Pak. J. Biotechnol. Vol. 13 special issue II (International Conference on Engineering and Technology Systems (ICET'16) Pp. 130 - 134 (2016)

DATABASE SECURITY INCURSION RECOGNITION TECHNIQUE USING NEURAL NETWORK

D.Saravanan

Faculty of operations & IT, IFHE University, IBS Hyderabad

ABSTRACT

Database Intrusion Detection System (IDS) is an expert system looking for evidence of attacks on known vulnerabilities of the system. It holds a statistical model of the behaviour of a user on a system under surveillance. There are several techniques, protocols, and algorithms to increase the security level of database. In such works, there is a lack of time complexity analysis of the techniques. This time complexity has occurred due to the comparison process carried out at each time the user query is given i.e., comparing the profiles of online transactions and the stored authorized transactions each time when the query is received. This system time complexity also affects the system performance in terms of their precise security. It learns the habits a user working with the computer and to raise warnings when the current behaviour is not consistent with the previous learnt patterns, thus detecting whether the user is authentic or not. The system can be implemented using MATLAB. MATLAB is a numerical computing environment. It allows matrix manipulations, plotting of functions and data implementation of data. The learning process by neural network avoids the unauthorized transactions in the DBMS and reduces the time complexity the project improves the performance of the database system. The neural network implementation will show the effectiveness of the proposed IDS technique in securing the database from the intruders. The performance of the proposed technique is evaluated by utilizing different statistical performance measures.

Index Terms – Security systems, Data base security, Database applications, Overloads, Database Attacks.

I. INTRODUCTION

Database is an organized collection of data. The data is typically organized to model relevant aspects of reality in a way that supports processes requiring this information (for example, finding a hotel with vacancies). A general-purpose database management system (DBMS) is a software system designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, Postgre SQL, SQLite, Microsoft SQL Server, Microsoft Access, Oracle, Sybase, dBase, FoxPro and IBM DB2. The database is not generally portable across different DBMS, but different DBMSs can inter-operate by using standard such as SQLand ODBC or JDBC to allow a single application to work with more than one database. Database security concerns the use of a broad range of information security controls to protect databases (potentially including the data, the database applications or stored functions, the database systems, the database servers and the associated network links) against compromises of their confidentiality, integrity and availability. Security risks to database systems include:

- Unauthorized or unintended activity or misuse by authorized database users, database administrators, or network/systems managers, or by unauthorized users or hackers (e.g. inappropriate access to sensitive data, metadata or functions within databases, or inappropriate changes to the database programs, structures or security configu rations)
- Malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services;
- Overloads, performance constraints and capacity issues resulting in the inability of authorized users to use databases as intended;
- Physical damage to database servers caused by computer room fires or floods, overheating, lightning, accidental liquid spills, static discharge, electronic breakdowns/equipment failures and obsolescence.

II. EXISTING SYSTEM

Detection System for Security in Database using COUNTER BLOOM FILTER (CBF). This approach is considered as transaction level approach and is used to detect the malicious transactions in the database. The protection of the database by the use of encryption techniques, where the database may be encrypted but this kind of system may lead to query degradation. The overall system architecture is divided into two parts

- Learning phase.
- Intrusion detection phase

The learning phase generates authorized transactions profile automatically and is used at detection phase to check the behavior of executable transactions. In a counter bloom filter it generalizes a bloom filter data structure by allowing the membership query and the CBF can be changed dramatically by insertions and deletion operations.

A. Problem in Existing system

- This technique fails against unknown database attacks.
- This approach is not effective because manual generating transaction profiles mechanism is more time consuming process maybe undetected.
- It does not detect transaction level dependency; hence some of them attack the database.
- It does not provide time efficiency. It is so because each time a transaction takes place the

pattern needs to be learnt. Previously occurred patterns need be learnt thereby wasting time during the leaning and detection for intrusion or not.

III. PROPOSED SYSTEM

The objective of the project is to avoid the unauthorized transactions in the DBMSusing NEURAL NETWORK. The proposed database security system includes Authorized Profile Creation, Authorized Profile Learning and Security Checking. The proposed system works on the idea that intruders work of given set of pattern. Pattern learning is a part of the system. Learning the process leads to less time consumption, this helps to improve the actions to be carried out.

Advantage:

- Improves the performance of the system.
- Reduces processing time.

It is based on the fact that most intruders act in a determined pattern. The pattern is stored in the database and is used dynamically when the intrusion is followed. Since the pattern is saved it is less time consuming as it is saved previously. Performance is related to the work done as well as the time consumed. If a system consumes less time, it performance is improved.

