ISSN 2395-1621

Security System For Multi-Tier Web Application

[#1]Snehal Gaikwad, [#2]Pratiksha Ingavale, [#3]Shruti Yewale

[1]snehal44gaikwad@gmail.com
[2]ingavale.pratiksha@gmail.com
[3]shruti.yewale@gmail.com

[#123]Department of Computer Engineering, RMDSSOE, Pune

ABSTRACT

Now-a-days usage of internet has increased for various purposes like online shopping, online transaction, internet banking, etc. Almost everything is done online. With this increased usage of internet, websites are prone to attacks. Security system is nothing but an Intrusion Detection System (IDS) that models the network behavior of user sessions. It protects both the front-end web server as well as back-end database. It monitors both web and subsequent database requests. So, it is possible to identify attacks that independent IDS would not be able to identify. Our contribution is to find leaked data which is done by hacker. Next steps to detect the sql injection attack also for preventing Unauthorized access users.

Keywords— Anomaly detection, virtualization, multi-tier web application, data leakage detection, SQL injection, DDOS atttack.

ARTICLE INFO

Article History

Received :5th April 2016

Received in revised form :

7th April 2016

Accepted : 9th April 2016

Published online :

11th April 2016

I. INTRODUCTION

Web services are widely used by people. Web services and applications have become popular and also their complexity has increased. Most of the task such as banking, social networking, and online shopping are done and directly depend on web. As we are using web services which is present everywhere for personal as well as corporate data they are being attacked easily. Attacker attacks backend server which provides the useful and valuable information thereby diverging front end attack.

Intrusion detection systems have been widely used which is able to detect the attacks. It protect the multi-tiered web services by matching misused traffic patterns or signatures . Security System models isolated user sessions which include both the web front-end as HTTP and back-end as File . In our system, we are using lightweight virtualization technique which will assign each user's web session to a dedicated container . Here, we will match each web request with its subsequent database queries which will be associate with the accurate container ID. Separate ID is assigned to each container. The container based web architecture provides isolation that will be helpful in detecting Future Session-Hijack attacks. In lightweight virtualization environment there are various containers for running multiple instances of web server. These containers are separated from each other. Containers can be easily created and destroyed for each and it lasts for only short time. When attacker attacks single user session then it will not affect other user sessions.It is our main objective to create software that is usable, intuitive, simple, and functions well consistently.

SQL injection

In SQL injection, attacker executes malicious SQL statements and controls the web application's database server. Because of SQL injection, attacker can add, modify and delete records in a database which will affect data integrity. In this way, attacker gains unauthorized access to any sensitive data .

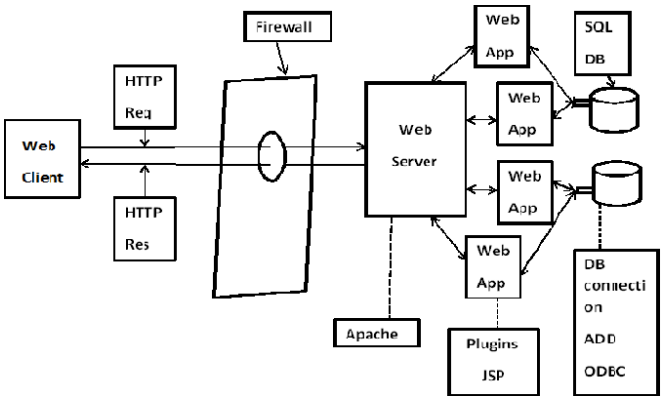
DDOS ATTACK

A Distributed Denial of Service (DDOS) attack makes an online service unavailable by overloading it with traffic from multiple sources . The malicious hacker does this by imposing a group of remotely-controlled computers to send a flood of network traffic to the target. The target becomes so busy to serve attacker's requests that it doesn't have time to respond to legitimate users' requests.

Data Leakage

Data leakage is the big issue for industries & different institutes. It is very hard for any system administrator to find out the data leaker among the system users. It is creating a serious threat to organizations. It can destroy company's brand and its reputation.

The system architecture of our security system is as shown below:



System Architecture

Data leakage algorithm:

Input: Input data D = D1, D2, D3,……..,Dn saves into the hash table.

