ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Advance Detecting and Preventing Intrusions in Multi-Tier System

Seema V. Gajare, Prof. Baban H. Thombre

PG Student, Department of Computer Engineering, Shree Ramchandra College of Engineering, Lonikand, Pune, India

Assistant Professor, Department of Computer Engineering, Shree Ramchandra College of Engineering, Lonikand, Pune, India

ABSTRACT: Web applications have become one of the most significant ways to provide a broad range of services to users. Internet services and applications have become an inseparable part of daily life, enabling communication and themanagement of personal information from anywhere. To accommodate this increase in application and data complexity, web serviceshave moved to a multi-tier design wherein the webserver runs the application front-end logic and data are outsourced to a databaseor file server.As more and more sensitive data is available over the internet,attackers or hackers are becoming more fascinated in such data revealing which can cause immensedamage. Their clever actions and detailed technical knowledge help them access information we really don't want them to have. Web application attacks are the most rampant and vast security threats. Intrusion Detection and Prevention Systems (IDPS) are essential for preventing network attacks. Attacks like SQL Injection (SQLI) and Cross Site Scripting (XSS) are accountable for the largest security breaches. SQL Injection occurs when user input is not clarified for escape characters which is then provided to an SQL statement. It shows results in the dormant handling of the statements performed on the database by the end user of the application. If XSS vulnerability is present on a website, then an attacker can craft code that executes when other or new users open the same website, leads to users to interact with the malicious background entity created by the attacker. The objective of this system is to detect and prevent attacks such as SQL Injection, Cross Site Scripting and to provide both front-end and back-end security. This system not only detects tamper on database but also secures three-tier web application. This system minimizes false positive rate and restores the original values in the database.

KEYWORDS: Cross Site Scripting Attack, Intrusion Detection and Prevention Systems, Multi-Tier or Three-Tier Web Application, Security, SQL Injection Attack, Tamper

I.INTRODUCTION

Web delivered services and applications have improved in both popularity and complexity over the past few years.No one on the Internet is immune from security threats. In the race to develop online services, web applications have been developed and deployed with least attention given to security risks, resulting in a surprising number of corporate sites that are vulnerable to attackers. Noticeable sites from a number of regulated industries including financial services, government, healthcare and retail are probed daily. Web applications are used to perform key tasks or website functions. They include forms that collect personal, classified and confidential information such as medical history, credit and bank account information as well as user satisfaction feedback. The consequences of a security breach are great such as loss of revenues, damage to credibility, legal liability and loss of customer trust.

SQL injection is a code injection technique, used to attack data-driven applications, in which immoral SQL statements are inserted into an entry field for execution.SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is typically known as an attack vector for websites but can be used to attack any type of SQL database.

Cross Site Scripting attack uses the tags such as script tags to confuse appearance of HTML format of web application. Absence of input validation leads to such attacks. XSS is an attack in which front-end of the website is the

DOI: 10.15680/IJIRCCE.2017. 0506121

starting point of the attack. If an XSS susceptibility is existing on a website, then an attacker can present the code that executes when other or new users open the same website. It gives rise to users to interact with the malicious entity created by the attacker [1, 3].

II. Related work

Piyush A. Sonewar and Nalini A. Mhetre proposed a model which contains SQL Injection attack and Cross Site Scripting attack. In SQL Injection, attacker achieves security vulnerabilities of the web application to alter the valid SQL query designed by the programmer. The effect of this attack differs according to the SQL queries being injected. XSS is a web application attack where attacker crafts a 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. They have shown mapping model in which requests are mapped on queries which is helpful in detecting and preventing attack [1].

Y. J. Park and J. C. Park have showed Web Application Intrusion Detection System (WAIDS). This intrusion detection system is built on an Anomaly Intrusion Detection model for spotting input validation attacks in contrast to web applications. After detection of attack, type of attack is determined by the system in detail. This system can detect unknown abnormal web request and reduce false positives rate using extended global sequence alignment algorithm. This system also uses extended alignment algorithm to build a profile of the normal web request. And it can detect abnormal requests at runtime also. This technique is a combination of four steps likedata collection, keyword extraction, similarity measurement, filter and report.This system can prevent Input Validation Attacks during whole service time [2].

