy is the solution of code injection attacks.It detects and prevents XSS and other related attacks. Mozilla Foundation has developed this policy and it was first implemented in Firefox4.It makes the use of http header. It provides a robust mechanism for preventing the injection of malicious content.It is a declarative policy and computer security concept.

CSP can be used to enforce the following rules:

1.No untrusted sources

2.External resources can only be loaded from a set of clearly defined trusted sources.

3.No inline resources

4.Inline JavaScript and CSS will not be evaluated.

5.The JavaScript eval function cannot be used.

The main objective of CSP is to detect and prevent content injection attacks.Hence CSP is required to security to the websites .Many websites uses this policy to secure themselves from attackers.
Major attacks faced by websites are:

1. CROSS SITE SCRIPTING: It is a hacking technique used by attacker of injecting malicious javascript to the output of the web application.It is difficult to detect and prevent.
XSS is extraordinary then any other attack as it involves three parties:attacker,client and the targeted website.It is a client side attack.

In XSS attack,the attacker injects a website by his malicious javascript code.when the user visits this webpage,the script gets downloaded and executed.Hence all the confidential information of the user gets transferred to the attacker.

Types of XSS attacks are:

(1) PERSISTENT XSS: It stores the malicious code in a resource such as database which is managed by server and later displayed to the useris also known as reflected XSS.This type of attack is most commonly used by attackers.

(2) NON PERSISTENT XSS: It is also known as reflected XSS.This type of attack is most commonly used by attackers.In this type of attack ,the malicious code is not persistently stored.hence,it is immediately reflected to the user.

(3) DOM-BASED XSS: This attack operates when DOM-based Environment is changed or updated in the client side.

2. CROSS SITE REQUEST FORGERY: Also known as the 'sleeping giant'because many websites fail to protect against them and generally are ignored by service providers and security communities.Cross Site Request Forgery is also known as session riding,XSRF and Confused Deputy Attacks.In CSRF,the user is forced to execute the unwanted action on the trusted server application in which he has currently authenticated. Both csrf and xss are the cross site vulnerabilities caused by javascript.

3. SQL INJECTION: SQL Injection is the code injection attack in which an attacker injects malicious SQL query to a database which results destruction,modification in the database. It is a server side attack.

This paper analysis the code injection attacks which are described in introduction. This paper is divided into two sections.Section1 describes existing techniques which mitigates the XSS attack.Section 2 describes the comparison between existing technique..Rest of the paper describes the limitation,conclusion,future work of CSP and references.

SECTION: 1

Existing Techniques

NOXES: Noxes is a client side solution.It acts as a web proxy and performs on the basis of automatic and manually generated rules.

WORKING OF NOXES

The major objective of Noxes is to prevent the transfer of user data to the attacker.It helps to identify and block malware.Noxes acts as a application level firewall which gives user a full control of each and every connection entering and leaving out the local machine.If there is any mismatch ,it prompts the user either to allow or block out the connection.

PERFORMANCE OF NOXES

The table given below presents the statistical information about the analysed websites.Out of 6,460,952 total links,724,438 points to external domain.173,917 of these external link were requested by the browser.

Statistical information about the analyzed web pages	
Number of links	6,460,952
Number of external links	724,438
Number of requested internal links	698,483
Number of requested external links	173,917

We observed that there was no leakage of the information in the 55000 of the visited pages.Hence,anyone can visit these links freely.One can conclude that noxes gives better performance than other server side solution.

SWAP:SWAP is Secure Web Application Proxy.It acts as a reverse proxy.It is a server side solution to mitigate xss attack.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (299-302), Month: January - March 2015, Available at: www.researchpublish.com

WORKING OF SWAP

1. Encoding of all javascript calls to script ID.
2. Loading of each web page in browser and tracking of scripts that are trying to get executed.If any malicious code is identified then it is not encoded.
3. Decoding of all the scripts to regain the original form.

PERFORMANCE OF SWAP

Size(kb)	w/o SWAP	w/SWAP	SWAP	Factor
1	27.31	196.11	168.80	7.18
10	53.84	200.50	146.66	3.72
50	120.50	331.80	211.30	2.75
100	166.23	427.66	261.43	2.57

An experiment has conducted to measure the magnitude of performance of SWAP.The size of test pages on which test has been conducted given in the table above and are1kb,10kb,50kb,100kb.

