A PREVENTIVE APPROACH USING THE DATA MINING OF TRANSACTION AUDIT LOG FOR DATABASE INTRUSION DETECTION

[1] Dr. Yagnik A. Rathod

[1]Asst. Prof., Computer Department, Government Engineering College, Dahod, India

Email: {[1]rathod.yagnik@gmail.com}

Abstract

Information is a key component in today's global business environment. An organization, institute, or business firm uses various database management systems for managing its crucial information. The security mechanism provides by DBMS is not enough to prevent intruders or detect anomalous behavior. Unauthorized users and sometimes authorized users to execute malicious commands intentionally or by mistake, cannot be detected and prevented by a typical security mechanism. Intrusion detection system finds intrusive action and attempts by detecting the behavior of user's action. Security features can be enhanced by adding intrusive detection technology to the Database management system. Data mining is to identify valid, novel, potentially useful, and ultimately understandable patterns in massive data. It is required to apply data mining techniques to detect various intrusions. In this paper mechanism based on data mining is discussed to detect malicious action in DBMS.

Key Words: Data Mining, Database Security, Log mining, Intrusion detection

1. Introduction

The security mechanism for a Database is the system, processes, and procedures that protect a database from malicious attempts to steal (view) or modify data. Intruder activity can be differentiated as authenticated misuse, malicious attacks, or instinctive actions made by authorized individuals or processes.

Information is the greatest precarious resource for numerous establishments [1]. In the present global era, the growth and future of any industry depend on the accessibility of crucial data and the protection of critical data. A system administrator is responsible for applying security policies and monitoring access. Identifying malicious action or intruder is the main challenge for DBA. Security threats need much attention in the information domain due to networked data.

Security is a concept that includes the following properties [1]: authenticity (promises that a provision or portion of data is authentic), confidentiality (nonappearance of unlawful discovery of service or piece of information), integrity (protection of service or portion of data against unlawful and/or unnoticed alteration), and availability (security of provision or portion of data against conceivable denials of service instigated by malevolent activities). Identification and authentication are provided by using User ID and Password. DBA is responsible to provide Identification and authentication detail to users. Users can log in to the Database management system with the use of provided details only. DBA can maintain a log of activities performed by users for forensics. The main challenge for a database administrator is to protect data from unauthorized access or miscellaneous behavior. An authorized user can damage the integrity of data intentionally or by mistake. Password can be stolen or some users can use the masquerade technique. The features and techniques provided are not enough to prevent the system from vulnerability. Unfortunately, the threat of intrusion is very crucial and the suggestion of immediate intrusion detection apparatuses for DBMS is a rational and appropriate footstep.

Various security attacks can be differentiated as 1) any intruder who does not have access right to the database tries to access or modify crucial data. 2) Legitimate user with sufficient privileges intentionally challenges the integrity of the database (legitimate access to database servers but should not access database data) and 3) Denial of service type of attack where intruder tries to cause delays or block, in accessing services for legitimate users. We are interested in preventing malicious behavior and intrusive action by authorized and unauthorized users that exploit system vulnerability.

The database management system can avoid access to the database through identification and authentication but cannot detect malicious actions performed by intruders. In many situations, the execution of intrusive commands can be remaining unnoticed and can compromise with data. DBMS security policies are incorrectly configured and create loopholes for hackers to enter into the system.

The DBA does not spend much time on security activation. All security provisions provided by the database security mechanism are not utilized properly (like authentication, encryption, usage rights for users and enabling audit log, etc.), which permits interlopers to catch

admittance to database information. An intruder can trick the implementation of the database and find out paths to access the server and sometimes get data by performing statistics on different results and exploring those flaws. Unsanctioned consumers "still" the secluded particulars of official users in order to admittance to the database server. And more serious is authorized persons use their rights to enter into the system and execute miscellaneous Commands to access critical data.

An intruder can trick the implementation of the database and find out paths to access the server and sometimes get data by performing statistics on different results and exploring those defects. Unauthorized users "still" the private details of authorized users in order to access the database server. And more serious is authorized persons use their rights to enter into the system and execute miscellaneous Commands to access critical data.