Meixing Le and Brent ByungHoon Kang have described a Double Guard, an IDS system that shows the network behavior of user sessions across both the front-end web server and the back-end database. DoubleGuard forms a container-based IDS with multiple input streams to produce alerts. This paper have showed 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. By observing both web and subsequent database, requests are able to ferret out attacks that independent IDS would not be able to identify. They have used multi-tier system. Furthermore, paper have showed the limitations of any multi-tier IDS in terms of training sessions and functionality coverage [3].

V. K. Malviya and S. Sauravshowed that Cross Site Scripting is the major threat for web application as it is the most basic attack on web application. Cross Site Request Forgery and Session Hijacking are affected by XSS. They have also given approaches for XSS prevention, XSS avoidance, XSS detection and removal. Their aim is to study and join the understanding of XSS and their origin, appearance, kinds of hazards and lessening efforts for XSS.The types of XSS attacks are non-persistent (reflected) XSS, persistent (stored) XSS and DOM-based vulnerabilities. There is one more type that is not as common as those three types, induced XSS. XSS in any form is dangerous for web applications. XSS are basically script injection attacks. This script can be written in any scripting language such as JavaScript, ActionScript and VBScript. Improper input handling is the reason behind the XSS attacks [4].

A. Kiezun and M. D. Ernst have presented a technique for finding SQL Injection and Cross Site Scripting attacks in web applications. It helps in creating inputs that uncover SQLI and XSS vulnerabilities. This technique is based on input generation, dynamic taint propagation and input mutation to find a variant of the input that exposes a vulnerability.This paper presents an automatic technique for creating inputs that expose SQLI and XSS vulnerabilities. The technique generates sample inputs, symbolically tracks taints through execution including through database accesses and mutates the inputs to produce concrete exploits. The automated tool, Ardilla, implements the technique for PHP.Using a novel concrete plus symbolic database to store taint, Ardilla can effectively and accurately find the most damaging type of input-based web application attack i.e. stored XSS. A novel attack checker that compares the output from running on an innocuous input and on a candidate attack vector allows Ardilla to detect vulnerabilities with high accuracy [5].

A. M. Chandrasekhar and K. Raghuveerhave combined techniques like k-means, fuzzy neural network and Support Vector Machine classifier. This combined technique not only makes IDS more effective but also produces a less complex system with better results. Intrusion detection is a significant 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 evidences. The aim of k-means cluster module is to partition a given set of data into clusters, where data belonging to different clusters should be as different as possible. Neural networks are significant tool for classification.The ability of high tolerance for learning-by-example makes neural networks flexible and powerful in IDS. Support Vector Machine classifier is used because it produces better results for binary classifiers as compared to other classifiers [6].

III. PROPOSED SYSTEM

The system shows strong data detection and protection of web applications while at the same time it minimizes the false positive rate. The objective is to secure three-tier web applications for detecting and preventing different types of attacks, to detect the tampering attack for database activity and to provide both front end and back end security. Many Systems are providing one way security for the web applications protecting a web application in terms of interface and at back end. The proposed system designs idea in breakdown model to evaluate security of the web applications along with its database in every step. The proposed system assumes that both web server and database server are vulnerable to various attacks such as Cross Site Scripting attack and SQL Injection attack. Cross Site scripting attack generally attacks on HTML of the web page being loaded. Attacks such as SQL Injection do not require conceding the web server. 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. Thus through this concept, it is possible to detect intrusions and create a secure network. After successful detection of the attack leads to detection of tamper and perform restoration of the original data
[2,4].

