ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

Web Gate Keeper: Detecting Encroachment in Multi-tier Web Application

Sanaz Jafari

Research Scholar Department of computer engineering Bharati Vidyapeeth Deemed University, Pune, India

Prof.Dr.Suhas H. Patil (GUIDE)

Head Of the Department of Computer Engineering Bharati Vidyapeeth Deemed University, Pune, India

ABSTRACT

The Internet services and different applications become vital part of every person's life. Web services and applications have smoothened the way of users for the faster data access. To meet the user's requirements for faster data retrieval, the web applications have moved towards multi-tier design where in the back-end server contains the data base and application interface or web server acts as the front end. The wide-spread use of web services has made them a target of attackers. Web Gate Keeper provides Intrusion Prevention Systems at both the ends (web server and database server). The prevention logic of our system works on session tracking and control. Through these Web Gate Keeper provides a secure environment for the application. Web Gate Keeper provides security for multi-tier web applications. Previously, IPSs were developed for Web servers and database servers separately. This scenario leads to the need of a system that prevents both of these simultaneously. The project aims at developing a system to provide complete security from external intruder.

Key words: external Intruder, Intrusion Prevention, IPSs, Multi-tier Systems, session tracking.

1. INTRODUCTION

From last some years these internet services & web applications become very popular among the users. Because of the popularity of all these services & applications comes with the some problems. So because of this our day to day needs such as banking, shopping & networking become are done only depends on web only. The services which user going to use it its work on two ends front end & back end, on front end because of the user interface user can make use of it & at the back end all the data which user is going to use are going to stored in the data server. The target of the attack is focuses on the data server because all the personal & corporate data which is very important information. For attacking attacker needs the data access so they move their concentration from front end to back end, because all the vial information is stored at the back end on server.

Intrusion detection systems have been widely used to detect the attacks which are known by matching misused traffic patterns or signatures to protect the multi tiered web services. The IDS have the power to detect the misuse of the network by tracking the abnormal behavior of the network. Attackers send all the different abnormal network can detected by the IDS database & web. This will not allow the attacker to enter into the server. But if attacker uses the normal traffic to attack the web server it can not be detected by the IDS.

There are two types of network IDS:

1. Anomaly detection
2. Misuse detection

In this IDS alert is generated when it detects the attack. IDS mainly focuses on the analyzing the audited data which is encountered this can be done to avoid the wrong detection. In this environment IDS work very dynamically & decides whether these actions are useful in this environment or not.

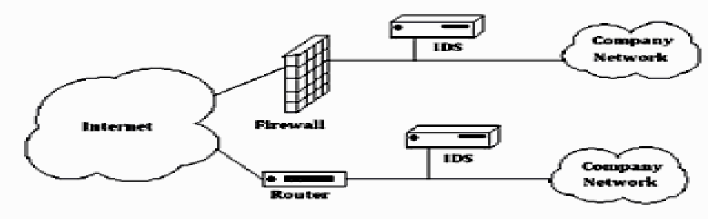


Fig 1.1: Simple Intrusion Detection System.

There are following three measures to evaluate efficiency of Intrusion Detection System:

1. Accuracy – Inaccuracy occurs when an IDS signals that an abnormal action is taken in the given environment.

2. Performance – The performance of the system describes the quality of that system. If the performance of IDS is poor then real time detection is not possible.

3. Completeness – When IDS fails to detect an attack then incompleteness occurs. This is very difficult to evaluate because it is impossible to have a global knowledge about all the attacks

To detect the attacks in multi-tier web services Double Security System is used. In this double security system front end work as a HTTP & back end work as a file or SQL for network transactions, creating normality model of isolated user sessions we are going to use it in this system. This Double Security System uses lightweight virtualization techniques which help to assign user the specific user's web sessions to a dedicated container which provides an isolating virtual environment. So this system will take the every request with the specific container ID. So Double Security System will do the mapping of the accurate profile into proper and accurate account by making use of web server & database traffic.

Open VZ is used for implementing the Web Gate Keeper container architecture. This container based architecture along with casual mapping also provides an isolation which helps in safeguarding the session hijacking attacks of the users. The lightweight virtualization helps in running innumerous copies of the web server instances in different containers. Hence, everyone is differentiated from the rest. Each client is assigned a dedicated container so that even when the attacker attacks the session, it is limited to that session only and doesn't damage the other user session.