Over the last few years, data mining has concerned a lot of consideration because of the augmented generation, communication, and storage of high-volume information and an impending requisite for mining valuable data and knowledge from them [2]. Data mining refers to a collection of methods by which large sets of stored data are filtered, transformed, and organized into meaningful information sets [3]. data mining techniques can be applied for the effectiveness of computer security and especially for database security in terms of intrusion detection.

2. Related Work

A completely computerized database intrusion detection arrangement that discourses together insider and foreigner outbreaks that can hinder openings that goes unobserved by the system or host-based intrusion detection systems are the requirement of time. A flexible system is presented in [4], which is proficient in refinement with the databases' aggregate complication and dynamic environment. A novel Data Structure called Octraplet along with Naive Bayes Classifier is used for storing the SQL queries. A method recommended by Bertino et al. [5] generates the standard outline for each role for inconsistent outline findings. Likewise, they used the Naïve-Bayes classifier [6] to discover inconsistent SQL queries. The analysis exertions are prepared to identify and prevent SQL injection outbreaks along with evaluating the effectiveness of the ModSecurity web application firewall in thwarting SQL injection outbreaks in [7]. When malevolent consumers want to acquire confidential data or carry mutilation for a web application, they can deliver inadvertent input to the application as a substitute for ordinary consumer input [8]. This inadvertent input which formulae SQL testimonials is acknowledged as the SQL Injection (SQLI) attack. The SQL injection, a known online outbreak, has been a thought-provoking network security concern that sources each year lots of money of economic damage globally as well as a huge volume of consumers' confidential information out-break. The contribution presented in [9] is a greatly accurate SQL injection recognition technique grounded on a neural network. It firstly obtains authentic handler URL access log information from the Internet Service Provider(ISP), guaranteeing that the methodology is actual, effective, and concrete. Thereafter, conducts an arithmetical investigation on usual information and SQL injection information. Founded on the arithmetical outcomes, proposes eight kinds of features, and learned an MLP prototypical. A survey effort presented through [10] provides a categorization of existing IDS, an all-inclusive analysis of distinguished current works, and a summary of the datasets usually utilized for assessment commitments. It furthermore offerings evasion procedures used by invaders to evade uncovering and argues forthcoming investigation challenges to aggressively handling such procedures so as to make computer organizations further protected. A strategy is proposed in [11], to cultivate an Intrusion Detection (ID) methodology, realized and incorporated inside the database server, that is accomplished by distinguishing uncharacteristic consumer appeals directed towards a DBMS. The significant indication here is about acquiring profiles of relevant consumers and software intermingling through a database. A database demand that diverges from these outlines is then characterized as inconsistent. A foremost element of this effort comprises the archetype realization of this ID mechanism in the PostgreSQL database server.

Intrusion is characterized as any group of actions that attempt to compromise the integrity, confidentiality, or availability of resources [11]. One model presented by Chung et al. in DEMIDS [12] references the relationship of data items in terms of their access and identifies the scope in which data items can be accessed by users and decide about the behavior of the user. Lee et al. illustrated a model [13] that describes an approach that creates a fingerprint of valid transactions and uses it for detecting miscellaneous behaviors. One approach presented by Lee et al. [14] uses time signatures for real-time databases. It allocates time stamping for data items that got accessed and uses time stamping for discovering a violation of security policies. One approach presented by Yi Hu al. [15], focuses on the relationship between data items of the database. It finds out dependencies between different read and write operation and try to analyze the update scenario That is, which data items must be read or written before a data item gets updated and which others are written after the update. By ensuring that transaction conforms to data dependencies, tries to avoid intrusion. The Hidden Markov Model has been proposed in [16] to detect malicious data corruption. In [17] a real-time intrusion detection mechanism based on the profile of user roles is proposed. An intrusion attack and isolation mechanism were proposed in [18]. This method uses triggers and transaction profiles to track the items read and written by transactions and isolates attacks by rewriting user SQL statements. Lee et al. [19] have used time signatures in discovering

database intrusions. They attach time signatures with data items. Intrusion is detected if a transaction tries to modify a data item that is already updated within the time range.