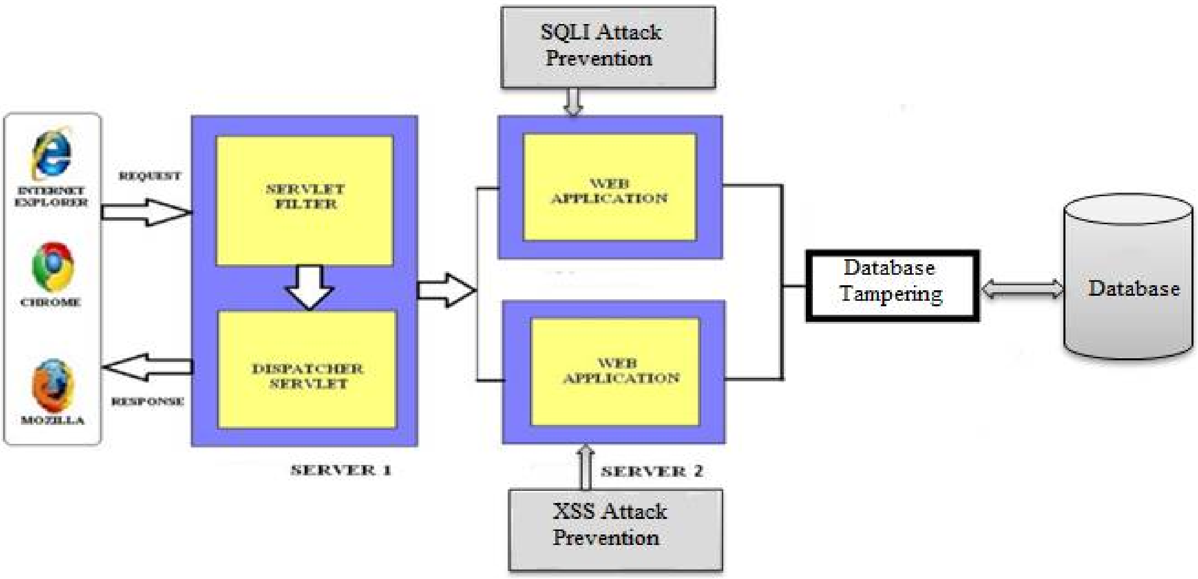


Fig. 1: Advanced Detecting and Preventing Intrusions in Multi-Tier System

Fig. 1 shows that request can come from the browsers like Internet Explorer, Google Chrome, Mozilla Firefox. This request goes to SERVER 1 i.e. Apache Tomcat server or Glassfish server. Users can choose one server between these two. Servlet Filter is a pluggable and reusable component which intercepts and mediates all relevant requests and responses. Dispatcher Servlet is the servlet that accepts the incoming request then it gives processing of that requests to the handlers. Then the request goes to SERVER 2 i.e. Database server. Database servers contains web applications. The system detects tamper on database. Tamper also prevents attacks such as Cross Site Scripting and SQL Injection with the help of restoration of the original value or initial value in the database.

1. Cross Site Scripting Attack

Cross Site Scripting attack uses the tags such as script tags to confuse appearance of HTML format of web application. Absence of input validation leads to such attacks. XSS is an attack in which front end of the website is the starting point of the attack for other or new users visiting the website. If an XSS susceptibility is existing on a website, then an attacker can present the code that executes when other or new users open the same website. It gives rise to users to interact with the malicious entity created by the attacker. Once a connection has been established, typically via social-engineering strategies, it persuades a user to do something they should not do. The attacker is able to break into the websites of visitor’s computers. Attacker provide links to users that look like honest but it has malicious script code. When user clicks on such link then user’s information will be available to the attacker.