Step1: Arrange all input data into matrix format (save into log files).

Step2: Consider m as a selected data act as a new selected data.

Step3: m position gets changed after allocated time period.

Step4: If M's data get hacked.

Step5: Data leakage is occurs.

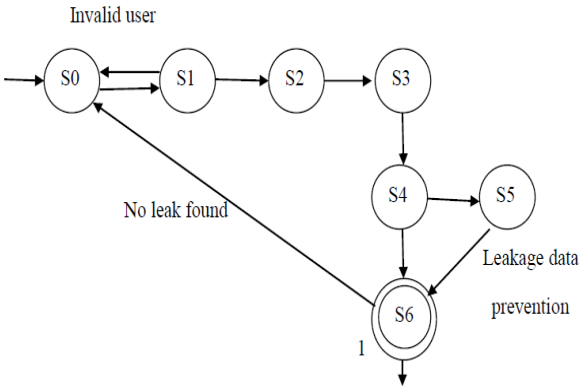
Step6: We have to check the leakage data and prevent it.

Step7: Using Revert back function we have to get original data.

Step8: When user calls that corrupted file, hash function gives to usera previous data.

Step9: Return True.

II. MATHEMATICAL MODEL



Let S be the security system,

S={S0, S1, S2, S3, S4, S5, S6}

Input={User login}

Output={Intrusion detection & prevention}

Where,

S0 - User data

S1 - Read data (signature verification)

S2 - Check for threshold

S3 - Attack model identification

S4 - Data leakage detection

S5 - Leakage data prevention

S6 - Intrusion detection

Success Conditions : Intrusions detected

Failure Conditions : Intrusions not detected

III. RESULT

SQL injection attack

Whenever attacker injects malicious query into the webserver, he will not get access to the database. The message will be displayed on the screen as Sql Injection Attack Detected.

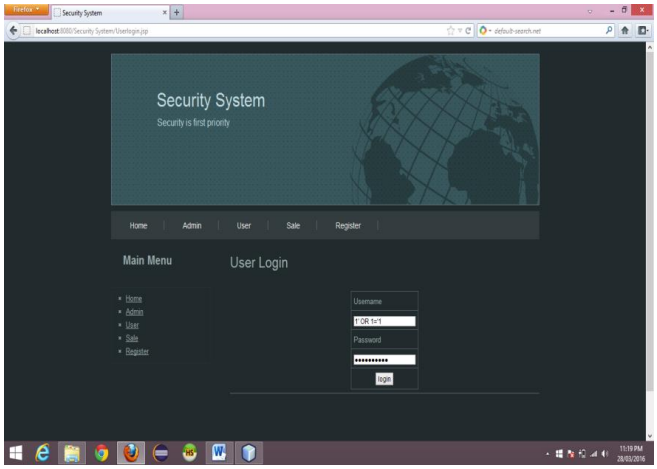


Fig SQL injection query

DDOS attack:

Whenever user tries to upload file whose size greater than threshold limit then it will not be uploaded. Error message will be displayed on the screen.

