Database And File Intrusion Detection System

Mihir S. Thuse, Sayali P. Joshi, Pranav S. Paranjpe

Abstract— The need for secure storage of data has become a necessity of our time. Medical records, financial records and legal information are all in need of secure storage. In the era of globalization and dynamic world economies, data outsourcing is inevitable. Security is ma major concern in data outsourcing environment, since data is under the custody of third party web servers. In present systems, third party can access and view data even though they are not authorized to do so, so allowing the employee of the organization to update the database. This may lead to serious data theft, tampering or data leakages causes severe business loss to data owner. In this project we have proposed a novel solution to detect database intrusion using Log Mining approach. Log files are unalterable files at runtime, automatically created by Web servers to have trace of transactions performed on any web applications. Considering purchaser database at server-side and by comparing this with the transactions traced from the log files, we can detect database tampering for any indifference found. Finally by using dynamic management view of SQL, we can find who altered what data field and when. Our project thus provides hassle-free solution for server-side database intrusion.

Index Terms— Intrusion, Log Mining, MD5, database, web application, transaction set vector, hashed file key.

1 Introduction

An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a Management Station. Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection systems can also be system-specific.

There are three main types of Intrusion Detection Systems viz Network intrusion detection system (NIDS), Host-based intrusion detection system (HIDS) and Stack-based intrusion detection system (SIDS). Our proposed system is a type of Host-based intrusion detection system. It aims to provide detection of tampering made in databases, which contain client transactions for any particular web application, on the server side by unauthorized users. The transactions made by customers through the web application are stored in a database on the server which necessarily should remain unaltered and consistent. Our system also detects intrusions made in the owner-specific files uploaded on the server. All this is done at the request of the owner, thus providing the owner more flexibility.

The project aim is to develop an Intrusion Detection System (IDS) for tampered data in web services on the server-side for - Databases using Log Mining Algorithm and owner uploaded files using MD5 Algorithm. The proposed system aims to simplify the task of intrusion detection and simultaneously provide better speed up solution to find perfect intrusions. This project implements an E-Commerce website using Java Servlet Pages. The client transactions database would be maintained using MySQL 1.2.1 beta. For maintaining the consistency of purchaser database, the project also provides a feature of implicitly mailing a forensic analysis report as per the time interval set by the owner, using the concept of triggers. In an effort to avoid further damage, it provides database restoration of purchaser.

2 Background Motivation

We referred several IEEE papers during the initial stages of our project. 'An Effective Log Mining Approach for Database Intrusion Detection'[1] published with IEEE paper standards describes that due to sudden proliferation of networked applications, database-centered applications are facing a rapidly growing number of threats. Malicious outsiders launch attacks to access or corrupt data by stealing access control credentials or exploiting application vulnerabilities. On the Server side, server admin may sabotage databases by abusing privileges. Although various intrusion prevention and detection mechanisms are employed to protect against outsider and insider attacks, they are not very effective in detecting attacks targeted at databases at the server-side. The abovementioned paper exhibits a novel scheme for identifying malicious transaction patterns to detect attacks launched either by outsiders or insiders on server database.

Taking this research into consideration we have developed our intrusion detection mechanism giving more stress to the comparison of server-side database with the application server logs, rather than identifying any kind of malicious transaction patterns. A major advantage of our approach is that it does not require any modification and enhancement to the storage system software.

Another IEEE paper titled ‘Storage Based Intrusion Detection Storage Intrusion Networks’[2] describes storage systems as the next frontier for providing protection against intrusion. Storage systems see changes to persistent data, several types of intrusions can be detected by storage systems. Intrusion detection (ID) techniques can be deployed in various storage systems. The abovementioned paper studies how intrusions can be detected at the server side and in Storage Area Network environments. It also proposes novel approaches for storage based intrusion detection and discusses how features of state-of-the-art block storage systems can be used for intrusion detection.

Keeping the basic idea of providing intrusion detection for stored files, we have provided a facility where owner can upload .txt format files, in our project. These files mainly would contain certain confidential data which needs to be preserved securely. However, hackers on the web world would want to intrude/tamper such data. Keeping this in consideration, we provide intrusion detection for the confidential data stored in these uploaded files. The main advantage of our approach is its quicker execution time and its execution only on the request of the owner, which is made possible through the usage of MD5 Algorithm. The CPU usage of our intrusion detection system is minimal when compared to other intrusion detection systems.