2. SQL Injection Attack

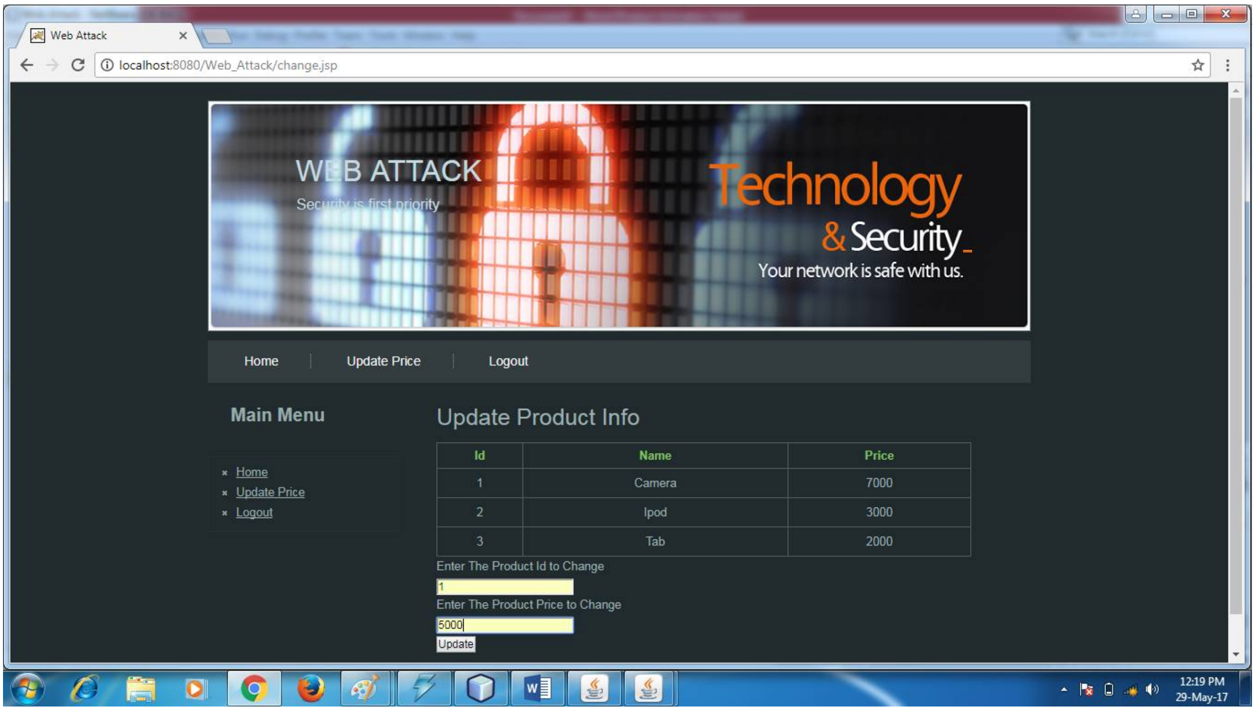
In a SQL Injection attack, attacker is well-known about SQL syntax. He submits fake entries in webpage forms with the purpose of gaining more direct access to the back end database. Input can be given in such a way that SQL queries can bypass the web server and affect the database. These attacks attempt to gain useful information like username and password combinations or subtle business data. SQL Injection attack attempts to change the databases via altered input strings provided to the web applications. SQL Injection attack may also take efforts to change data such as prize of product or to delete original data. Due to the expected nature of these types of applications, an attacker can craft a string using exact Structured Query Language (SQL) commands, and at the same time he knows that it can be used to force the database to show goods results in which he is interested. These strings can be entered in places like search boxes, login forms and even directly into a Uniform Resource Locators (URLs) to contradict simple clientside security measures on the page itself. A successful SQL Injection can read sensitive data, modify data also executes administrative operations. In this ways SQL injection attack may effect in major breaches of cyber security.

IV. ALGORITHM

Enhanced XSS Guard algorithm splits the existing web sites into three types i.e. black list, white list, grey list. When a user visits web site, this algorithm scans the page source of that web site and identifies the script presents in that web site. It checks whether these scripts matches with the black list stored in database. If it is found to match with the black list scripts, then the user is warned about the situations. Similarly, when a user visits a web site, the Enhanced XSS guard scans the page source of that web site and identifies the scripts present in that site and checks whether these scripts matches with the white list scripts stored in the database. If it is found to match with the white list scripts, then the page is sent to the respective domain. Now, for third type, when a user visits a web site, the Enhanced XSS Guard scans the page source of that web site and identifies the scripts present in that site and checks whether these scripts matches with both the white list and black list scripts stored in the database. If it is found to match equally with both the scripts, then the web page is called as a grey site and it can be tested far along for umpiring its kind [12].

MD5 algorithm is Message Digest algorithm. It is used as database intrusion detection algorithm. It takes input as a message of arbitrary length and produces output in the form of 128-bit. It splits the input in blocks of 512-bit each. This algorithm consists of phases like padding phase and compression phase. In padding phase, some (1-512 bits) additional bits are added to the input message. Then the length of the initial message is transformed to 64-bit binary string. In compression phase, a compression function is used on each 512-bit block and generates 128-bit output.