IV. LITERATURE SURVEY

Chung et al. [1] presents a misuse detection system called DEMIDS which is personalized to relational database systems. This approach uses audit logs to derive profiles that describe typical behavior of the users working with the DBS. The profiles computed can be used to detect misuse behavior in particular insides abuse. Lee et al. [3] have proposed a real-time database intrusion detection using time signatures. Real-time database systems have a deal with data that changes its value with time. This intrusion detection model observes the database behavior at the level of sensor transaction. If a transaction attempts to update a temporal data which has already been updated in that period, an alarm is raised. UdaiPratapRao and Dhiren R. Patel [5] propose database intrusion detection mechanism to enhance the security of DBMS. In a typical database environment, it is possible to define the profile of transactions that each user is allowed to execute. In their approach, they use the transactions profile and overall system architecture is divided into two parts, learning phase and intrusion detection phase. The learning phase generates authorized transactions profile automatically and is used at detection phase to check the behaviour of executable transactions. There is also implementation of the detection phase with the help of Counting Bloom Filter (CBF) and comparing both the approaches. Wenhui et al., [6] proposed a two-layer mechanism to detect intrusions against a web-based database services. However, they have not used different level of granularity or intra-transactional and inter-transactional features in their model. Hu et al., [7] determine the dependency among data items where data dependency refers to the access correlations among data items. These data dependency are generated in the form of association rules.

V. EXPERIMENTAL SETUP

When the project is made to run it will first display a login authentication page which includes login and sign up option, so if user had already registered then user will go for login option or else user will select sign up option for registration. After the registration process a particular user have to login by giving input as username, password. The user enters the information according to the fields' provided. This is the information saved to the database. This stored data is on which the intrusion is detected. Intrusion i.e. any unwanted trying to access information that is not within the limits.The student has the least access as they can access only the student information. Whereas the teacher can access the teacher's as well as the students information. It will be able to make changes to the student information. The admin has the most mobility accessing all the pages i.e. student , teacher as well as admin. The intrusion needs to be detected if any access which is not authorized is trying to do so. This is done by learning all patterns done by the user.

A. Users Registration

The login window is where the entire new user registers. The registration helps to create account which would give access to the data base. The users have to user data about themselves according to the "register as" chosen by them. The use rid is unique for all the users. This gives security and even prevents from wrong access. The home page is as one registers as. There is separate home page for a student. The student has to fill the information after login. The various buttons like delete, insert, and update are to be used for any data manipulation.

For transaction purpose, all the home page for all register has been given some transactions buttons. The students are provided with

- Select: allows the user to retrieve data which has already saved within the database during a previous transaction.
- Update: allows the user to change or update the user information that is already present in the database. This button is used to keep the database up to date on the user's most current details.
- Delete: is used to remove any detail from any of the field. It allows the user to remove nay data which the user has found redundant or is no longer applicable to the user.
- Insert: is the most important button within the student page. Whenever a new student logs in it allows that user to input or insert his or her details within the database.

The teacher is provided with a different home page. The transactions done by the teacher are

- Select: allows the user to retrieve data which has already saved within the database during a previous transaction.
- Update: allows the user to change or update the user information that is already present in the

database. This button is used to keep the database up to date on the user's most current details.

- Delete: is used to remove any detail from any of the field. It allows the user to remove nay data which the user has found redundant or is no longer applicable to the user.
- Insert: is the most important button within the student page. Whenever a new student logs in it allows that user to input or insert his or her details within the database.

This page differs from the student page as this page only available to the teacher the teacher to access any student profile page i.e. it gives the access to perform any operation on the student details on any student in the database and it has special function button known as "Student". The "Student" button allows lastly any user can register as admin. The admin can also perform various transactions.

- Select: allows the user to retrieve data which has already saved within the database during a previous transaction.
- Update: allows the user to change or update the user information that is already present in the database. This button is used to keep the data base up to date on the user's most current details
- Delete: is used to remove any detail from any of the field. It allows the user to remove nay data which the user has found redundant or is no longer applicable to the user.
- Insert: is the most important button within the student page. Whenever a new student logs in it allows that user to input or insert his or her details within the database.

This page like the teacher profile page also allows the admin to access the student profile through the "student "button and perform various transactions in the student page. It has complete access in the page. The addition feature present in this page is given by the "teacher" button. The "teacher" button allows the admin to gain super user access to the teacher profile and allows the admin to perform various transactions in the teacher's profile as well.