3. Proposed Approach

The proposed mechanism is based on anomaly detection and has a learning, detection, and response phase. Very briefly, the behavior of database transactions is collected as a first step to feed the learning phase. Once the database utilization behavior is established, the behavior learned from audit data is used to concurrently detect database intrusions in the detection phase. For intrusive behavior, this mechanism will alert the database administrator. The fundamental leitmotif of the methodology will be to apply data mining techniques to the collected audit data to compute models that accurately capture the actual behavior (i.e., patterns) of intrusions and normal activities. A simple groundwork for intrusion detection is accumulating numerous normal and anomalous behaviors of Database transactions in the audit log. An audit log can be generated by enabling an audit mechanism for recording system events. An audit log has a massive volume of audit data having a large number of audit records and a large number of system features (fields of the audit records), efficient and intelligent data analysis tools are required to discover the behavior of system activities. Data mining usually denotes the practice of digging out expressive representations from huge provisions of data.

A. Architecture

We have presented the proposed model diagram in Fig.1.we can describe architecture in three phases. The first phase includes a module for learning the behavior of transactions. We have an offline audit log which was collected by techniques like triggering or some audit log mechanism that can be used. By applying data mining techniques to learn the behavior of anomalous and valid transactions and using that for classification of legitimate and illegitimate transaction behavior as rule-set. During the second phase, Preprocessing module is used for extracting the necessary information from transactions users perform with the database. Generally, a consumer operation is among BEGIN and END announcements in the transaction and contains various clauses like select, insert, update, etc. and attributes of the database upon which operations are performed. In preprocessing we will extract keywords, operations, and target object attributes and kept them in the dataset. So the output of preprocessing is a dataset or transaction set which can be used for the next module. This dataset is like a record in a table containing transaction ID, operation, all user attributes in transaction and time of execution, and more fields if required and stored in a data structure like audit log records. Data collected as a result of pre-processing step is used to generate its behavior by using the same data mining technique. A detection module is needed to check generated behavior of transactions with a rule set of learned behavior from history. If the transaction is classified as an illegitimate transaction, the database administrator got an alert and decides the response in the final phase.