V. SIMULATION RESULTS (SCREENSHOTS)



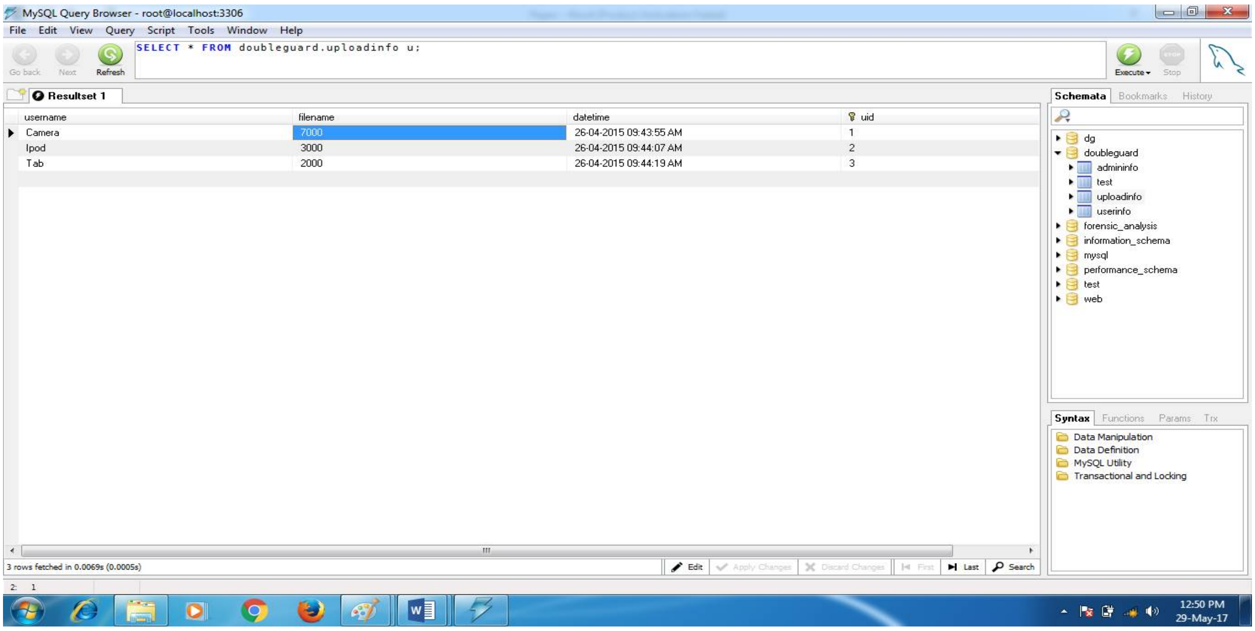
Update Product Info

Id	Name	Price
1	Camera	7000
2	Ipod	3000
3	Tab	2000

Enter The Product Id to Change: 1

Enter The Product Price to Change: 5000

Fig. 2 : Update Product Price from Front-End



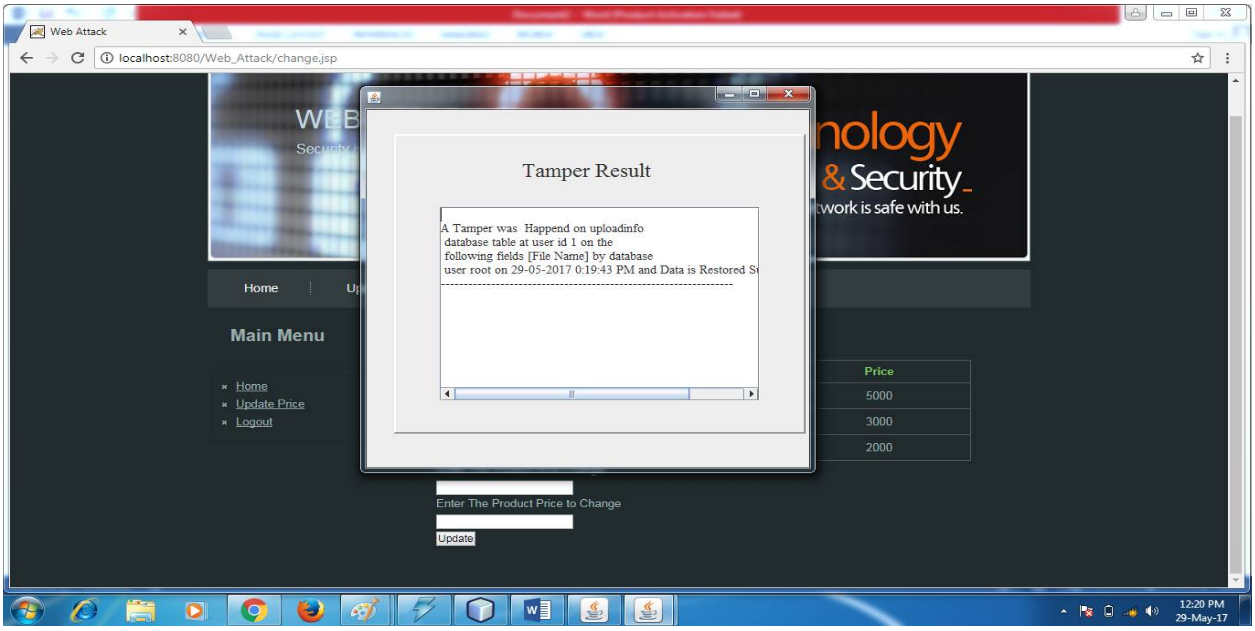
```
SELECT * FROM doubleguard.uploadinfo u;
```