B. Transaction

The data manipulation refers to all the actions taken by each user. The user maybe logged as admin, teacher or student. These include

- Select: allows the user to retrieve data which has already saved within the database during a previous transaction.
- Update: allows the user to change or update the user information that is already present in the database. This button is used to keep the data base up to date on the user's most current details.
- Delete: button is used to remove any detail from any of the field. It allows the user to remove nay data which the user has found redundant or is no longer applicable to the user.
- Insert: button is the most important button within the student page. Whenever a new student logs in it allows that user to input or insert his or her details within the database.

The teacher page has an extra functionality.

- Student: allows the teacher to access any student profile page i.e. it gives the access to perform any operation on the student details on any student in the database.

The admin is provided with student as well as teacher functions.

- Student: allows the teacher to access any student profile page i.e. it gives the access to perform any operation on the student details on any student in the database.
- Teacher: allows the admin to access any teacher profile page i.e. it gives the access to perform any operation on the teacher details on any teacher in the database.

The above when done the information (data) is stored. It is saved for further reference. The data is stored in a database. The transaction is stored in individual tables for the student, teacher or admin. Each table is accessed using the sql query codes. Since everything is saved in the database the sql commands need to be put to get result for any transactions.

C. Audit Log Table

Audit log or audit trail is security-relevant chronological record, set of records. It provides documentary evidence of the sequence of activities that have affected at any time a specific operation, procedure, or event. The action performed by all users is saved in a table. The audit log table includes the session id, sequence no., command, operator, and user id and operation time. The activities are stored according to sessions. Thus, the movement of any user throughout the process is saved in order of transverse.

Session id refers to the time when a user logs on to the system. A session continues till all transactions are complete. Session id serves as a record for all the transactions that a user does till the user disconnects with the system.

- Sequence no is used to check the flow of any transaction. The transaction of any number is recorded. A user may have more than one sequence no. if it does have more transactions. The sequence no. is auto incremented. A sequence no. cannot be same as the session changes so do the sequence.
- Command refers to buttons through which any transactions takes place. The actions displayed in the database. It keeps track of the events.
- Operation time displays the time take to complete any transaction. The user may be accessing it for a longer duration of time.
- User id is the name by which any new user registers. It is through which all users have a different identity. It maintains the uniqueness. All transaction can be checked for the user name.

D. Profile Creation

The profile for each user is created after the audit log table is created. The profile is created by the fields of the audit log table namely session id, sequence no, username, transaction is, and operator. Based on the user's session id, the authorized profile transactions are extracted and created for each user. For any operations to be performed it is done with the connection to the database. Based on the user's session id, the authorized user's transaction is extracted and created the authorized profile for each user. The profile is unique for any user, thus separating the activities of one user from the other.

Generation of Profile Creation

- Input: Audit Log Table.
- Output: Authorized Profile Creation.
- Sort the session id and session name.
- Session id must be unique.
- Extract for each session id from audit log table.
- Save it another table.
- Profile created.

VI. EXPERIMENTAL OUTCOME

Successful login is when for any user id the password entered is same as it was entered at the time of registering. If the password matches the user id it logs in the system this is the basic level of protection given to prevent any intruder from accusing. No field should be empty. Empty field returns a message "Invalid password". It is the basic security provided to any user within the system. Even if the user mane may be achieved one still has to input the valid password for access. Any field in the home window is not being kept empty. If a user wants to commit transactions no field in the window should not be left unfilled. If any field is left unfilled the transaction is not complete and an error message is shown. "Can't be empty". This feature is implemented so as to keep the data integrity of all the details in the database.

Test Case Id: 01, Test Case: Correct password

Input: From the database, Expected Output: Password accepted, Actual Output : Password accepted. Login successful. Refer fig 1

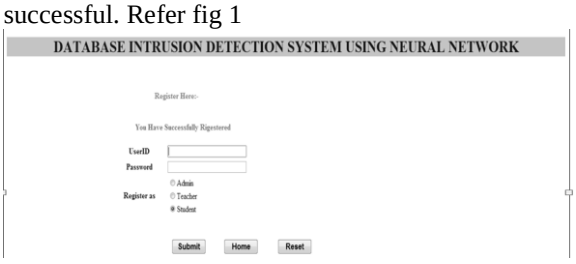

Fig. 1 Password accepted

Test Case Id: 02 Test Case: Incorrect password entry Input any type of symbols and values Expected Output: Password rejected. Actual Output : Password rejected. Refer fig 2