1.1 INTRODUCTION TO MULTI-TIER WEB APPLICATION

In this three tier model at the database server its very hard to find which transaction request is send by which user its because of no communication separation between web server & database server. That is shown in following fig. 1.2

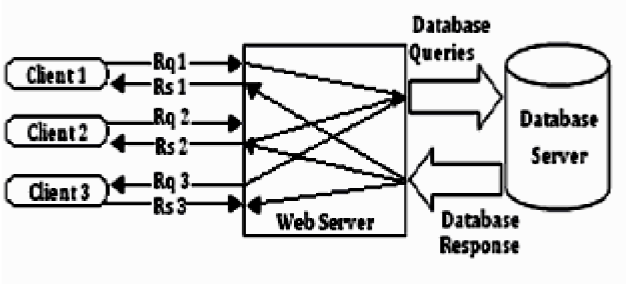
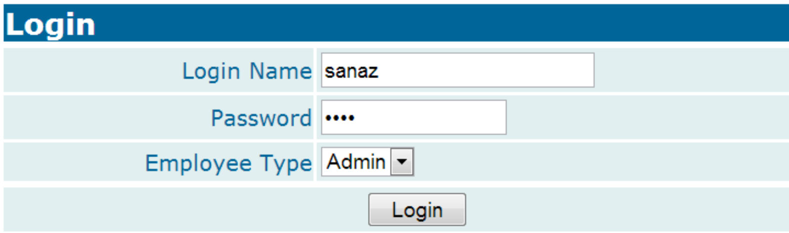


Fig 1.2: Classic three-tier model.

1.2 TYPES OF ATTACKS

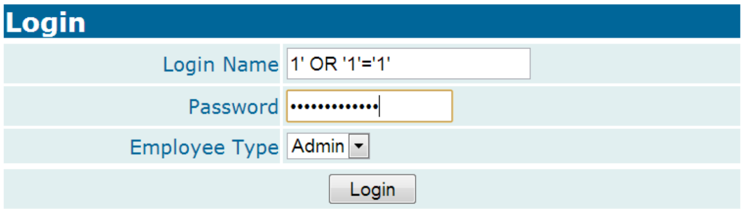
1. INJECTION ATTACK

SQL injection vulnerabilities allow attackers to insert SQL commands as a part of user input. When an SQL query is constructed dynamically with maliciously-devised user input containing SQL keywords, attackers can gain access or modify critical information such as a credit card number in a database without proper authorization. For example, Figure 6 shows a sample program in Java using JDBC that uses a SQL query to authenticate a user via id and password. The query is dynamically created via the program statement in bold. In the query in Figure, id and password are obtained


Login

Login Name	sanaz
Password	••••
Employee Type	Admin

Injection attack can be done as follow -

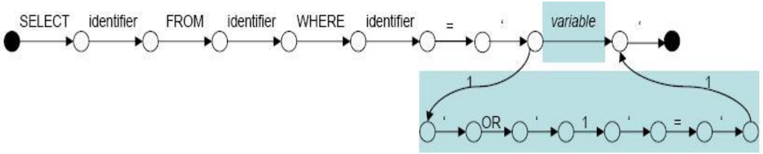

Login

Login Name	1' OR '1'='1'
Password	•••••••••••••
Employee Type	Admin

So here application query can be formed as below –

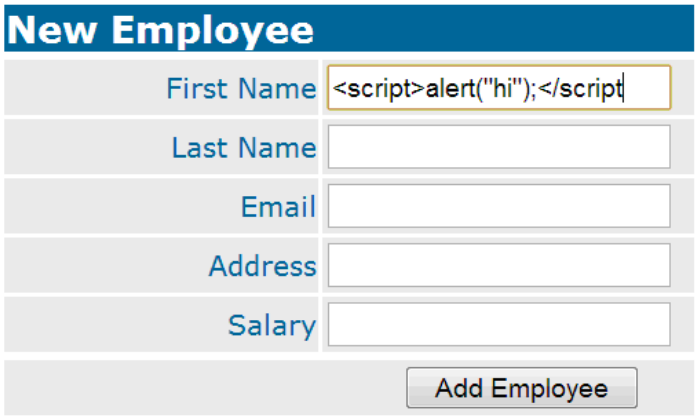
SELECT user info FROM users WHERE id = '1' OR '1' = '1' AND password = '1' OR '1' = '1';
And unauthenticated user will get the access to the system.