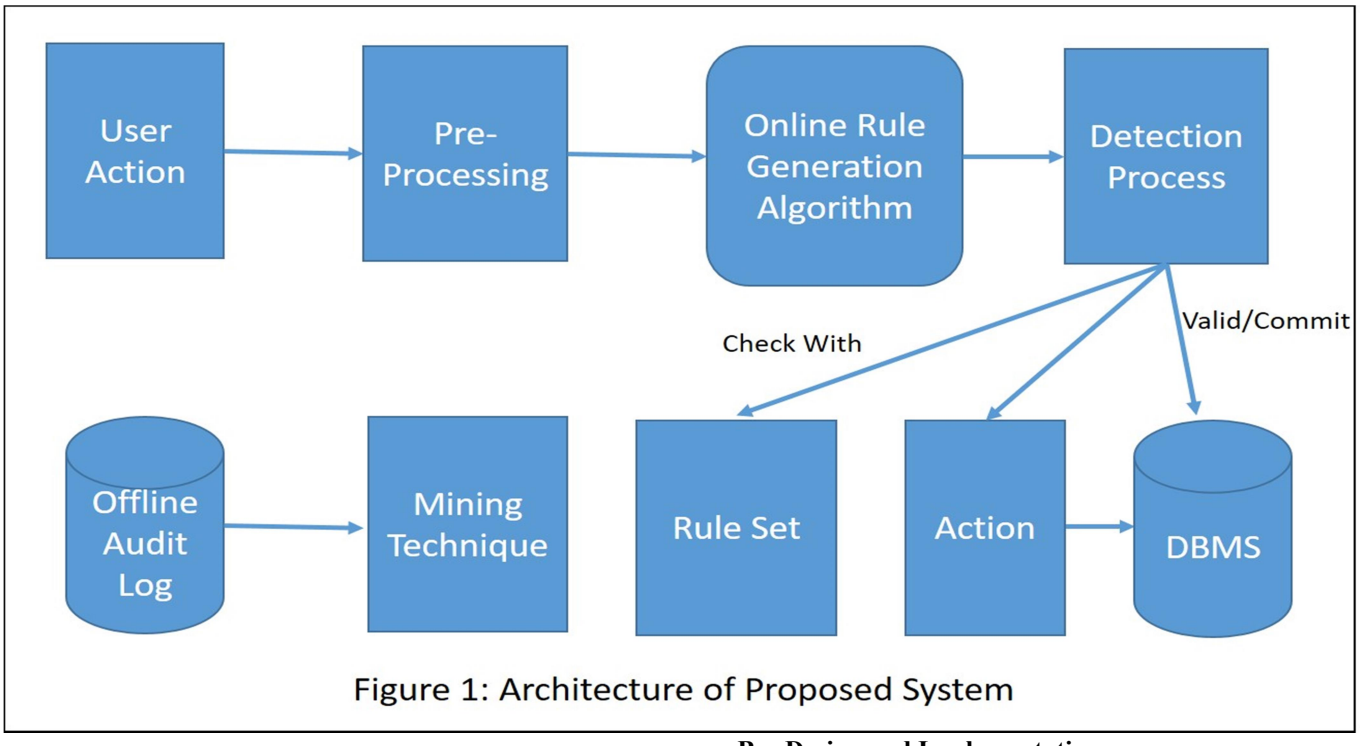

Figure 1: Architecture of Proposed System

B. Design and Implementation

We are using a small database schema represented in E-R Diagram for banking enterprise in Fig.2. Various relations like account, employer, customer, and payment and relationship like depositor, borrower, and loan-payment, etc. are identified in Fig.1 for considered schema Attributes for each relation is also mentioned. We are using notations like A1, A2, and A3 for the relation of the customer and follow this sequenced for attributes of all relations. So we will refer to all attributes with the allotted symbol only.

Select * from loan order by amount desc, loan_no asc.

The transaction is recorded in the second raw of table 1. All historical transaction identified as legitimate or illegitimate is recorded in the mentioned format. All transactions performed with the database contain SQL queries. In each SQL query, some attributes of relation are read and some attributes are updated and these details are available in the audit log. Our approach uses a data mining technique to prepare the model for the behavior of the data set. Legitimate and illegitimate classified records of audit logs are used to learn behavior. Naïve base classification is used to classify a user's transaction as legal or illegal based on the learned model. We are identifying a few logically correct queries and the same way identifying a few malicious queries also. We are using relations specified in Fig.2. These all SQL queries are performed on bank enterprise relations shown in Fig.2.We convert those queries into a specified format and use it as a dataset and using a naïve base classifier to get results with confusion matrix as below. In almost 88% of cases, we will get correctly identified class instances and we will make more effort to improve our results.