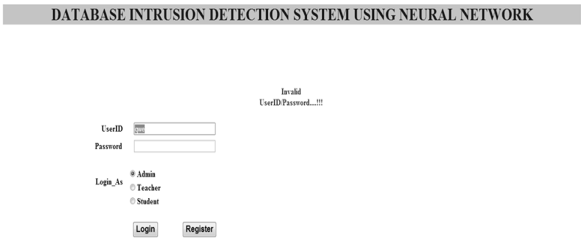

Fig.2 Password rejected

Test Case Id 3, Test Case: Null input field, Input: Blank input field ,Expected Output: Input rejected, Actual Output: Input rejected. Refer Fig 3

Fig.3 Null input

Test Case Id: 04, Test Case : Correct input field ,Input: Any type of values, Expected Out : Input accepted, Actual Output: Input accepted. Value added to database. Refer fig 4

Fig.4 input accepted

Test Case Id: 5, Test Case: Existing ID ,Input: Integer, Expected Output: Invalid input. Actual Output : ID already exists. Refer fig 5

Fig.5 Existing ID

VII. CONCLUSION AND FUTURE ENHANCEMENT

The system traces a complete transaction of any particular user from the time the user logs into the system till the exit. This complete transaction is to be

learnt by the neural network. For any user the flow is learnt. The session id is created for any user which tracks all the movement of the user. The movement is stored in the audit log table. Audit log table contains the user id, user name, command type, action performed. The learning algorithm takes input from the audit log table. Since audit log table serves as a collection of information of the entire user accessed. It makes it relatively easier for the pattern detection algorithm to understand the work of any particular user. The learning algorithm has to learn this pattern only once. It saves time as once the pattern is learnt it can be reused. The system works on the principle that user will follow a set of path continuously. There will be no deviation on any account. If this condition is done then any changes on the path will lead to error. The learnt pattern matches with any current work of the same user. It is assumed that it would follow the same path. Hence being able to detect the intrusion, if there is any.

A. Future Enhancement

The intrusion detection works if the condition that the user would follow the same path is applied. But it may also be in some case that the user may deviate from the path usually chosen .Thus it would be more efficient if slight deviation from previous path is ignored. For any particular user more than one path should be considered. There is always scope to reduce the time taken for any unauthorized transaction to be detected.

REFERENCES

[1] C. Y. 0chung, M. Gertz and K. Levitt, DEMIDS: A MisuseDetection System for Database systems, IFIP TC-11 WG 11.5 Conference on integrity and internal control in information system Pp. 159-178 (1999).

[2] Fonseca, Vieira, M. and H. Madeira, Integrated intrusion detection in database In: Bondavalli, A., Brasileiro, F, Rajsbaum S.(eds), LNCS, Springer, Heidelberg 4746: 198-211 (2007).

[3] Saravanan, D., Segment Based Indexing Technique for Data file. Procedia of computer Science 87: 12-17 (2016).

4] Saravanan, D., Video Substance Extraction Using Image uture Population based Techniques. ARPN Journal of Engg. and Applied Science 11(11): 7041-7045 (2016).

[5] Lee S.Y., Low W.L and Teoh P., DIDAFIT: Detecting Intrusions in Database Through Fingerprinting Transactions, proceedings of the 4th International Conference on Enterprise Information system(ICEIS) Pp. 121-128 (2002).

[6] Srivastava, A., Sural, S. and Majumdar, A. K., Weighted intra-transactions rule mining for database intrusion detection, Proceedings of the Pacific-Asia knowledge discovery and data mining (PAKDD), lecture notes in artificial intelligence, Springer Pp. 611-620 (2006).

[7] D. Saravanan, Effective Multimedia content Reterival. International journal of Applied Enivornmental Sci. 10 (5): 1771-17783 (2015).

[8] D. Saravanan, Various Assorted Cluster Performance xamination using Vide Data Mining Technique.Global journal of pure and applied Mathematics 11(6): 4457-4467 (2015)

[9] Udai Pratap Rao, G.J.Sahani and Dhiren R.Patel, Detection of Malicious Activity in Role Based Access Control Enabled Databases. International Journal of Information Assurance and Security 5(6): 611-617 (2010).

[10] Wenhui S. and Tan T., A novel intrusion detection system model for securing web based database systems, Proceedings of the 25th annual international computer software and application conference (COMPSAC), Pp. 249-254 (2001).

[11] Y. Hu, B. Panda, A data mining approach for database intrusion detection, Proceedings of the ACM Symposium on applied computing Pp. 711-716 (2004).