ISSN 2395-1621

Advance Detecting & Preventing Intrusions In Multi-Tier System

[#1]Seema Gajare, [#2]Prof.Baban Thombre.

[1]gajareseema@gmail.com
[2]babanthombre@gmail.com

[#1]Department of Computer Engineering
[#2]Asst. Professor, Department of Computer Engineering

Shree Ramchandra College Of Engineeering, Lonikand,
Pune, India.

ABSTRACT

Fast internet growth and increase in number of users make network security essential in recent decades. Lately one of the most hot research topics in network security is Intrusion Detection Systems (IDSs) which try to keep security at the highest level. SQL Injection (SQLI) is a type of attack that targets the back-end of the web application. Attacker crafts the query in such a way that once it is fused with web application's request it acts upon the database related with that web application revealing crucial data from database. All the SQL queries can be used effectively to exploit web application logic to get access to the database. SQL Injection is most of the times outcome of insufficient input validations. This type of attack is generally injected through text boxes which pass data to web application via web requests or sometimes using URLs. In Cross Site Scripting (XSS) attacks, attacker tries to inject a client side scripting to the remote server. XSS generally attacks the HTML of the web page being loaded. ActiveX, JavaScript can be affected by this attack. XSS can reveal cookie information. The scripts can be hosted somewhere by the attacker. Attacker provides a link to the users that looks genuine but has malicious script code. Once user reaches the link script run on client's machine allowing attacker to gain vital information. We also provide more security on backend system. When any attacker directly attack on back end database, then our tempering system also running on server for detecting that activity. Once any changes made then tempering system detect that attack and successfully restored that initial value.

Keywords: Intrusion Detection Systems (IDS), SQL Injection (SQLI) attack, Cross Site Scripting (XSS) attack, Tempering, Security.

ARTICLE INFO

Article History

Received: 26th November 2016

Received in revised form :

26th November 2016

Accepted: 28th November 2016

Published online :

29th November 2016

I. INTRODUCTION

SQL injection attack tries to alter the databases via modified input strings provided to the web applications. Input can be crafted in such a way that SQL queries can bypass the webserver and affect the database. SQL Injection is a type of attack that targets the backend of the web application. Cross Site Scripting (XSS) attacks uses the tags such as script tags to obfuscate appearance and HTML format of web application. Absence of input validation leads to such attacks.

This paper presents a technique and automated finding security vulnerabilities in Web applications. Multi-user web applications are responsible for handling much of the business on today's internet. Such applications often manage sensitive data for many users, and that makes them attractive targets for attackers: up to 70% of recently reported vulnerabilities affected web applications [4]. Therefore, security and privacy are of great importance for web applications.

Two classes of attacks are particularly common and damaging. In SQL Injection, the attacker executes malicious database statements by exploiting inadequate validation of data flowing from the user to the database. In Cross Site Scripting, the attacker executes malicious code on the victim's machine by exploiting inadequate

validation of data flowing to statements that output HTML.[4]

Web Application:

Three tier web applications consist of presentation logic, business logic and data logic. Presentation logic is where User Interface (UI) is developed using which users initiate web requests. Business logic is where the validations and web service functionalities are written. Data logic is related with all the database queries generated as a result of web requests.

Attacks:

Enormous types of attacks are possible on the web applications. Here Cross Site Scripting attacks and SQL Injection attacks are covered for which a model is proposed in next section. It reveals crucial data from database.

SQL Injection Attacks:

SQL injection is a type of attack that targets the back-end of the web application. Attacker crafts the query in such a way that once it is fused with web application's request it acts upon the database related with that web application revealing crucial data.

II. LITERATURE SURVEY

"A Novel Approach for Detecting of SQL Injection and Cross Site Scripting Attacks", 2015. In SQL Injection, attacker exploits security vulnerabilities of the web application to alter the valid SQL query designed by the programmer. The impact of this attack varies according to the SQL queries being injected. XSS is a web application attack where attacker crafts a Uniform Resource Locator (URL) in such a way that it seems to be legit, but in fact it is not. It's like a trap attack, in which once the user visits this crafted URL the attacker executes some malicious code in user's browser. [1]

"Web Application Intrusion Detection System for Input Validation Attack", 2008, a network-based intrusion detection system is proposed the system classified the normal connections and attacks. After detection of attack, type of attack is determined by the system in detail. Using conjugated training function and validation dataset caused: faster training, less overhead, less memory consumption and over-fitting prevention. [2]

