A Survey on Website Attacks Detection And Prevention

[1] Bipin A Raval, [2] Dr.Priyanka Sharma
[1] Student M.Tech(Cyber Security),[2]Professor (IT)
[1,2]Department of Information technology
[1,2]Raksha Shakti University, Ahmedabad, India.

ABSTRACT

In Web Security Applications Has Become Important As The Information Processed By Web Applications Has Become Censorious Corporations, Customers, Organizations, And Country. In Web Attacks Manage A Border Array Of Information Which Includes Financial Data, Medical Records, Social Security Numbers, Intellectual Property And National Security Data. There Many Attacks Through Which Web Based Attacks Are Believe By Security Experts To Be The Greatest And Less Understood Of All Risks All Related To Confidentiality, Availability, And Integrity. This Paper Will Introduced Website Attack And Also How To Detect And Prevent From The Attacks. Attacks Covered Are SQL Injection (SQLI), Cross Site Scripting(CSS), Cross Site Request Forgery(CSRF), Broken Access Control.

Keywords*: Website attacks, Security, Sql injection, XSS, CSRF, Vulnerability detection and Web applications*

I. INTRODUCTION:

In the World Wide Web is most of common communication methods in the world. Thousands of users are connecting to every day and every time to different web-based applications to search an information, message are being exchange, deal with each other, its connect with the business, financial operations and more things. There are some critical web-based services are targeted to of this several malicious clients intending to exploit possible vulnerabilities which could cause not only the disruption of the service, but many compromise the users and organization information. Many time, there are malicious user's success in exploiting different types of vulnerabilities.

II. ATTACKS ON WEBSITES:

1. SQL INJECTION:

SQL Injection Attacks (SQLIAs)-Structured Query Language (SQL) is an illuminate language used in database drive web applications which construct SQL statements that incorporate user supply data or text. If this process is done in an unsafe manner, then the web application in the website may be vulnerable to SQL Injection Attack i.e. If user give data is not properly validated, then user can modify data or craft a malicious SQL statements and can execute arbitrary code on the target Machine or modify the contents of database [2]. SQL Injection Attack is targeted on a process at the database layer which is connected to a web database application. This SQL Injection Attack exploits weakness or vulnerability in the target program to Properly verify the input supplied to it through a web form.

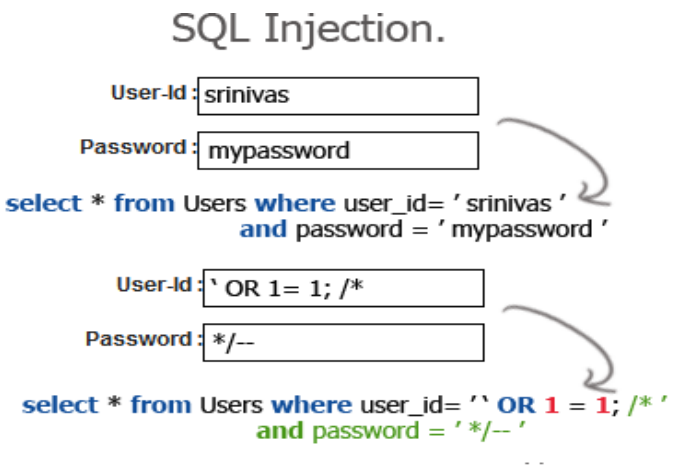

FIG 1.SQL INJECTION

Impact of SQL Injection

1) Confidentiality: Confidentiality is a main problem with SQL Injection attacks, SQL databases give the sensitive data and critical information which could be viewed by unauthorized users as a consequence of successful SQL injection attack [3].

2) Integrity: Successful SQL injection attack allows external source to make unauthorized modifications such as altering or even deleting information from target databases [3].

3) Authentication: Poorly written SQL queries do not properly validate user names and passwords, which allows unauthenticated entity or attacker to connect to the affected database or application as an authenticated user, without initial knowledge of the password or even user name.

4) Authorization: Successful exploitation of SQL injection vulnerability, allows attacker to change authorization information and gain elevated privileges if the authorization information is stored in the affected database

2. CROSS SITE SCRIPTING (XSS)

In a typical XSS attack the hacker infects a legitimate web page with his malicious client-side script. When a user visits this web page the script is downloaded to his browser and executed [8]. There are many slight variations to this theme, however all XSS attacks follow this pattern, which is depicted in the diagram below.

