ISSN:0975-9646

Web Application Security: A Survey

Leena Jacob, Virginia Mary Nadar, Madhumita Chatterjee

Department of Computer Engineering and Information Technology ,PIIT, New Panvel, Mumbai University

***Abstract*—Nowadays, Internet is widely used by many organizations. So the rise of web applications on the internet is also increasing which eventually give rise to various kinds of attacks on these web applications. In this paper we present few of the attacks such as SQL Injection, Cross-Site Scripting(XSS) , Insecure Direct Object References(IDOR) , Sensitive Data Exposure and Using Components with Known Vulnerability. Some of the existing detection and prevention techniques is analyzed along with a comparative study based on certain parameters.**

Keywords—Vulnerability, SQL Injection, Cross-site Scripting,Sensitive Data Exposure,Insecure Direct Object References,Using Components With Known Vulnerability.

I. INTRODUCTION

Web Applications can be defined as a program that is installed and running in remote web server which is delivered over the internet and responds to requests via HTTP protocol. Web developers create the web applications but they are not very well versed with the security concerns which thereby create a loophole for the vulnerability. A vulnerability can be defined as existing weakness in the system which allows the intruder to access the data or gain complete control of the application. In this paper we will be discussing about SQL Injection ,Cross-site Scripting, Sensitive Data Exposure, Insecure Direct Object References and Using Components with known vulnerability.

SQL Injection attack is a code injection attack which tricks the user in executing unintended commands or gaining access to data without proper authorization. When an application takes untrusted data and sends it to the browser without proper input validation or escaping then a Cross-site Scripting attack can occur which can hijack user's sessions, deface websites or re-direct users to malicious sites. Sensitive Data Exposure means that the sensitive data of user like credit card information, bank details etc is compromised .Insecure Direct Object References represents the lack of authentication level checks that results in incorrect level of administrative access to the system data. Using Components with known vulnerability represents those components in the system that has flaws which leads leakage of data or control of servers is lost.

The organization of the paper is as follows: Section I gives introduction about web application and need of security, ,Section II about SQL Injection attack ,Section III explains the Cross-Site Scripting attack, Section IV describes other web application attacks, Section V gives a comparative study of the listed attacks. The Section VI gives the conclusion of the study and Section VII includes the references.

II. SQL INJECTION ATTACK

In SQL Injection attack malicious code is injected into strings and then passed into database backend for parsing and execution. There are various types of SQL injection attack such as tautology, piggy-backed queries, logically incorrect queries, stored procedures, union query and Inference based attacks.

A. Research of SQL Injection Attack and Prevention technology

The author[1] has proposed a defensive model which is used to prevent SQL Injection attacks. The model provides input validation and also detects web address bar information for checking the sensitive character. In this model the user submits the form and data of web address bar. The server side checks the legitimacy of the IP address. If the input values are illegal then user is denied access to the login server.The testing format, length ,type and range of the input values are checked by the server side. If the input string is legal then the user is allowed to access the page. The server side verifies the privileges of the user and if the the user's number exceeds the access permissions then the user is timely blocked. The system then sends a message to the system administrator and the server records the injection attacks when all verifications are invalid. It detects the attacks by comparing the input size of the intended queries with the actual queries.This technique helps to protect the web application from tautology and logically incorrect queries.

B. An Approach to Detect and Prevent tautology Type SQL Injection in Web Service Based on Xschema Validation

The author[2] has proposed a technique that detects the tautology type of attack.This technique consists of three basic elements: SQL Scanner,XML File Maker and Xschema Validator.SQL Scanner intercepts the queries generated at run time before they are executed in the database server.The function of XML File maker is to analyze the intercepted query and collect inputs from that query.The inputs are then stored in XML document.For example-Suppose the SQL query for login mode is as follows:

*select * from user_info where U-Id ='leena' OR 1=1 - - 'password='';* .Now in this example two inputs that will be taken are leena and 1=1.These two input parameters will be taken and stored in the XML file. It is then passed to the Xschema Validator which will validate the input based on its rules.If the input parameters passes the validation it means that injection attack is not present .The parameters which does not pass the validation are called as

failures and is logged in a log file. This system has its uses in web application logging and security.