SQL state as below=



2. CROSS SCRIPTING ATTACK

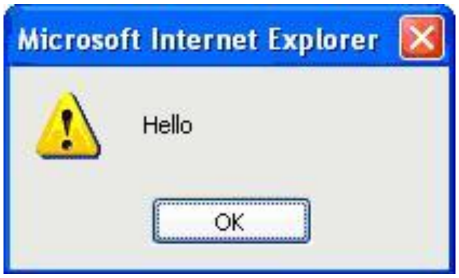
Cross-site scripting (XSS) vulnerabilities allow attackers to insert malicious scripts as a part of user input and the script is executed at other user's browser due to the lack of input validation. We provide a simple example of attacks exploiting XSS vulnerabilities. Consider a search engine that returns the search results including the same query given by a user. If the user input includes a script and the returned result page does not encode the script into HTML code, the script in the returned result page will be executed.



New Employee

First Name	<script>alert("hi");</script
Last Name	
Email	
Address	
Salary	

By adding java script into the inputs fields and submitting the content wil cause the cross script attack in the submition page.

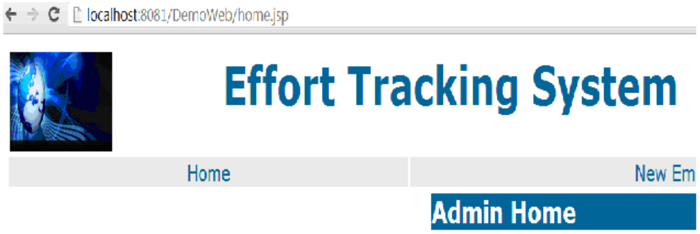


Microsoft Internet Explorer

Hello

3. PRIVILEGE ESCALATION ATTACK

Let's assume that the website serves both regular users and administrators. For a regular user, the web request Ru will trigger the set of SQL queries Qu, for an administrator, the request Ra will trigger the set of admin level queries Qa as shown in Figure 2. Now suppose that an attacker logs into the web server as a normal user, upgrades his/her privileges, and triggers admin queries so as to obtain an administrator data. This attack can never be detected by either the web server IDS or the database IDS since both Ru and Qa are legitimate requests and queries. Our approach can detect this type of attack since the DB query Qa does not match the request Ru, according to our mapping model.



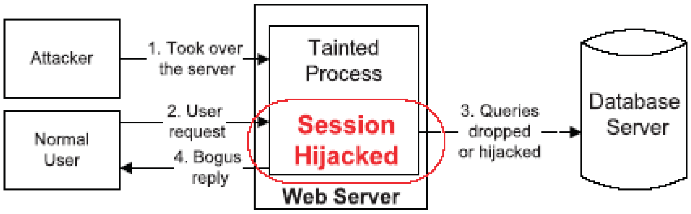
localhost:8081/DemoWeb/home.jsp

Effort Tracking System

By changing the URL of the application this attack can easily done on any system which is not taking care of handling such scenarios.

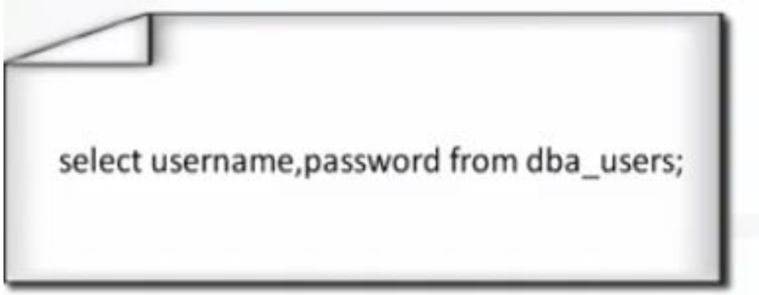
4. HIJACK FUTURE SESSION ATTACK

This class of attacks is mainly aimed at the web server side. An Attacker usually takes over the web server and therefore hijacks all subsequent legitimate user sessions to launch attacks which are shown in Figure 3. For instance, by hijacking other user sessions, the attacker can eavesdrop, send spoofed replies, and/or drop user requests. A session-hijacking attack can be further categorized as a Spoofing/Man-in-the Middle attack, Denial-of-Service/Packet Drop attack, or a Replay attack.