username	filename	datetime	uid
Camera	7000	26-04-2015 09:43:55 AM	1
Ipod	3000	26-04-2015 09:44:07 AM	2
Tab	2000	26-04-2015 09:44:19 AM	3

3 rows fetched in 0.0069s (0.0005s)

2: 1

Fig. 3 : Update Product Price from Back-End



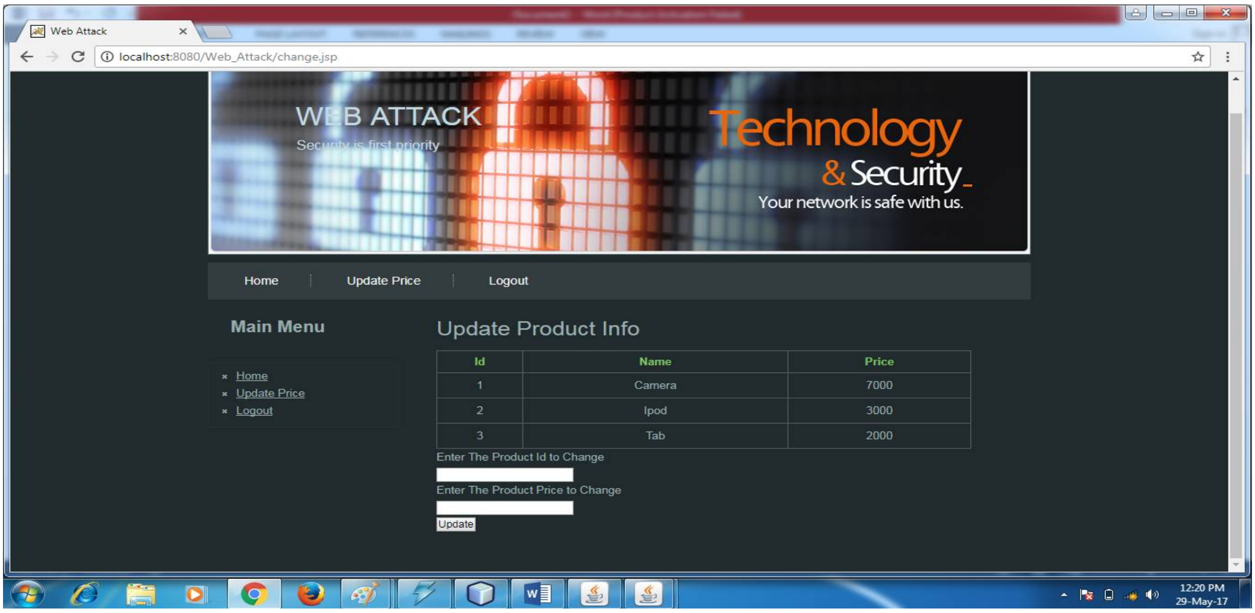
Tamper Result

A Tamper was Happend on uploadinfo
database table at user id 1 on the
following fields [File Name] by database
user root on 29-05-2017 0:19:43 PM and Data is Restored S