C. AMNESIA

AMNESIA stands for Analysis for Monitoring and Neutralizing SQL Injection attack. This method[3] can detect and prevent all types of SQLI attacks except for stored procedures.The working of this approach is a combination of static analysis and runtime monitoring. First it predicts the possible structure of all the legitimate queries that can be generated by web application's code. This technique builds a model of legitimate queries with the help of Java String Analysis(JSA)which computes the possible values for each hotspot's query string. Then an instrumented application is created that calls to the monitor which will check the queries generated at runtime before the call is made to the database.

In the runtime monitoring phase the query string is sent to the runtime monitor. Then it will parsed into sequence of tokens according to specific SQL tokens. The incoming query matches with structure present in the SQL-query model then the runtime monitor will execute that query. Otherwise the query is rejected and treated as malicious query. The disadvantage of this method was that it involved the usage of many different tools.

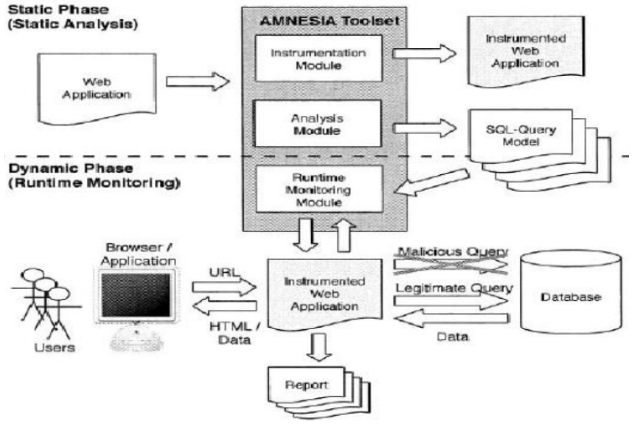

Fig.1. Overview of Amnesia[3]

D. CANDID

CANDID stands for Candidate evaluations for Discovering Intent Dynamically. This technique[4] can detect and prevent only tautology type of attack. The candidate inputs must satisfy two conditions ,firstly the inputs must be benign .Second ,the inputs must dictate the same path in a program. This mechanism computes the query supplied by the user by running the application on candidate inputs. These candidate inputs are considered to be self evidently non-attacking. The advantage of this method is it solves the issue of manually modifying the application to create prepared statements. The following figure shows the overview of Candid where the first part shows the off-line view and second part shows the run-time view.

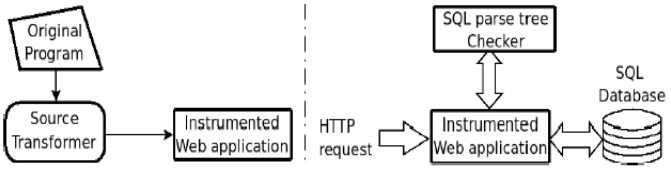

Fig.2. Overview of Candid[4]

III. CROSS-SITE SCRIPTING ATTACK

Cross-site Scripting attack is an attack or a vulnerability present in websites or web applications which allows the malicious users to insert their client side code into those web pages. When this code along with original web page gets displayed in the web client allows the attackers to gain access to that page. XSS attacks consists of three types such as Persistent XSS attack ,Non-Persistent attack which is also called as Reflected persistent XSS attack and DOM-Based XSS attack. The persistent XSS attack in which the malicious string originates from the website's database. The Non-Persistent XSS attack for which the malicious string originates from the victim's request. The DOM-Based XSS attack is also called as 'Type 0' attack ,here the vulnerability is in the client-side code rather than in the server-side code.

The existing XSS detection techniques includes the following:

- Detecting XSS vulnerability using HTML5[23]
- Dynamic Cookie Rewriting[24]
- Execution flow based method[25]

The XSS prevention methods consists of the following:

- Pattern Filtering Approach for Persistent XSS attack[22]
- Enhanced browser defense for Reflected Cross-site Scripting[5]
- Code Injection Detection Tool approach[27]
- Using the Zend framework[26]
- E-Guard Algorithm[28]

A. Classification of XSS Detection Techniques

The Cross-Site Scripting can be detected in two ways such as Static Analysis and Dynamic analysis.

Under static analysis we have the following techniques:

- Bounded Checking[12],
- Software Testing Approach[13],
- Taint Propagation[14] and
- Using Untrusted Scripts [15].