Confusion Matrix: a b classified as: 23 1 | a = L

7 8 | b = IL

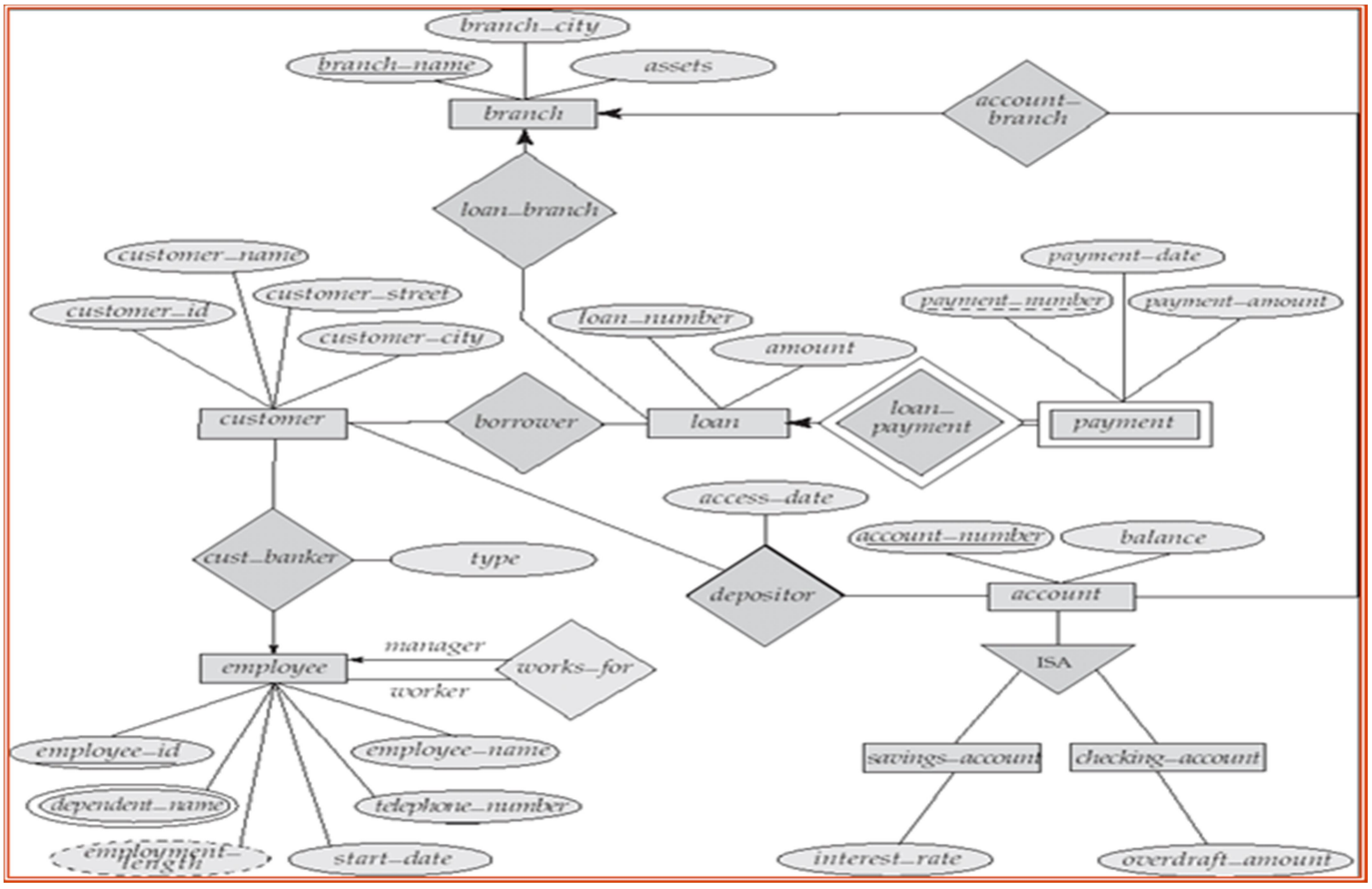

Figure 2: E-R diagram for Bank Enterprise [20]

Table 1: Transaction Audit Log

Id	A1	A2	A3	A4	A5	A6	A7	A8	A.	L/IL
1	r	r			r		w			l
2	r		w	r		r		w		l
3	w	w		w	w	w		w		IL

Correctly Classified Instances: 8 88.8889 %
Incorrectly Classified Instances: 1 11.1111 %

4. Conclusion

Several mechanisms required to protect the information, like authentication, access rights, encryption and decryption, and auditing, have been implemented in DBMS. Still many available security mechanisms are not able to identify intrusive actions. In fact, anomalous transactions executed by unauthorized users by using vulnerabilities to get access to system data and unauthorized database transactions executed by authorized users cannot be detected and stopped by typical security mechanisms. Database Intrusion Detection techniques using auditing and learning behavior of transactions have been explained in this paper and it may help to detect the malicious activities in the DBMS. We can make the response phase more accurate by taking corrective action against malicious activity.

5. References

[1] Marco Vieira Henrique Madeira, “Detection of Malicious Transactions in DBMS”, Dependable Computing, 2005. Proceedings. 11th Pacific Rim International Symposium on 12-14 Dec. 2005.

[2] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers (2001).