5. DIRECT DB ATTACK

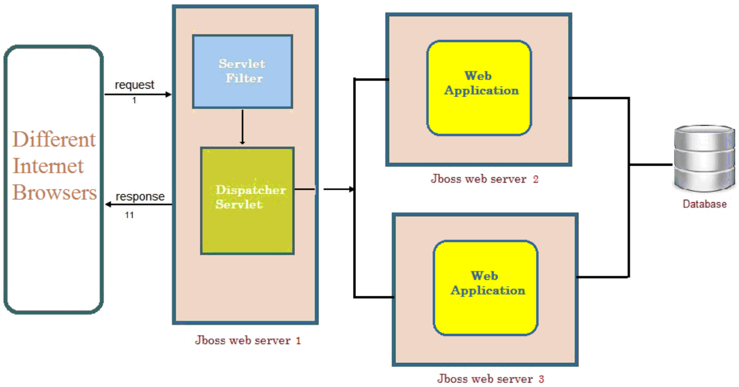
It is possible for an attacker to bypass the web server or firewalls and connect directly to the database which is shown in Figure 5. An attacker could also have already taken over the web server and be submitting such queries from the web server without sending web requests. Without matched web requests for such queries, a web server IDS could detect neither. Furthermore, if these DB queries were within the set of allowed queries, then the database IDS it would not detect it either. However, this type of attack can be caught with our approach since we cannot match any web requests with these queries.



```
select username,password from dba_users;
```

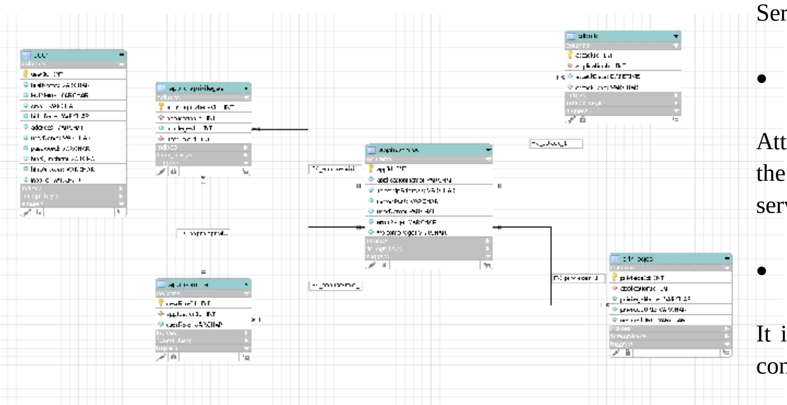
Above query can be fired to DB server residing on the same machine where application server resides.

2. SYSTEM ARCHITECTURE



All request will be processed from Server1's Servlet Filter, It will take care of Session validation and session tracking. Then control goes to Dispatcher Servlet which will take care of dispatching request to appropriate service. Database server will be accessible only to server 2 and server 3 where actual web application is hosted. A web application with use cases having more than one user and n privileges will be implemented to demonstrate above implementation.

Web gatekeeper protects the web application by registering application with the web based system which ask user about the application details, privileges used in the application, and user roles.
User roles will get mapped with the privileges. So to handle this scenario one system needs to be implemented with below database model.



Each request will be processed through the mapped URL servlet. Request Processor Servlet will be responsible to scan each request, verify it for all possible attacks. In case of attack, it will notify it to the user and admin of the application through email. If no attack found then it will create a new request which will be passed to the actual application for the further processing.

Session tracking will be done using the entry page and exit page of the application. User enters to the application without coming from the entry page will be prohibited to enter into the application and redirected to the application error page.

This application can be developed using the Apache Tomcat web server with My SQL Database. As both are open source and minimize the project implementation cost.

3. SYSTEM DESIGN AND WORKING

This Web Gate Keeper is designed in such way no user will have direct access to database server as well as application server on which web application is hosted. First all request which will be processed from Server 1s Servlet Filter, it will take care of session validation and session tracking. Then control goes to dispatcher Servlet which will take care of dispatching request appropriate service. Database server will be accessible only to Server 2 and Server 3 where actual web application is hosted. This helps to prevent different types of attacks on web servers & web applications.