"Double Guard: Detecting Intrusions in Multitier Web Applications", 2012, in this paper, the present Double Guard, an IDS system that models the network behaviour of user sessions across both the front-end webserver and the back-end database. By monitoring both web and subsequent database requests, are able to ferret out attacks that independent IDS would not be able to identify. Furthermore, we quantify the limitations of any multitier IDS in terms of training sessions and functionality coverage. [3]

"On Security Issues in Web Applications through Cross Site Scripting (XSS)", 2013, in this paper Cross Site Scripting (XSS) is the major threat for web application as it is the most basic attack on web application. It provides the surface for other types of attacks like Cross Site Request Forgery, Session Hijacking etc. There are three types of XSS attacks i.e. non-persistent (or reflected) XSS, persistent (or stored) XSS and DOM-based vulnerabilities. There is one more type that is not as common as those three types, induced XSS. In this work the aim to study and consolidate the understanding of XSS and their origin, manifestation, kinds of dangers and mitigation efforts for XSS. [4]

"Automatic Creation of SQL Injection and Cross Site Scripting Attacks", 2009, the presented a technique for creating SQL injection and cross-site scripting (XSS) attacks in Web applications and an automated tool, Ardle, that implements the technique for PHP. Our technique is based on input generation, dynamic taint propagation, and input mutation to find a variant of the input that exposes a vulnerability. [5]

"Intrusion Detection Technique by using K-means, Fuzzy Neural Network and SVM classifiers", 2013 [6], in this paper the proposed intrusion detection is an important component in network security. IDS helps the information security community by increasing detection efficiency, reducing the manpower needed in monitoring and helping to learn new vulnerabilities by providing legal evidence [6].

III. EXISTING SYSTEM

Figure 1 show the overall system architecture. Web applications are deployed over a network. Clients will access the web application via a web server. A web service will be given to each client separately. Client will send a web request using UI of web application, based on the web request, database query will be generated and data will be fetched back to the particular client.

The attack on the system will always be in the form of web request. It may contain complicated tags like <script>, <iframe> etc. that will be targeted to attack the client's web browser as a form of XSS attack or a well-structured SQL query that will attack over the database of the web application forming a SQL Injection attack

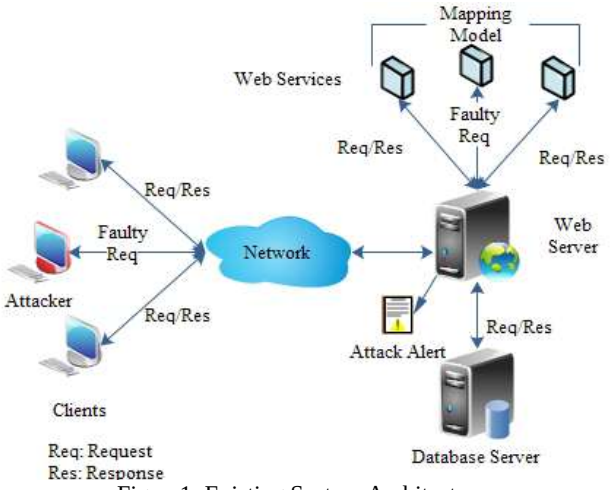

Figure1: Existing System Architecture

The system is differentiated based on request and query mapping model. Two web applications are developed one static web application for deterministic mapping model of request and query and other dynamic application for non-deterministic mapping model.

Web application in which the query generated is same for each web request regardless the time or parameters such applications are defined here as static. For example the download link for one file remains same, once that file is uploaded to any web application. Similar is the case for images and other non-changing data but web applications like social networking sites, blog sites etc. where queries generated may change based on time or data being passed through the same web request, such web applications are defined as dynamic web applications.

In static web application, in which responses generated to a web request are fixed. So it will be generating a fixed query set for static web applications. This static mapping is used for detection of attacks. Any varied response than the expected one is considered as potential threat. That can be later evaluated by admin. In case of dynamic web application the query responses may vary time to time based on parameters or time at which page is loaded. For dynamic web applications non-deterministic query set must be considered where each set may have different database queries.

