Cross Site Scripting Attack and Broken Authentication and Session Management Attacks

Charu[1], Deepika[2]
[1]PG Scholar [M.Tech], [2]Assistant Professor, IITB Jhundpur, DCRUST Murthal,India

Abstract: ***Cross Site Scripting Attack (XSS) and a broken authentication and Session Management attack is perhaps one of the most common application layer attack technique used by attacker to deface the website. Broken Authentication and Session Management attack is ranked 2nd in the Open Web Application Security Project and Cross Site Scripting Attack (XSS) is ranked 3rd in the Open Web Application Security Project (OWASP)[1] top 10 vulnerability list. In this paper, we present a detailed review on various types of Cross Site Scripting Attack, and Broken Authentication and Session Management Attacks. Also explain Detection and Prevention of Cross Site Scripting Attack and Broken Authentication and Session Management Attacks.***

Index Terms: ***Attacks, Cross Site Scripting Attack (XSS) and Broken Authentication and Session Management attack, Detection, Prevention.***

I. INTRODUCTION

In XSS Attack some malicious script is injected in to the application through the input sources through which user session can be hijacked, user can be redirected to malicious website or link. The script contains the HTML or JavaScript codes which executes in to the user browser.

In Broken Authentication and Session Management Attack, Web application does not ensure for the unbreakable, unrepeatable authentication and authorization mechanism. Due to this an attacker can personify himself as a website user and can cause harm to the user and service providers. According to Cenzic 80% of web application were found vulnerable to session management in 2012[2].Broken Authentication and Session Management includes all aspects of handling user sessions and authentication mechanism [3].

II. CROSS SITE SCRIPTING

Cross site scripting (XSS) is a common vulnerability in web application Programs. XSS occurs when HTML or JavaScript code is injected into the database through input sources. If these inputs are not filtered from server side, then severe consequences may occur such as accessing and transmitting cookies, Redirecting to third party websites. The victims of the XSS are the most common sources such as comments, feedback, and search engines. They target authenticated users and system administrators

When an attacker gets a user's browser to execute script, the script will run within the security context of the hosting web site. Successful XSS attack allows attackers to hijack user's account via cookie, redirecting user to another website from the website visited and there by facilitating other types of attacks such as phishing or drive-by-download attacks. XSS attack poses significant risk in cases where the browser interacts closely with file system on the user's computers for loading content. XSS attacks are commonly used to hijack sessions of users visiting websites facilitating e-Commerce, wherein malicious Script/code runs on user's client and captures cookie

Corresponding Author: [1]

1. ***Ms. Charu, PG Scholar,*** *IITB Jhundpur,DCRUST Murthal, India*
   Email Id: *charukhurana4@gmail.com*
2. ***Ms. Deepika, Assistant Professor,*** *IITB Jhundpur,DCRUST Murthal, India.*
   Email Id: *dhulldeepika11@gmail.com*

Information of user's browser allowing hijacking of session. There are different types of XSS attacks, primarily two types called "There are different types of XSS attacks, primarily two types called "Stored or Persistent Attack and "Reflected or Non-Persistent Attack.

A. Stored or Persistent Attack

In stored attacks the malicious code injected by the attacker is stored permanently in the database. The code is accepted as a part of input field. The victim then receives the malicious code when it requests the stored information.

B. Reflected or Non-Persistent Attack

Reflected attack occurs off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. They are also delivered to victims via another route, such as in an e-mail message, which reflects the user to another link or to an un-trusted source or server. Occurs when the server does not properly sanitize the output server to a visiting web browser/client.

C. DOM Attack

DOM-based is unique form of XSS, used very similarly to non-persistent, but where the JavaScript malware payload doesn't need to be sent or echoed by the Web site to exploit a user.

D. Impact of Cross-site Scripting Attack (XSS)

The impact of Cross-site scripting attack varies depending up on the level of access and type of information gained by the attacker. The consequences of successful XSS attack range from annoyance to compromise of user's credentials.

Typically XSS attacks are used as part of larger scheme of attacks such as redirection of visitors of a trusted website to malicious websites, malware propagation, e-Commerce frauds etc.

III. BROKEN AUTHENTICATION AND SESSION MANAGEMENT

Broken authentication and session management attack occurs whenever there is session hijacking or false authentication occurs. It includes management of handling all aspects of authentication and sessions which can affect the web servers, application servers, web application environment and can cause misuse of privileges. For e.g. the attacker can change message or can misuse information retrieved. Various cryptographic algorithms and session management tokens are used by developers for this kind of attacks but still it is a major issue. For handling authentication and session issues various issues should be kept in mind such as:

Password Strength

Passwords should have minimum length and appropriate use of alphabets, numbers and special keywords to avoid guessing.

Password Use

Number of login attempts should be restricted as authenticated users will never re attempts if he forgets the password. After attempted maximum attempt user should be restricted to avoid attack.

Password

A single password change mechanism should be there to avoid the security flaws. The mechanism should ask for old and new password but changing email address or changing phone number should be avoided.

Password Storage

Password should be stored in with encrypted or hashed method to protect them from exposure.

Session Id Protection

Session IDs should be long, complicated, random numbers that cannot be easily guessed. It should be changed frequently during a session to reduce how long a session ID is valid. Session IDs must be changed when switching to SSL, authenticating, or other major transitions. Session IDs chosen by a user should never be accepted.