The various approaches under Dynamic analysis are :

- Browser Enforced Embedded Policies[16],
- Syntactical Approach[17],
- Proxy Based Approach and Interpreter Based Approach [18].

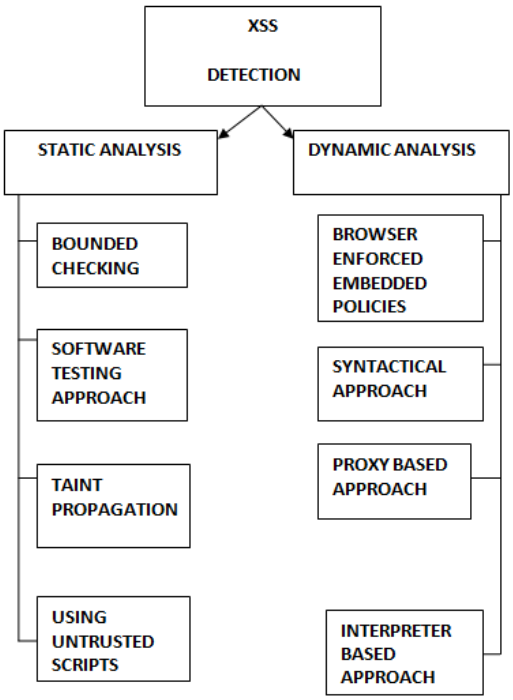

Fig.3. XSS Detection Techniques

B.Classification of XSS Prevention Techniques

Cross-site Scripting prevention can be either server side or client side. The following figure illustrates the techniques present under server and client side.

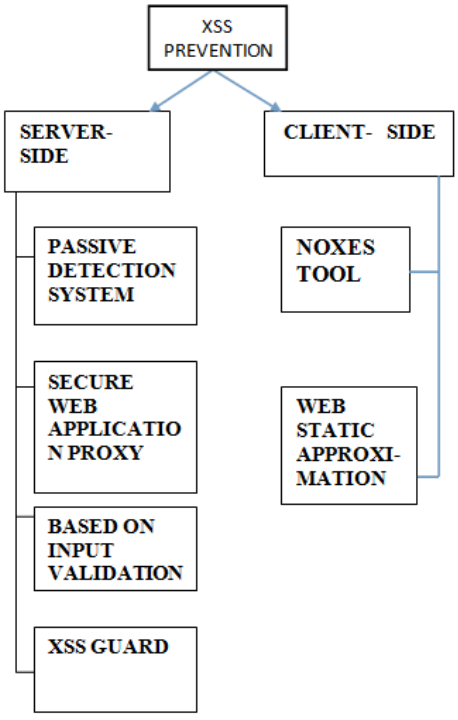

Fig.4. XSS Prevention Techniques

Under the server-side we have techniques like Passive detection system, Secure Web Approximation proxy[19], Based on Input Validation and XSS Guard[20] .

The client-side have techniques like Noxes tool[11],Web Static Approximation[21].

C.Enhanced Browser Defense for Reflected Cross-Site Scripting

The author[5] has proposed an approach that works only for reflected XSS attack. This paper implements a mechanism that is based on encoding unfiltered reflections for detecting vulnerable web applications. It provides accurate higher detection rate of exploits. XSS-Me is finest open source tool that is used to detect SQL and XSS vulnerabilities. XSS-Me is a Mozilla Firefox add-on that is present in the side bar and points to fields or forms of a webpage. It restates an array of XSS strings preferred by the user which includes opening new webpages and finally testing the results. The existing system locates forms that are hidden and visible and list them in tabs. Then heuristic testing is done which then marks the XSS vulnerable page as Failure else it Passes .It will also check for the Javascript special characters and if any vulnerable charater is found it leads to Warnings. This approach has made an enhancement to existing XSS-Me where the the input string for which the page is XSS vulnerable is encoded/escaped and then finally blocked .