Swap introduces performance delay as the comparison is done with swap and without swap.The performance of swap decreases with increase in the size of file.We observed from above table that a swap protected server will face delay in sending response to the client as compared to without swap protected server.

Delay in sending response is caused by two reasons:

1. All the traffic is retransmitted instead of linear transmission as a reverse proxy is placed between the client and server.As a result ,there is delay in sending the data to its destination.
2. Each page has to pass from the javascript testor component before it passes to the client.

SECTION: 2

COMPARISON BETWEEN EXISTING TECHNIQUES

SWAP is the server side XSS mitigation technique which operates on a reverse proxy.whereas Noxes is a client side solution which acts as an application level firewall.SWAP is dependent on the service providers to mitigate XSS attacks whereas Noxes is the first implemented client side solution which provides security to websites without depending upon the service providers.SWAP introduces delay in performance and is limited to javascript to prevent attacks whereas Noxes performs better than SWAP as it does not face performance overhead.SWAP does not seems better for high performance whereas Noxes suits better for high performance .Finally,we conclude that Noxes is better than SWAP.

Sno	Parameters	SWAP	Noxes
1.	Type of solution	Server side solution	Client side solution
2.	Dependency on service providers	Depends upon the service providers to mitigate XSS attack.	Does not depends upon the service providers to mitigate XSS attack.
3.	Limitation	SWAP introduces performance delay and is limited to javascript.	Noxes does not provide procedure to identify errors.
4.	Performance	SWAP is not suitable for high performance.	Noxes performs better than SWAP.

II. LIMITATION OF CSP

CSP is facing challenges in its implementation.It has been not implemented in websites to its full extent.CSP is lagging behind than other policies.challenges faced by CSP includes:first not to delete inline scripts prefered by websites that is removing inline scripts leads more time for the users to load the page.secondly,adopting CSP leads of breaking functionalities that is as high as a website is secure,it will be less involved in business activities.Hence it can harm business more than XSS attack. CSP is unable to identify the other attacks such as CSRF.CSP does not provide the order to which content is included.It only describes which data can be included.

III. CONCLUSION AND FUTURE WORK

CSP is not implemented to its full extent in the websites.Some websites adopted this policy has not get any profit.We conclude that no technique is full efficient as it suffers some percent of limitations.As high as a website is secure ,it will face losses in business.Removing of inline scripts lead to take more time to load the page.We observe that program and advertisements should be introduced and implemented to encourage the users to adopt this policy.So that more websites adopt CSP rather adopting by individual websites.According to our opinion,users must be alert and seriously concern about website attack.Rather on depending upon the service providers,users must take necessary action to secure websites.We suggest that certain improvements should be done to make CSP usable by websites.We hope that the analysis ofsecurity policy of content done by us will help to secure websites in future.

ACKNOWLEDGEMENT

We are happy to express our thanks to our teachers for their guidance. We are also like to thanks to our colleagues and all those people who have helped directly or indirectly to complete this review paper.

REFERENCES

[1] Engin Kirda, Christopher Kruegel, Giovanni Vigna, and Nenad Jovanovic. Noxes: A client-side solution for mitigating cross site scripting attacks. In Proceedings of the 21st ACM Symposium on Applied Computing (SAC), 2006.

[2] Lwin Khin Shar and Hee Beng Kuan Tan, "Defending against Cross-Site Scripting Attacks", IEEE Computer, Vol. 45(3), pp 55-62, March-2012

[3] “Reining in the Web with Content Security Policy” by Sid Stamm, Brandon Sterne and Gervase Markham, WWW 2010.

[4] “An Introduction to Content Security Policy byMikeWesthttp:// www.html5rocks. com/en/tutorials/ security/ content-security-policy

[5] CSP 1.0 W3C Candidate Recommendation:http://www.w3.org/TR/CSP/

[6] CSP 1.1 W3C Editor’s Draft http://w3c.github.io/webappsec/specs/contentsecuritypolicy/csp-specifi cation. dev. html .