For both static and dynamic web application, we are generating mapping model by passing the web application through practice sessions. Based on practice sessions of static web application valid download query will be stored as true mapping for that particular file. In case of dynamic web application, each activity of a valid user can be categorized. Each web request can be categorized, considering a web application of a blog, like login, adding a web article, reading an article, commenting on that article etc. can be used to differentiate the legit mapping. Using this categories non-deterministic mapping can be made accurate up to some extent. This practice session are performed in attack less environment.

Along with the mapping static and dynamic mapping model a robust validation for SQL injection and XSS is provided. A strong parsing of the input is made. XSS attacks are possible via variety of ways like providing malicious URL to users or by including tags like <input>, < DOCTYPE> etc. or even functions like alert () in input web request.

As shown in Figure 2, while using dynamic web application, each client will be allocated a separate web service for him/her. Whenever user will make a web request instead of directly generating a SQL query for directly we are processing this request for any kind of SQLI attacks or XSS attacks based on input parameters. This phase implements a strong input distillation. Input checking is a very critical in case of XSS attacks. Before passing the XSS infected web request directly to the database where it can get permanently stored on data logic of web application, this phase tries to inform the admin of possible XSS or SQLI attacks.

Once input validations are passed, SQL query or queries will be generated. Even this first phase of security is dodged by intelligence of the hacker, the infected web request will be sent for mapping pattern matching. Now this generated SQL query or queries will be checked for mapping pattern. Matching will be done on the basis of web request that is diseased with either SQLI or XSS attack, and the subsequent web query generated which includes the same attack bypassed from previous phase. If matching valid pattern are found then only that query will be fired over database to fetch the results otherwise will be treated as potential threat and can be accessed by admin If the query is bypassed from the second step the generated query will affect the database whether or not that query is infected. If that query is in fact infected system will fall to XSS or SQL injection attack. Alerts can be generated via email or text messages based on the results of mapping model.

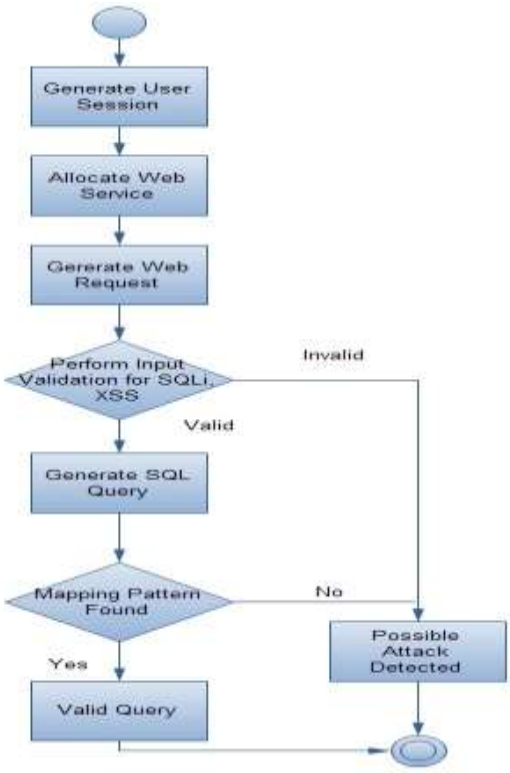

Figure 2: Flow for Dynamic Web Application

IV. PROPOSED SYSTEM

A. System Architecture

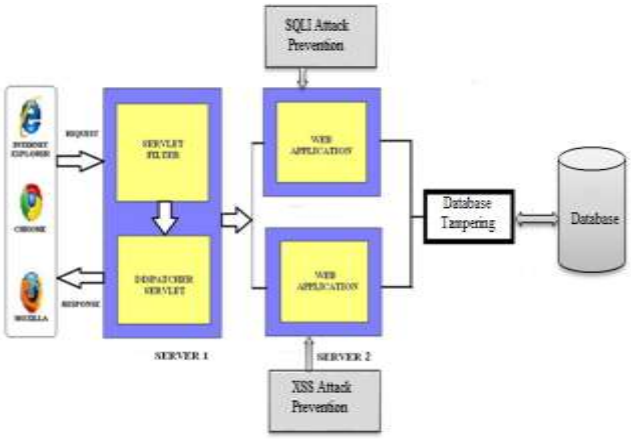

Fig 3 : Detecting and Preventing Intrusions in Multi-Tier System with Tempering Detection

Our aim to enable strong data detection and protection for web applications while at the same time we minimize the false positive rate. Our objective to secure three tier web applications for detecting and preventing different types of attacks. Detecting the tempering attack for database activity. Provide both side security front-end and back- end.