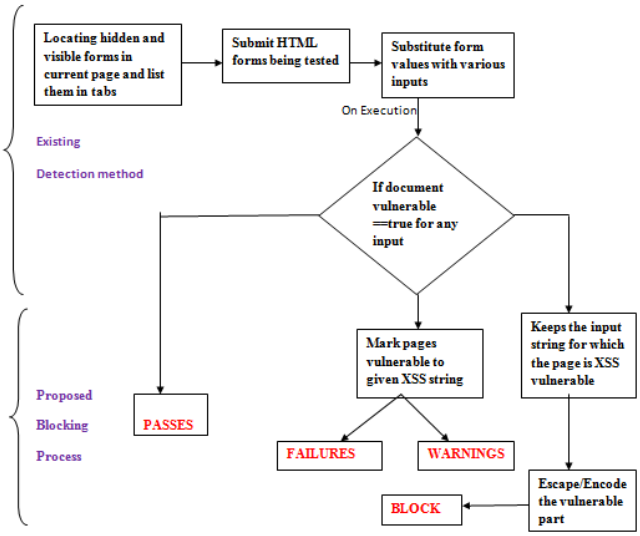

Fig.5. Proposed Defense Model[5]

This method provides a client side solution in Mozilla Firefox which achieves high fidelity and high performance.

IV .OTHER WEB APPLICATION ATTACKS

In this section we will consider attacks such as Sensitive Data Exposure, Insecure Direct Object References and Using Components with Known Vulnerability.

A. Sensitive Data Exposure

It is very essential to store the sensitive data such as credit card information, bank details etc otherwise the attackers can steal such sensitive data and get benefits out of it.We need to protect the sensitive data while storing,retrieving or sharing with the front end.This paper[7] proposes a security policy based on dynamic database to prevent sensitive data exposure using Oracle database.Here the

focus is on restricting the data extraction using the SELECT statement. The following figure shows the architecture of SDF which first builds SDF catalog and then generates custom SDF security policy function .

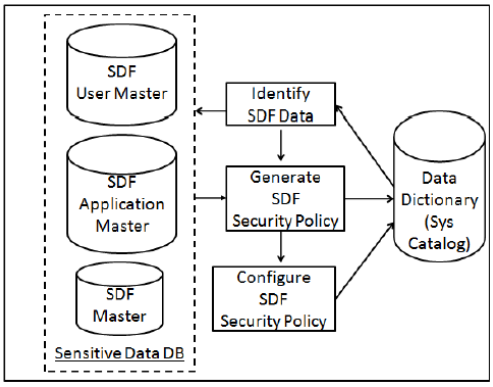

Fig.6. Architecture of Sensitive Data Filter [7]

The model consists of SDF(Sensitive Data Filter) which will execute the query if no security policy is imposed on the objects .This approach is highly dynamic and flexible thereby preventing sensitive data exposure. The advantages of this model are as follows:

- It is easy to implement.
- It does not require extra coding efforts as it works at the database level.
- It reduces human efforts due to endless configurations.

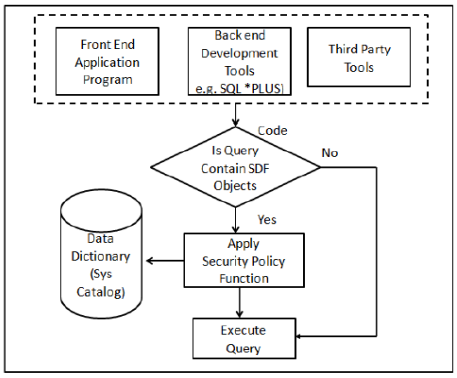

Fig.7. Work Flow of SDF[7]

B. Insecure Direct Object References

The IDOR[6] represents the flaws in the system design without the full protection mechanism for sensitive data or resources.Suppose an organization provides its users the financial data report via website with an assumption that no user will see other user's data. If the URL is (eg. /accounts/viewDetail?eid=0010).So here the eid becomes very predictable,any user can enter eid as 0020 and view the details of that particular eid.Here the report id (eid) is considered to be direct reference to the object.This paper shows the vulnerability of IDOR with the help of tools such as WebGoat and BURPSITE along with some preventive solutions such as Access Reference Map and GUID(Globally Unique Identifier).

C. Using Components with known Vulnerability

When a component with a known vulnerability exists in the software system ,it risks the system of data leakage or loosing conrol over the server increases. It may even affect the working of other components. The author[10] presents a Vulnerability Alert Service process that tracks the known vulnerabilities existing in software systems during their entire life cycle.