Price
5000
3000
2000

Enter The Product Price to Change

Fig. 4 : Tamper Detection



Update Product Info

Id	Name	Price
1	Camera	7000
2	Ipod	3000
3	Tab	2000

Enter The Product Id to Change

Enter The Product Price to Change

Fig. 5: Restoration of Original Price Successfully

VI. PERFORMANCE ANALYSIS

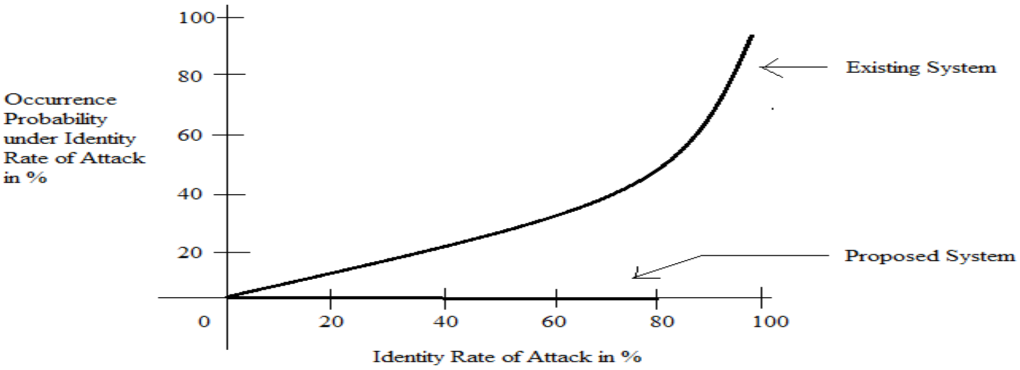


Fig. 6: Graphical Representation of Comparison of Existing System and Proposed System

The Fig. 6 shows comparison of existing system and proposed system. In existing system, after attack is happened, admin blocks that particular user. In proposed system, system detects attack before it happens and also prevent attack with the help of restoration of original values in the database as well as front-end. Tamper is detecting on database. Tamper means log maintenance. It shows which data is affected at what time. It also shows that original values are restored successfully.

VII. CONCLUSION AND FUTURE WORK

We have recognized threats of SQL Injection and Cross Site Scripting attacks using Intrusion Detection System. We detects and prevents these attacks. Earlier approaches deals withalert generation when attack happens but this system detects tamper. This system also prevents attack by successfully restoring correct value after attack detected. This system provides both front-end and back-end security. In future, detection and prevention of SQL Injection and Cross Site Scripting attacks can be installed on wide range of machines having different operating system and platforms. The query processing mechanism can be made simpler by applying Natural Language Processing (NLP) so as to convert simple English sentences into SQL 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, Jan 04-06, 2013.
[7] 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.
[8] Priyadarshini R., Jagadiswaree D, Fareedha. A, Janarthanan M,'A Cross Platform Intrusion Detection System using Inter Server Communication Techniqu', IEEE-International Conference on Recent Trends in Information Technology, ICRTIT IEEE MIT, Anna University, Chennai. June 3-5, 2011.
[9] 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.

[10] T. V. Narayan Rao, V. Tejaswini, K. Preethi,‘Defending Against Web Vulnerabilities and Cross Site Scripting”, JGRCS, Volume. 3, No.5, May 2012.

[11] LwinKhinShar, HeeBengKuanTan,‘Automated removal of cross site scripting vulnerabilities in web applications’, Information and Software Technology 54, 467478, 2012.

[12] M. James Stephen, P.V.G.D. Prasad Reddy,’Prevention of Cross Site Scripting with E-Guard Algorithm’, International Journal of Computer Applications (0975 8887) Volume 22, No.5, May 2011.

BIOGRAPHY

Miss. Seema V. Gajare
I have completed B.E. (Information Technology) and now pursuing M.E. (Computer) from SavitribaiPhule Pune University. My area of interests are Cloud Computing, Database and Data Mining.

Miss. Baban H. Thombre
He is working as Assistant Professor in Computer Department at Shree Ramchandra College of Engineering (SRCOE), Pune. He obtained B.E. from SavitribaiPhule Pune Universityand M.Tech. from Jawaharlal Nehru Technological University, Hyderabad. His area of interests are Computer Networking and Cloud Computing.

DOI: 10.15680/IJIRCCE.2017. 0506121