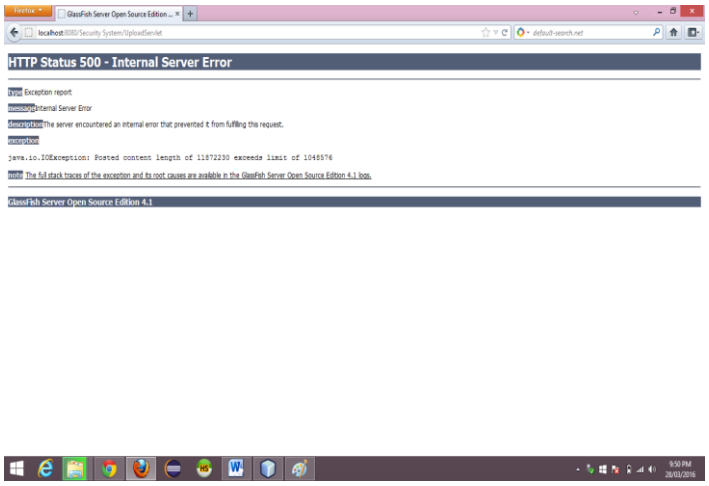
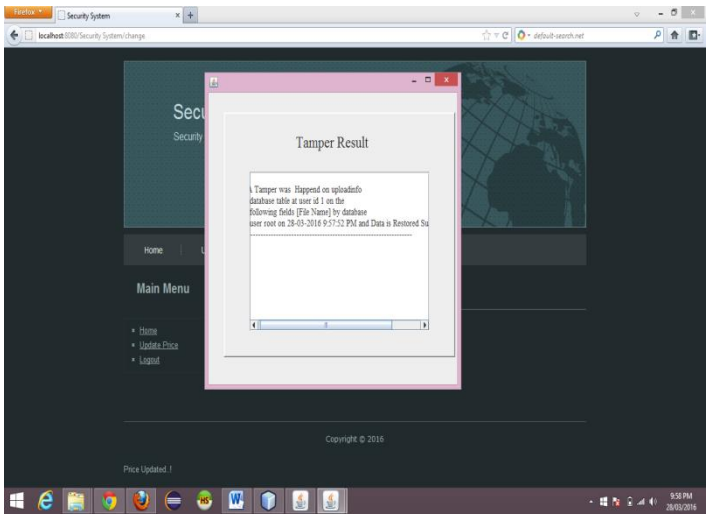


Fig DDOS attack Error.

Data Leakage detection:
When attacker tries to update the price of product then message will be displayed on the screen. Also, price will be restored to original value.



IV. CONCLUSION

In recent there is more trend toward use of web services due to their functionality and scalability. But the major problem is requirement of preserving the privacy. The proposed system fulfils this requirement of security. An intrusion detection system models normal behavior for multi-tiered web applications from both front-end web (HTTP) requests and back-end database (SQL) queries. As compared to previous approaches that correlated or summarized alerts generated by independent IDSs, Security system forms container-based IDS with multiple input streams to produce alerts. Hence, this security system is able to identify a wide range of attacks with minimal false positives..

REFRENCES

[1] SANS, "The Top Cyber Security Risks," http://www.sans.org/ top-cyber-security-risks/, 2011.

[2] National Vulnerability Database, "Vulnerability Summary for CVE-2010-4332," http://web.nvd.nist.gov/view/vuln/detail? vulnId= CVE-2010-4332, 2011.

[3] National Vulnerability Database, "Vulnerability Summary for CVE-2010-4333," http://web.nvd.nist.gov/view/vuln/detail? vulnId=CVE-2010-4333, 2011.
[4]Autobench,http://www.xenoclast.org/autobench/, 2011.

[5] "Common Vulnerabilities and Exposures," http://www.cve. mitre. org/, 2011.

[6] "Five Common Web Application Vulnerabilities," http://www. symantec.com/connect/articles/five-common-web-application-vulnerabilities, 2011.

[7] H. Debar, M. Dacier, and A. Wespi, "Towards a Taxonomy of Intrusion-Detection Systems,"Computer Networks, vol. 31, no. 9, pp. 805-822, 1999.

[8] Verwoerd and R. Hunt, "Intrusion Detection Techniques and Approaches," Computer Comm., vol. 25, no. 15, pp. 1356-1365, 2002.

[9] C. Kruegel and G. Vigna, "Anomaly Detection of Web-Based Attacks," Proc. 10th ACM Conf. Computer and Comm. Security (CCS ‟03), Oct. 2003.

[10] .Vigna, W.K. Robertson, V. Kher, and R.A. Kemmerer, "A Stateful Intrusion Detection System for World-Wide Web Ser-vers," Proc. Ann. Computer Security Applications Conf. (ACSAC ‟ 03), 2003.

[11] 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‟ lSymp. Recent Advances in Intrusion Detection (RAID ‟ 07), 2007.

[12] M. Roesch, "Snort, Intrusion Detection System," http://www.snort.org, 2011.

[13] A. Stavrou, G. Cretu-Ciocarlie, M. Locasto, and S. Stolfo, "Keep Your Friends Close: The Necessity for Updating an Anomaly Sensor with Legitimate Environment Changes," Proc. Second ACM Workshop Security and Artificial Intelligence, 2009.

[14] Openvz, http://wiki.openvz.org, 2011.