V. COMPARATIVE ANALYSIS

The following table compares some of the tools which identifies which all type of SQLI attacks it can detect and prevent.

Table.1. Comparison of SQLI Attacks

Attack	Tautology	Logically Incorrect Query	Union Query	Stored Procedure	Piggy-Backed Query	Inference
X Schema Validation[2]	Yes	No	No	No	No	No
Amnesia[3]	Yes	Yes	Yes	No	Yes	Yes
Candid[4]	Yes	No	No	No	No	No
Research on SQLI[1]	Yes	Yes	No	No	No	No

The following table compares attacks such as Sensitive Data Exposure, Insecure Direct Object References and Cross-site Scripting based on which application it will work .Some techniques can detect while others can prevent such attacks. It also checks whether it will function on server or client side.

Table.2. Comparison of Other web application attacks

Attack	Banking Application	Online Payment	Applications based on J2EE	Client/Server side
Sensitive Data Exposure[7]	Yes	Yes	No	Client side
Insecure Direct Object References[6]	No	No	Yes	Server side
Cross-site Scripting [5]	Yes	Yes	No	Client side

VI. CONCLUSION

We understand from the study that Amnesia works for all types except for stored procedure attacks, XSchema Validator works for tautology attack and Candid also checks for tautology attack.We also see that banking application and online payment applications are protected from sensitive data exposure attack. The Insecure Direct Object References works for applications based on J2EE but it does not support banking and online payment applications. The XSS attack works for banking and online payment applications. Some of the existing detection and prevention techniques of the above mentioned attacks is studied and analyzed. The study reveals that a lot of improvement can still be done in web application security for obtaining better results.

.REFERENCES

[1] Research On SQL Injection Attack And Prevention Technology",Authors: Li Qian, Jun Lu, 2015 International Conference on Estimation, Detection and Information Fusion(ICEDIF-2015)

[2] An Approach to Detect and Prevent Tautology Type SQL Injection in Web Service Based on XSchema Validation, Author: R. Joseph Manoj, Dr.A.Chandrasekhar ,M.D.Anto Praveena, 2014 International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 3 Issue 1, Jan 2014 Page No. 3695-3699

[3] Detection and Prevention of SQL Injection Attacks on Database Using Web Services Author:Laxman Singh, Sushmit M Vishwakarma, Yashvant Ugalmugale, Prasad Naikwade, 2014International Journal of Emerging Technology and Advanced Engineering(ISSN 2250-2459,ISO 9001:2008 Certified Journal,Volume 4,Issue4,April 2014).

[4] "CANDID: Dynamic Candidate Evaluations for Automatic Prevention of SQL Injection Attacks", Author:Prithvi Bisht, P.Madhusudan,V.N. Venkatakrishnan ,ACM Journal

[5] "Enhanced Browser Defense For Reflected Cross-site Scripting",Author : Bhawna Mewara, Sheetal Bairwa, Jyoti G, Vinesh Jain ,2014 978-1-4799-6896-1/14/$31.00 ©2014 IEEE

[6] "Identification And Illustration Of Insecure Direct Object References And Their Countermeasure", Author: Ajay Kumar Shrestha,Pradip Singh Maharjan,Santhosh Paudel,2015 International Journal of Computer Applications (0975 – 8887) Volume 114 – No. 18, March 2015

[7] Sensitive Data Exposure Prevention using Dynamic Database Security Policy, Author: Jignesh Doshi,Bushan Trivedi ,2014 International Journal of Computer Applications (0975 – 8887) Volume 106 – No. 15, November 2014

[8] "A Survey On Web Application Attacks", Author : Gurvinder Kaur Pannu, 2014 International Journal Of Computer Science and Information Technology,Vol.5(3),2014,4162-4166.

[9] "A Detailed Survey On Various Aspects Of SQL Injection In Web Applications: Vulnerabilities, innovative Attacks And Remedies", Author : Diallo Kindy,al-sakib Khan Pathan, 2012 International Journal

[10] Tracking Known Security Vulnerabilities in Proprietary Software Systems, Author: Mircea Cadariu, Eric Bouwers, Joost Visser, Arie van Deursen, 2014 TUD-SERG-2014-022.