FIG .2 XSS ATTACK

Impact of Cross-site Scripting Attack(XSS)

(1) With the help of XXS attack attacker easy access the account of user by stealing cookie session and also impersonate user and access sensitive information.
(2) XSS attacker stealing victim credential with the help of this attack cloning the page of website and then using the XSS vulnerability in order to serve it to the victims.
(3) XSS attacker easy access the sensitive information of victim using siphoning fraud such as card holder data.

3. CSRF ATTACK

Cross-site request forgery is a browser based attack where an attacker sends requests to the targeted victim website and tries to perform unauthorized actions. CSRF is an attack which forces an end user to execute unwanted actions on a web site, in which he/she is currently authorized. With the help of HTTP protocols functionality to send session cookie for each request to server, which helps server to confirm that the request is coming from authenticated user. CSRF attacker first study the request pattern i.e. types of request, parameters names, type of parameters values etc. After studied the request's URL pattern deeply, he embeds this URL in html tags of web pages or emails. Then attacker forces the authenticated user to execute this request. As user is authenticated browser automatically sends session cookie value with this request, server accepts this request and execute it.

FIG. 3 CSRF ATTACK

Impact of CSRF attack

(1) With the help of this attack attacker easy transfer money from one account to another.
(2) Attacker easy change the content of the website.
(3) Attacker easy change the password of user.

4. Broken Access Control

In web application the users are categorized in the different level of privileges. Access control determines how the web application allows access to functions to some users and not others, also called authorization. But attacker may be access higher level of authority.

Impact of broken access control

1.Access to confidential or restricted and sensitive information from website.
2.Execute unauthorized operation on website.
3.Tampering data of website.

III. Detection and prevention of website attack

SQL INJECTION

1. "An Authentication Scheme using Hybrid Encryption "(Indrani Balasundaram, E.Ramaraj)-(2011)[4]. Indrani

Balasundram and E.Ramaraj proposed an authentication scheme in which uses two algoritham .the first one AES(advance encryption standard) and second RSA(rivest-samir-adleman) to prevent from the sql injection . In this authentication scheme a unique secret key is assigned or fixed for every user. On the server side for RSA encryption server uses private key and public key combination. In this authentication scheme, on login query two level of encryption is applied.

symmetric key encryption is used to encrypt user name and password with the help of user's secret key.
To encrypt the query, the scheme uses asymmetric key encryption by using public key of the server.

The whole authentication scheme completed in three phases:
1.Registration Phase: this phase used for registration the user.
2.Login Phase: this phase used for the login the user.
3.Verification Phase: this phase verifies the user name and password for authentication.
The proposed scheme is very efficient, it needs 961.88ms for encryption or decryption and this can be negligible.

Some disadvantages exist with this approach:
1.. At registration phase there is no security mechanism.
2. In this approach client side and server side user secret key is Very difficult to maintain and complex.
3. This approach not prevent Url based SQL injection attacks

2."Automated generation of prepared statement to remove SQL injection vulnerabilities"(Stephen Thomas, Laurie Williams, Tao Xie) (2008) Stephen Thomas provide an algorithm for removing SQL vulnerabilities. In this algorithm prepared statement in SQL queries are replaced by secure prepare statements for removing SQL vulnerabilities. And Prepared statements have a static structure so prevents SQL injection attacks from changing the logical structure of a prepared statement [8]. In this approach they created a algorithm which replaces prepared statement and a corresponding tool for automated fix generation. To evaluate the capability of the algorithm and its automation conducted four case studies of open source projects. When some experiment on this approach it's result is provide 94% accuracy to remove sql vulnerable statement.

3."Combinatorial Method for Preventing SQL Injection Attacks" (R. Ezumalai, G. Aghila) 2009. It is a signature based SQL injection detection technique [6]. This method to detect SQL injection uses both static and dynamic approach. In this method in web site code they generate hotspots for SQL queries and after divide these hotspots into tokens and for validation tokens send it to the Hirschberg's algorithm, which is a divide and conquer version of the Needleman-Wunsch algorithm, used to detect SQL injection attacks. Since, it is defined at the application level, requires no change in the runtime system, and imposes a low execution overhead.