Many Systems are providing one way security for the web applications protecting a web application in terms of interface and at database end with proper recovering options is best part of the system. The proposed system designs idea in breakdown model to evaluate security of the web applications along with its database in every step.

Module Explanation:

User Module:
User is having login access. He is authorized user. He can update all personal information.

Sales Department Module:
Sales department work as a hacker. Here hacker change the database value of any product without authentication.

Admin Module:
Admin is the authorized person. He checks all the user activity records as well as profiles. He also observes the tempering on changing the values from database.

B. Proposed Mathematics

S= {I, O, F, S1, S2}

Where

S = System.

I = Incoming user requests changing database value.

O = Detect tempering on unauthorized updating database.

F = {f1, f2, f3, f4}.

where,

f1 = SQL Injection Methodology.

f2 = XSS attack detection.

f3 = Tempering detected.

f4 = Successfully restored original value.

S1 = Initial state is the state in which system is waiting for incoming user requests.

S2 = Final state is the detect tempering and restored successfully.

V. CONCLUSION

In this paper we have identified the threats of SQL Injection and XSS attack using Intrusion Detection System. Additional security measures can be provided using stored procedures. This approach applies mapping model to detect SQL Injection and XSS attacks. Also we have identified the tempering attack on database.

We have achieved this by isolating the flow of information from each web server session with a virtualization technique. Also, we have quantified the detection accuracy of our approach when we attempted to model static and dynamic web requests with the back-end file system and database queries.

REFERENCES

[1] Piyush A. Sonewar, Nalini A. Mhetre, “A Novel Approach for Detecting of SQL Injection and Cross Site Scripting Attacks”, International Conference on Pervasive Computing, 2015.

[2] Y J Park, J C Park, "Web Application Intrusion Detection System for Input Validation Attack", Third International Conference on Convergence and Hybrid Information Technology, 2008.

[3] Meixing Le, Brent ByungHoon Kang, “DoubleGuard: Detecting Intrusions in Multitier Web Applications”, IEEE Transaction on Dependable and Secure Computing Vol. 9, No. 4, July/August 2012.

[4] V. K. Malviya, S. Saurav, “On Security Issues in Web Applications through Cross Site Scripting (XSS)”,20th Asia-Pacific Software Engineering Conference,2013

[5] A. Kiezun, M. D. Ernst, “Automatic Creation of SQL Injection and Cross-Site Scripting Attacks”, ICSE, May 16-24, 2009.

[6] A. M. Chandrasekhar, K. Raghuveer "Intrusion Detection Technique by using K-means, Fuzzy Neural Network and SVM classifiers", 2013 International Conference on Computer and Informatics(ICCCI),Coimbatore, INDIA, Jan04-06,2013

[7] Lwin Khin Shar, Hee Beng Kuan Tan, "Automated removal of cross site scripting vulnerabilities in web applications", Information and Software Technology 54, 467–478, 2012.

[8] R.Priyadarshini, Jagadiswaree D, Fareedha. A, Janarthanan M, "A Cross Platform Intrusion Detection System using Inter Server Communication Technique" , IEEE-International Conference on Recent Trends in Information Technology, ICRTIT IEEE MIT, Anna University, Chennai. June 3-5, 2011.

[9] A. M. Chandrasekhar, K. Raghuveer "Intrusion Detection Technique by using K-means, Fuzzy Neural Network and SVM classifiers", 2013.International Conference on Computer and Informatics (ICCCI), Coimbatore, INDIA, Jan04-06,2013.

[10] R. Ludinard , E Totel,"Detecting Attacks against data in Web applications", Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on Digital Object Identifier, Page(s): 1 - 8, 2012.

[11] T. V. Narayan Rao, V. Tejaswini, K. Preethi, "Defending Against Web Vulnerabilities and Cross Site Scripting", JGRCS, Vol. 3, No.5, May2012.

[12] Debasish Das, Utpal Sharma, D K Bhattacharyya, "A Web Intrusion Detection Mechanism based on Feature based Data Clustering", IEEE International Advance Computing Conference (IACC) Patiala, India, 6-7, March 2009.