3 PROJECT SCOPE

The aim of our project is to develop an intrusion detection system on server side for a web application hosted on a server. Intrusions in database, which will consist of all transactions taking place in the web application, would be detected on demand by the owner with the help of application logs. The project will automate the process of forensic analysis on tampered data. The data owners and system administrators can have secured system with our model. In case of intrusion, the time, date and the intruded fields in the database can be detected with the help of the system and hence further damage can be avoided. The user credentials of the server side database can be used to detect who was the responsible for tampering.

The project would be explained through developing a locally hosted online shopping application which would also provide the owner with added facility of uploading files to the server so as to maximize usage of server space. Our project would also provide intrusion detection for the files uploaded to the server.

4 SYSTEM DESCRIPTION

The basic portioning of the system is done in two parts namely, client side and server side. Both the sides are efficiently operable on Windows OS Platform. The server side database is maintained by MySQL 5.0. It contains various tables for storing admin login credentials, purchaser information as well as sale information.

The UI of the project is majorly developed using Java Servlet Pages. The UI provides a simple and user friendly interface for any type of user. The coding of the project is done on Java Platform using various utilities provided in Java packages.

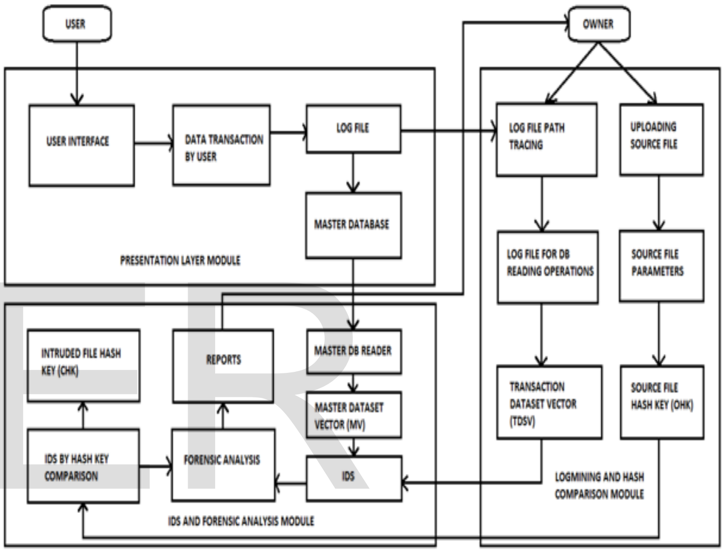
USER
OWNER
USER INTERFACE
DATA TRANSACTION BY USER
LOG FILE
MASTER DATABASE
PRESENTATION LAYER MODULE
LOG FILE PATH TRACING
UPLOADING SOURCE FILE
LOG FILE FOR DB READING OPERATIONS
SOURCE FILE PARAMETERS
TRANSACTION DATASET VECTOR (TDSV)
SOURCE FILE HASH KEY (OHK)
LOGMINING AND HASH COMPARISON MODULE
INTRUDED FILE HASH KEY (CHK)
REPORTS
MASTER DB READER
MASTER DATASET VECTOR (MV)
IDS BY HASH KEY COMPARISON
FORENSIC ANALYSIS
IDS
IDS AND FORENSIC ANALYSIS MODULE

Fig. System Architecture

Presentation Layer Module: This module basically consists of the GUI. Here the purchaser can make the purchase of the desired product, submit its information, make online payments. Each purchase made by the consumer would be stored in the form of a transaction, which would be stored in the log file and also in the master database at the server-side. On the other hand, the owner here can login and check the administrator account, upload the files onto the website.

Log-mining and Hash Comparison Module: On choosing for the option of database intrusion detection, by the owner, necessary transactions made by purchasers would be traced and copied into a Transaction DataSet Vector (TDSV). Similarly on opting for file intrusion detection, by the owner, the source file parameters would be obtained and corresponding source file hash key (OHK) would be generated.

IDS and Forensic Analysis Module: For database intrusion detection, comparison of MDSV obtained from master database and TDSV would be done and sent for further forensic analysis. Similarly for file intrusion detection, Source File Hash Key and Intruded File Hash Key would be compared, checked for intrusion and would be sent for final report. Final reports of detected intrusions would be mailed to owner. The web-application user (purchaser) has following facilities-

1. The user does not require any authentication to access the web application.
2. The end user can go through the web application and select any item of his choice to purchase.
3. While entering the information for item purchase, the data to be entered by the user should be valid and in the proper format as expected by the respective data field.

The application owner has following facilities-

1. The application will be used on computer, based on Windows Operating System.
2. Authentication would be done each time the owner would want to login.
3. The owner will have freedom to check for intrusion detection as per his/her convenience.
4. In case of intrusion, the owner can restore the original data of the database from the log file entries.
5. The system would be mailing the report of the detected intrusion to the owner.