XSS (CROSS SIDE SCRIPTING)

1."An Execution-flow Based Method for Detected Cross-Site Scripting Attacks"(Qianjie Zhang, Hao Chen, Jianhua Sun)(2010)[7] Qianjie Zhang, Hao Chen, Jianhua Sun presents an execution-flow analysis for JavaScript programs running in a web browser to prevent XSS attacks[9]. In this approach to model the client-side behavior of Asynchronous JavaScript and XML or AJAX applications under normal execution they use Finite-State Automata (FSA). In this method system is establish in proxy mode. In this mode the proxy analyzes the execution flow of client-side JavaScript and checks them to be with rules and regulation to the models generated by FSA. Before the requested web pages arrive at the browser It stops high risk malicious scripts, which do not conform to the FSA. This method is evaluated against many real-world web applications and the result shows that it protects against a different of high risk malicious scripts to prevent XSS attacks and has an acceptable performance overhead.

2."Automatic Creation of SQL Injection and Cross-site Scripting (XSS) Attacks(Ardilla)" (Adam Kie_zun, Philip J. Guo, Karthick Jayaraman, Michael D.Ernst)(2010) Adam Kie_zun has suggested a technique for finding vulnerabilities in Web Application such as SQL injection attack and Cross site scripting (XSS)[9].This technique use an automated tool called Ardilla. To find vulnerabilities in web application this method uses static code analysis. Before the application is deployed this technique works source code of the application, creates concrete inputs that expose vulnerabilities and operates. And to discover vulnerabilities in the code it analyses application internals. It is based on input generation, taint propagation, and input mutation to find variants of an execution that exploit

vulnerability. Ardilla tool is designed for php applications.

Some disadvantages also with this approach are:
1.Developer availability and learning is required.
2.Adjustment of the source code is needed in this approach.
3. It is very difficult and complex to patch the vulnerabilities If the main developer skips the project.
4.This is tested on PHP based applications.

CSRF (CROSS SITE REQUEST FORGERY)

1.Checking Referee Header:
HTTP request contain different parameters such as date and time format, Http version, character sets, content encoding and url, one of these parameters contain the URL of site from which request originates, that parameter name is 'Referee'. Before request forwarding to server This parameter can be used by browser to check requests domain on client side. So that web application developers check Referee header to protect web applications from CSRF. This can be applied in case of critical operation like change user password, a transfer an amount, change user privileges and purchasing items etc. This will allow only same domain request to execute.

2.Anti CSRF

To protect website from CSRF attacks a library developed in C# for ASP.NET developers. To protect web application against CSRF attack HTTP module which can be added to web application. This module itself generate token and checking it on every page of web site, assuming it inherits from System. Web Page and contains ASP.NET form. Library need to be added as a reference to web site and related settings has to be done in web configuration file. For adding CSRF token to the ASP.NET application is to use View State in combination with View State User Key. Because of session id will be used as a unique key to identify user the user requires View State to be enabled and as well as session to be enabled. Anti CSRF module works without this requirement and hence provide more independent environment. Anti CSRF requires Cookies to be enabled on Users browser and when browser will get closed cookies used on browser cleared. It uses hidden field to carry out CSRF token.

3.CSRF detector

CSRF detector detects CSRF attacks with the notion of visibility and content checking of suspected requests. The idea is to stop and catch a suspected request containing values, parameters and related them with one of the visible forms present in an open window. If suspected request is an exact match, then suspected request is changed to make it benign, then it is launched to the remote website to identify the content type, this content type is then matched with the expected content type. IF there any mismatch between request attribute values or content type results in a warning. Moreover, at a later stage for CSRF attack detection it does not require storing URLs or tokens to be matched. This can be implemented as a Firefox plug-in. If it is cross site request Once it detected the CSRF attack, we can stop that request and also blacklist that particular site. Hence this detector will be useful to prevent CSRF attack.