[11] E. Kirda, C. Kruegel, G. Vigna and N. Jovanovic, "Noxes: A client-side solution for mitigating cross site scripting attacks," In Proceedings of the 21st ACM symposium on Applied computing, ACM, (2006), pp. 330-337.

[12] Y. W Huang, F. Yu, C. Hang, C. H. Tsai, D. Lee and S. Y. Kuo, "Verifying Web Application using Bounded Model Checking," In Proceedings of the International Conference on Dependable Systems and Networks, (2004).

[13] Y.-W. Huang, S.-K. Huang, T.-P. Lin and C.-H. Tsai, "Web application security assessment by fault injection and Behavior Monitoring," In Proceeding of the 12th international conference on World Wide Web, ACM, New York, NY, USA, (2003), pp.148-159.

[14] D. Balzarotti, M. Cova, V. Felmetsger, N. Jovanovic, E. Kirda, C. Kruegel and G. Vigna, "Saner: Composing Static and Dynamic Analysis to Validate Sanitization in Web Applications," In IEEE symposium on Security and Privacy, (2008).

[15] G. Wassermann and Z. Su, "Static detection of cross-site Scripting vulnerabilities," In Proceeding of the 30th International Conference on Software Engineering, (2008) May.

[16] T. Jim, N. Swamy and M. Hicks, "BEEP: Browser-Enforced Embedded Policies," In Proceedings of the 16th International World Wide Web Conference, ACM, (2007), pp. 601-610.

[17] Z. Su and G. Wassermann, "The essence of command Injection Attacks in Web Applications," In Proceeding of the 33rd Annual Symposium on Principles of Programming Languages, USA: ACM, (2006) January, pp. 372-382.

[18] T. Pietraszek and C. V. Berghe, "Defending against Injection Attacks through Context-Sensitive String Evaluation", In Proceeding of the 8th International Symposium on Recent Advance in Intrusion Detection (RAID), (2005) September.

[19] P. Wurzinger, C. Platzer, C. Ludl, E. Kirda and C. Kruegel, "SWAP: Mitigating XSS Attacks using a Reverse Proxy", ICSE Workshop on Software Engineering for Secure Systems, IEEE, (2009), pp. 33- 39.

[20] P. Bisht and V. N. Venkatakrishnan, "XSS-GUARD: Precise dynamic prevention of Cross-Site Scripting Attacks," In Proceeding of 5th Conference on Detection of Intrusions and Malware & Vulnerability Assessment, LNCS, vol. 5137, (2008), pp. 23-43.

[21] Z. Su and G. Wassermann, "The essence of command Injection Attacks in Web Applications," In Proceeding of the 33rd Annual Symposium on Principles of Programming Languages, USA: ACM, (2006) January, pp. 372-382.

[22] "Preventing Persistent Cross-Site Scripting (XSS) Attack By Applying Pattern Filtering Approach", Authors : Imran Yusof and Al-Sakib Khan Pathan,2014.

[23] "Detecting Cross Site Scripting Vulnerabilities Introduced by HTML5" , Authors: Guowei Dong.et.al , 2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE).

[24] "Protecting Cookies from Cross Site Script Attacks Using Dynamic Cookies Rewriting Technique", Authors : Rattipong Putthacharoen, Pratheep Bunyatnoparat, 2011 ICACT2011.

[25] "An Execution-flow Based Method for Detecting Cross-Site Scripting Attacks",Authors : Qianjie Zhang, Hao Chen, Jianhua Sun.

[26] "Developing a Security Model to Protect Websites from Cross-site Scripting Attacks Using Zend Framework Application",Authors : Yousra Faisal Gad Mahgoup Elhakeem and Bazara I. A. Barry, 2013 International Conference On Computing, Electrical And Electronic Engineering (ICCEEE).

[27] " CIDT: Detection of Malicious Code Injection Attacks on Web Application",Authors : Atul S. Choudhary and M. L. Dhore, International Journal of Computer Applications (0975 – 8887) Volume 52– No.2, August 2012.

[28] "Prevention of Cross Site Scripting with E-Guard Algorithm",Authors [:M. James Stephen.et.al, International Journal of Computer Applications (0975 – 8887)Volume 22– No.5, May 2011.