Web Gate Keeper can prevent :

- Privileged Escalation Attack-

The scenario where the user logs in and then upgrades his privilege so that neither the application IPS not the database IPS detect the anomaly

- Hijack Future Session Attack-

An attacker usually takes over the web server and therefore hijacks all subsequent legitimate user sessions to launch attacks which includes Spoofing/Man-in-the-Middle attack, an Exfiltration Attack, a Denial-of Service/Packet Drop attack, or a Replay attack

- Injection Attack-

Attackers can use existing vulnerabilities in the web server logic to inject the data or string content that contains the exploits and then use the web server to relay these exploits to attack the back-end database.

- Direct DB Attack-

It is possible for an attacker to bypass the web server or firewalls and connect directly to the database.

3.1 Hardware

We will be using 3 workstations with the control logic residing on one of them. These workstations will form a network that will be connected with the help of a router.

SOFTWARE

Different web servers are required for speed up the different services.

Therefore an application dependent database server & a separate web server is needed to perform the control logic of the system.

WORKING

when the user want to log into the web application and he put wrong password into that then that respective session retires and user allowed to try it again.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a presentable response.

When a user want to access of a database query, the Web Gate Keeper system look out a special rights of user and then through entitlement service it provides service to the respected user. With the help of entitlement service the upgrading of special rights of the user is prevented. After this also if such an activity is taking place the session is immediately expires and intrusion details are saved in log _le for future reference.

4. PERFORMANCE

Because of the use of 2 different servers we are using in our Web Gate Keeper it will help to speed up the system. Therefore server on which Application resides is a separate one from the server where IDPS resides; hence it won't affect the speed of the application. The control logic in the host controller selects the web server to satisfy the web requests. Hence selection of a web server with minimal pending requests can be made and hence processing of the system speeds up.

5. SECURITY

In our System, we are storing the vital information about the application (for which the system will work) in encrypted and secure format. Also the admin details will be safely stored. And as the system itself works for the security this information will not be easily accessible according to our architecture.

6. CONCLUSION

We presented an Intrusion Detection System that builds models of normal behavior for multi-tier web applications from both front-end web (HTTP) requests and back-end database (SQL) queries. We have shown that such correlation of input streams provides a better characterization of the system for anomaly detection because the intrusion sensor has a more precise normality model that detects a wider range of threats. Furthermore, we 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. It is an application independent system and hence dynamic web server which provides better security for data and web application.

7. REFERENCES

[1] Meixing Le, AngelosStavrou, Brent ByungHoon Kang," Double Guard: Detecting Intrusions in Multitier Web Applications", IEEE Transactions on dependable and secure computing, vol. 9, no. 4, July/august 2012.

[2] F. Valeur, G. Vigna, C. Kru¨ gel, and R.A. Kemmerer, "A Comprehensive Approach to Intrusion Detection Alert Correlation," IEEE Trans. Dependable and Secure Computing, vol. 1, no. 3, pp. 146-169, July-Sept. 2004.

[3] Open, http://wiki.openvz.org, 2011.

[4] Joomlacms, http://www.joomla.org/, 2011.

[5] http://www.dummies.com/how-to/ content/ examining-different types-Of-intrusion-detection s.html

[6] http://advanced-network-security .blogspot .in/2008/04/threemajor-Types-of-ids.html

[7] M. Cova, D. Balzarotti, V. Felmetsger, and G. Vigna, "Swaddler: An Approach for the Anomaly-Based Detection of State Violations in Web Applications," Proc. Int'l Symp. Recent Advances in Intrusion Detection (RAID '07), 2007.

[8] Karen scarfone, Petermell,"Guide to Intrusion Detection and Prevention Systems (IDPS)", NIST National institute of Standards & Technology (Technology Administration U.S. Department of commerce, Special Publication 800-94

[9] http://www.omnisecu.com/security/infrastructure-and-emailsecurity/ Types-of-intrusion-detection-systems.htm.

[10] http://en.wikipedia.org/wiki/Multitier_architecture#Comparison_ With _ MVC _ architecture.

Prof. Dr. Suhas H. Patil is Head of the Department of Computer Engineering of Bharati Vidyapeeth University, Pune, India. He is having rich experience of Computer Engineering and Information Technology. He is having more than 22 years of academic experience. He obtained Ph.D. and M.E. He has published more than 130 research papers in reputed journals and proceedings of international and national conferences.

Miss. Sanaz Jafari is pursuing Master of Technology in Computer Department of Bharati Vidyapeeth University, Pune, India. Her areas of interest are Software Engineering. She is currently doing her thesis work at Bharati Vidyapeeth University, Pune, India.