BROKEN ACCESS CONTROL

Detection
(1) For the detection of broken access control use access control policy should be clearly documented. If their problem in access control document, then your site is vulnerable to access authenticated user.
(2) The code that implements the access control policy should be check and such code should be well structure modular and most likely centralized [9].
(3) You can protect communication Chanel which control under administration
(4) In the web application, if there are categories of users that can be accessed through the interface, verify each interface to make sure that only authorized users can have allowed access

Prevention:
(1) For the prevention broken access control use access control matrix to define the access control rules. With the help of this access control matrix the policy should documented what type of user access the system, and what function and content each of these types of users should be allowed to access so the attacker cannot bypass it.
(2) Attackers will use path traversal method so don't store the sensitive configuration file on web root.
(3) Ensure that the first n characters of the fully qualified path to the requested file is exactly same as the "document root"

IV. CONCLUSION:

Web applications have been growing extremely fast with innovative programming languages and technologies. This results in challenges for web application security, which requires extensive and continuous efforts from security researchers. This paper provides survey on few web site attacks such as SQL Injection (SQLI), Cross Site Scripting(CSS), Cross Site Request Forgery(CSRF), Broken Access Control with detection and prevention schemes. But with various complexities such as an increasing amount of application code and logic, multiple web applications are integrated and embedding third-party programs with security it is not possible to provide clean solution to all these attacks.

V. REFERENCES:

1. A Survey On Web Application Vulnerabilities(SQLIA,XSS)Exploitation and Security Engine for SQL Injection.(https://www.researchgate.net/profile/Rahul_Johari2/publication/254034977_A_Survey_on_Web_Application_Vulnerabilities_SQLIA_XSS_Exploitation_and_Security_Engine_for_SQL_Injection/links/552f70e40cf22d437170e1cb.pdf)

2. Web Vulnerability Detection and Security Mechanism. (http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.3290&rep=rep1&type=pdf)
3. CSRF Vulnerabilities and Defensive Techniques (http://www.mecs-press.org/ijcnis/ijcnis-v4-n1/IJCNIS-V4-N1-4.pdf)
4. A Survey on Website Attacks Detection And Prevention (http://ijaegt.com/wp-content/uploads/2014/12/409272-pp-238-244-satish-.pdf)

5. Indrani Balasundaram, E.Ramaraj "An Authentication Scheme for Preventing SQL Injection Attack Using Hybrid Encryption(PSQLIA-HBE"(ISSN 1450-216X Vol.53 No.3 (2011),pp.359-368)

6. Rattipong Putthacharoen, Pratheep Bunyatnoparat " Protecting Cookies from Cross Site Script Attacks Using Dynamic Cookies Rewriting Technique" Feb. 13~16, 2011 ICACT2011. Method for Detecting Cross-Site Scripting Attacks".

7. Rahul Johari, Pankaj Sharma, "A Survey On Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQL Injection" 2012 International Conference on Communication Systems and Network Technologies 978-0-7695-4692-6/12, IEEE DOI 10.1109/CSNT.2012.104, 2012.
8. A Survey on Web Application Vulnerabilities (SQLIA, XSS)Exploitation and Security Engine for SQLInjectionCommunication Systems and Network Technologies (CSNT), 2012 International Conference on Date of Conference: 11-13 May 2012 Author (s):Johari, R.;Sharma, P. USIT, GGSIP Univ., Delhi, India

9. OWASP. https://www.owasp.org/index.php/CSRF, Cross-Site Request Forgery, Testing for CSRF (OWASP-SM-005)

10. Katkar Anjali S., Kulkarni Raj B., "Web Vulnerability Detection and Security Mechanism", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-4, and September 2012.

11. " Automatic Creation of SQL Injection and Cross-Site Scripting Attacks "ARDILLA(Adam Kie_zun, Philip J. Guo, Karthick Jayaraman,Michael D. Ernst)

12. Tatiana Alexenko, Mark Jenne, Suman Deb Roy, Wenjun Zeng , "Cross-Site Request Forgery: Attack and Defense", IEEE CCNC 2010

13. Hossain Shahriar and Mohammad Zulkernine "Client-Side Detection of Cross-Site Request Forgery Attacks", 21st International Symposium on Software Reliability Engineering .