5 Theoretical Framework

The basic idea of the project lies in mining the log files generated by the application server. The application server such as – Tomcat- contains a logs folder where all the log files generated during application processing reside. These log files are dynamic in nature as well as they can’t be edited. The last log file of this folder contains detailed application logs along with exceptions that occur during the run time of a particular application. Further processing with the help of these log files needs these log files to be copied to a text file, so that they can be converted to a static and editable format. Whenever the owner of the application wishes to opt for intrusion detection the database will be checked for any tampering and if found any indifference, the detailed report will be mailed to the owner. The report will contain the information about what data have been tampered, when it had been tampered and who was the server administrator during that period.

In addition to this, after sending the report to the owner, the intruded fields of the database will also be restored with their original values so as to avoid further damage of the owner. The application owner will also be provided with a facility to upload any text file on the server. The intrusion done on these owner uploaded files will also be detected once the owner opts for the file intrusion detection choice. For this procedure, we are making use of MD5 algorithm which allows us to generate 128 bit hexadecimal message digest of the file content. The server side database will have a table for storing the information related to the files stored on the server. The proposed system is feasible for any E-Commerce and E-banking application.

6 Algorithms For Major Modules

6.1 For database intrusion detection:

```
a. Start
b. Get path of the application server log file
c. Read the contents of the log file
d. Copy the contents to a new text file with a specified path
e. Read the content and store in a string vector
f. Get the string object
g. Get the transaction trace in log object and put it in TDSV
h. Store the database content in master object vector i.e MV
i. Get TDSV size as n
j. For i=0 to n-1
k. Search for TDSV(i) in MV with tid
l. If transaction id (tid) found then
m. Compare TDSV(i) with MV object
n. If equal then no intrusion
o. Else alert intrusion and go for forensic analysis
p. Fire a query using DMV to get when(date and time) the intrusion was done and who(by the user credentials) did the intrusion.
q. Compare TDSV(i) and MV to get which field has been tampered
r. Prepare a report in text file and mail to the owner
s. End
```

6.2 For file intrusion detection:

```
a. Start
b. Get file name, path, modified date and time (DT1) while uploading the fie
c. Apply MD5 and get hash key (HK1)
d. Store step b and c parameters in database
e. While opting for file intrusion detection, get current modified date and time(DT2)
f. If (DT1=DT2)
g. No intrusion
h. Else apply MD5 algorithm to the file and get current hash key(HK2)
i. If( HK1=HK2) then conclude that file intruded but not modified
j. Else file has been modified
k. Generate a report in text file and mail it to the owner
l. End
```

7 SYSTEM FEATURES

1. The owner has to login to check intrusion and to avail other facilities.
2. The intrusion time along with the intruded fields will be detected.
3. After the intrusion detection the intruded fields will be restored with the original values.
4. Intrusion detection is on owner’s demand.
5. In case the owner forgets to check the intrusion, a trigger will assure to check the intrusion after a specified time interval.
6. Intrusion in uploaded files will also be reported.
7. After every intrusion detection, the report will be mailed to the owner.
8. After every successful purchase through website, the purchaser will be mailed the entire report of his purchase.
9. The process of intrusion detection becomes easier and faster with the help of proposed system.

CONCLUSION

In this paper, an effective approach, based on the application server logs, has been introduced which simplifies the task the tamper detection on the server database. Also the paper provides the solution for the file intrusion detection with the help of MD5 algorithm. Our experiment showed that the proposed method can achieve the desired positive results considering the assumptions and constraints of it. As a part of future work, we can study to provide more protection to the fields which are frequently being intruded. Some special mechanism can be developed to secure these fields from being intruded. Also we can work to include the other file formats such as .pdf, .ppt, .doc to be given protection by the system.

ACKNOWLEDGMENT

We take this opportunity to thank our Prof. R. B. Joshi HOD, Computer Engineering Department and Prof. Mrs. Harmeet Kaur Khanuja (Project Guide) for their extensive support towards this project and inspiring us and helping us in all possible ways at any time.

REFERENCES

[1] An Effective Log Mining Approach for Database Intrusion Detection, Yi Ru, Alina Campan, James Walden, Irina Vorobyeva, Justin Shelton. Computer Science Department, Northern Kentucky University

[2] Storage-Based Intrusion Detection for Storage Area Networks (SANs), Mohammad Banikazemi Dan Poff Bulent Abali. Thomas J. Watson Research Center, IBM Research, Yorktown Heights.

[3] Hu, Y., and Panda, B.: A Data Mining Approach for Database Intrusion Detection, In Proceedings of the 19th ACM Symposium on Applied Computing, Nicosia, Cyprus, 2008