IV. CROSS SITE SCRIPTING (XSS) DETECTION AND PREVENTION

"BIXAN: Browser Independent XSS Sanitizer for Prevention of XSS attacks "(Sharath Chandra v, S.Selvekumar) (2011)

[4] A technique is proposed which is invoked when user injects code in the field of web application. The HTML content is passed to the XSS sanitizer. Sanitizer parses the HTML content and checks the presence of static tags. The static tags are retained while rests of tags are filtered out. Even static tags contain dynamic content which are filtered out by JavaScript tester. After filtering HTML content is converted into DOM. The sanitized user content is retained in DOM. This approach was tested by cheat sheet [5] contained 114 scripts and result is obtained in various web browser. It was found that all 114 scripts were filtered out but BIXAN .Moreover it reduces the anomalous behavior of web browsers. The weakness of this solution is that it is at the server side and browser source is modified for obtaining results.

"Protecting Cookies from Cross Site Script Attacks Using Dynamic Cookies Rewriting Technique", (Rattipong Putthacharoen , Pratheep Bunyatnoprat) (2011)

[4] A technique is proposed which is implemented as a Web Proxy. As a Web Proxy Server it will automatically rewrite the cookies that are sent back and forth between the web applications. With this technique the cookies at the browser database are not valid for web application; the Web Proxy will automatically rewrite the cookie with the randomized value before sending cookie to the browser. The returned cookie from the browser is rewritten back by the web proxy before forwarding it to the web server. The Web Proxy is placed between the client and the web server. There are four domains which are kept to identify the cookie which is Name, Domain, Path, and Port. The original value of the cookie and the randomized value is also kept in the same table. This table is stored at the web proxy database server. The web proxy server will use this information to rewrite back cookies. The drawback of this approach is the compatibility issues which occur while implementing the proxy server and the single point failure issue.

"S2XS2: A Server Side Approach to Automatically Detect XSS Attacks" (Hossain Shahriar and Mohammad Zulkernine) (2011)

[6] A detection framework is proposed which has six modules. The first module injects boundary for content generation location .They apply two types of boundaries HTML Comment and JavaScript comment. Comments are injected with token to identify duplicity. The policy is generated for attack detection. The second module is policy storage which stores policy for attack detection. The third module is web server which represents web program container where injected boundaries programs can be accessed. Web server generates response pages and forwards it to next module. The fourth module is feature comparator which compares response page with policy rules. The fifth module is attack detector which detects and removes malicious code and forwards response to next module. The boundary remover is last module which removes boundary injected during first module from safe content. This approach was tested in java and results were obtained with zero false negative. It also detects advanced XSS. The weaknesses of this approach are it suffers from false positive and response delays. The policy checking phase is also exhaustive.

V. BROKEN AUTHENTICATION AND SESSION MANAGEMENT DETECTION AND PREVENTION

"Automatic Detection of Session Fixation Vulnerabilities in Web Applications"(Yusuki Takamastu, Yuji Kosuga, Kenji Kono) (2012).

[7] A technique is proposed to check session fixation attack. Session fixation is an attack in which attacker forces visitor to use session ID passed by him and can use application as a visitor. In this approach they designed a system which works in three steps. First step is Packet Capturing captures all the packets observe the change of sessionIDs. The system lies between the user browser and the web application server. In second step Initial Inspection is done for checking the vulnerability for session ID. The third step is Attack Simulation in which system launches attack simulator. The attack simulator automatically generates the same environment as a real attacker performs as a virtual attacker and a victim. At this time virtual attacker access and login to web application with Session ID (SID) that attacker obtained. The virtual attacker checks whether he can login with the obtained SID or not. The response is checked for keywords. For example "welcome victim" if such type of keyword is obtained then web application is considered as vulnerable. The weakness of this approach is that it is difficult to manage such kind of virtual environment for real world application.

The primary recommendation for an organization is to make available to developers:

".A single set of strong authentication and session management controls" to prevent Broken Authentication and Session Management .

Such controls should strive to meet all the authentication and session management requirements defined in OWASP's Application Security Verification Standard (ASVS) areas V2 (Authentication) and V3 (Session Management). Such Controls have a simple interface for developers. ESAPI Authenticator and User APIs are good examples to emulate, use, or build upon. Strong efforts should also be made to avoid XSS flaws which can be used to steal session IDs.

VI. CONCLUSION

In this paper we present a detail review on Cross-site Scripting (XSS) Attack and Broken Authentication and Session Management Attacks. Also describe detection and prevention of Cross-site Scripting (XSS) Attack and Broken Authentication and Session Management Attacks.

REFERENCES

[1]. https://www.owasp.org/index.php/Category:OWAS_Top_Ten_Project.(Accesed in Jan 2013)

[2]. Cenzic vulnerability report 2013 http://info http://info.cenzic.com/rs/cenzic/images/Cenzic Application-Vulnerability-Trends-Report-2013.pdf.

[3]. https://www.owasp.org/index.php/Top_10_2013-A2

[4]. Sharath Chandra V.,S.Selvekumar , "BIXAN:Browser Independent XSS Sanitizer For Prevention Of XSS Attacks.ACM SIGSOFT ,September 2011 Volume 36 Number 5.

[5]. XSS prevention cheat sheet OWASP https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet.

[6]. Hossain Shaihriar And Mahammad Zulkernine , "S2XS2 :A Server Side Approach To Automatically Detect XSS Attacks , 2011 IEEE Ninth International Conference On Dependable ,Automatic Secure Computing , PP.7-17

[7]. Yusuki Takamastu, Yuji Kosuga, Kenji Kono "Automatic Detection Of Session Fixation Vulnerabilities "2012 Tenth Annual International Conference on Privacy, Security and Trust IEEE, PP-112-119.