[3] U. Fayyad, G. P. Shapiro, P. Smyth, The KDD Process for Extracting Useful Knowledge from Volumes of Data, Communications of the ACM, pp. 27-34 (1996).

[4] S. Jayaprakash and K. Kandasamy, "Database Intrusion Detection System Using Octraplet and Machine Learning," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1413-1416, DOI: 10.1109/ICICCT.2018.8473029.

[4] Heady, R., Luger, G., Maccabe, A., and Servilla, M. The Architecture of a Network Level Intrusion Detection System. Technical report, Computer Science Department, University of New Mexico, August 1990.

[5] Bertino. E, Kamra. A, Terzi. E, Vakali. A Intrusion detection in RBAC administered databases, in Proceedings of the 21st Annual Computer Security Applications Conference, USA, December. 2005

[6] Indu Singh, L Akshaya, Kejriwal, Adithya Agarwal. Conditional adherence based classification of transactions for database intrusion detection and prevention, in International Conference on Advances in Computing, Communications, and Informatics (ICACCI). ,2016

[7] B. I. Mukhtar and M. A. Azer, "Evaluating the ModSecurity Web Application Firewall Against SQL Injection Attacks," 2020 15th International Conference on Computer Engineering and Systems (ICCES), 2020, pp. 1-6, DOI: 10.1109/ICCES51560.2020.9334626.

[8] Vemulakonda, Rajesh, and Ketha Venkatesh. "SQLIADP: A Novel Framework to Detect and Prevent SQL Injection Attacks." Smart Intelligent Computing and Applications. Springer, Singapore, 2020. pp. 41-50.

[9] P. Tang, W. Qiu, Z. Huang et al., Detection of SQL injection based on artificial neural network, Knowledge-Based Systems (2020).

[10] Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets, and challenges. Cybersecurity, 20 (2019). https://doi.org/10.1186/s42400-019-0038-7

[11] Ashish Kamra, Elisa Bertino, and Guy Lebanon. 2008. Mechanisms for database intrusion detection and response. In Proceedings of the 2nd SIGMOD PhD workshop on Innovative database research (IDAR '08). Association for Computing Machinery, New York, NY, USA, 31–36. DOI: https://doi.org/10.1145/1410308.1410318

[12] Chung, C., Gertz, M., and Levitt, K.: DEMIDS: A misuse detection system for database systems. In the Proceedings of the Third International IFIP TC-11 WGll.5 Working Conference on Integrity and Internal Control in Information Systems, Kluwer Academic Publishers, pp. 159-178 (1999).

[13] Lee, S. Y., Low, W. L., and Wong, P. y.: Learning Fingerprints for a Database Intrusion Detection System. In the Proceedings of the 7th European Symposium on Research in Computer Security (2002)

[14] Lee, V. C. S., Stankovic, 1. A, and Son, S. H.: Intrusion Detection in Real-time Database Systems Via Time Signatures. In the Proceedings of the 6th IEEE Real-Time Technology and Applications Symposium (2000) “OpenStack Docs: Keystone, the OpenStack Identity Service.”

[15] Y. Hu, B. Panda, A Data Mining Approach for Database Intrusion Detection, Proceedings of the ACM Symposium on Applied Computing, pp. 711-716 (2004).

[16] Barbara, D., Goel, R., and Jajodia, S. Mining Malicious Data Corruption with Hidden Markov Models. In Proceedings of the 16th Annual IFIP WG 11.3 Working Conference on Data and

Application Security, Cambridge, England, July 2002.
[17] Elisa Bertino, Ashish Kamra, Evimaria Terzi, Athena Vakali, "Intrusion detection in RBAC-administered databases", 21st Annual Comp. Security App. Conference (ACSAC) 2005.
[18] Sin Yeung Lee, Wai Lup Low, Pei Yuen Wong, "Learning Fingerprints for a Database Intrusion Detection System", 7th European Symposium on Research in Computer Security (ESORICS 2002).
[19] Lee, V. C.S., Stankovic, J. A., Son, S. H. Intrusion Detection in Real-time Database Systems Via Time Signatures.
[20] Database system concepts 4th Edition By Silberschatz-Korth-Sudarshan.