Anomalous Database Transaction detection

Harshith Reddy Sarabudla
(Master of Software Engineering)

ABSTRACT

Databases may contain sensitive, confidential data which is a major target for attackers and securing these databases is a critical issue. Protection of this data is usually enforced at the application, network, and database level. The data protection at database level includes the Access control models to limit the permissions to of legitimate users to read, write data and encryption at times. These security models are sometimes insufficient to prevent misuse, especially insider abuse by legitimate users, the access control model can prevent users from accessing the data to which they are not authorized but it is unable to prevent misuse of the data by authorized users. So, a database intrusion detection system may be used at the database layer to detect any transactions that access the data without permission or any malicious activity in the database. In this paper, we propose an anomaly detection mechanism that detects anomalous transactions by using syntax-centric and data-centric approach.

1. INTRODUCTION

Data represent today a valuable asset for organizations and companies and must be protected. Ensuring the security and privacy of data assets is a crucial and very difficult problem in the modern networked world. Data must not only be protected from external attackers but also from users within the organizations who act with malicious intent. Protecting data from insider threats requires combining different techniques. One such technique is represented by the access control system that is implemented in the database management system. However, access control mechanisms provide a first layer of defense against insider threats, these mechanisms are unable to protect against malicious insiders that have access to the sensitive data. For this purpose, database intrusion detection systems are used to detect any anomalous transaction in the database. Malicious attacks in the database can affect the data in one database and may gain access to the other databases in the same network through valid users, it is important to identify the intrusion as soon as possible before the data can misused. The intrusion detection systems will be able to identify malicious activities in the database and log these activities and notifies the database admin accordingly.

Database intrusion detection systems usually relies on pre-defined rules or known attack patterns, but they are successful in detecting only known attacks. This is signature based intrusion detection. However, this method is successful only for identifying known attacks. It is incapable of acting against intrusions with new forms of attack or malicious user actions that seem normal. For identifying such new attacks, the non-signature or anomaly based intrusion systems are used.

Conventional database security techniques focus mainly on preventing outsider attacks and have barely been of any help to identify the insider threats. The Intrusion Detection System (IDS) proposed in this paper focuses on insider threats with concentration on intrusions in RBAC based databases. The proposed intrusion detection scheme uses an anomaly detection mechanism for Role based access controlled databases.

Most non-signature or anomaly based detection systems use database logs to understand user access patterns and classify benign, malicious transactions. These models learn from the transactions they flag malicious/ genuine. Every time a new attack is found, it is added by the detection model to the training set in order to prevent future database transactions with similar patterns. However, anomaly detection systems are not 100% efficient. They generate huge number of false alerts. The abundance of false positives and false negatives makes it difficult for the security analyst to identify successful attacks and to take remedial actions.

Each IDS generates a huge amount of alerts where most of them are real while the others are not (i.e., false alert) or are redundant alerts. Sometimes genuine user queries might be deviating from the normal query, this is sometimes considered as a malicious intrusion and the detection system denies running this transaction and alerts the administrator. Such command is added to training list and taken into consideration while making the decisions in the upcoming rounds which makes the detection system not totally efficient and generate false alerts.

In this paper, we present a more efficient intrusion detection system that aims at detecting anomalous transaction while reducing the detection time window. This is done by using a data-centric approach along with the conventionally used syntax-based approach which has some limitations.

2. LIMITATIONS OF SYNTAX-CENTRIC APPROACH

Most of the Database Intrusion Setection Systems use query syntax to build regular usage profiles. However, an attacker can modify a regular query into a different way and retrieve the same data. Syntax based approaches fail to detect intrusions in such cases.

In some cases, syntactically similar queries may generate vastly different results and two syntactically distinct queries may give similar results.
Consider the following query

SELECT Name, Salary FROM Employee WHERE ID = 102 AND Dept_id = 3;

Conversely, suppose we rewrite the above query as follows

SELECT Name, Salary FROM Employee WHERE ID = 102 AND Dept_id = 3 AND Name IS NOT NULL;

This query is syntactically different (three columns in the WHERE clause) but produces the same result as the first. Most syntax-based anomaly detection schemes are likely to flag this query as

anomalous with respect to the first since the second syntax, even legitimate deviates from the normal query syntax and end up be a false positive.

To overcome this limitation of syntax-based detection approach, we combine the syntax based approach with the data centric approach thus making the anomaly detection more efficient.

3. RELATED WORK

Database intrusion detection systems mainly use two approaches: misuse detection, looking for well-known attack patterns and anomaly detection, looking for deviations from typical user behavior. The first approach works efficiently against previously known and expected intrusion actions while latter can identify new attacks. This section describes some of the anomaly based approaches that have been proposed in the past to detect anomalous database transactions.

3.1 Using Access patterns of users

The approach proposed in [3] is based on mining SQL queries which are stored in database log files and useful information can be extracted from the log files every time the access patterns of the queries are compared for anomaly. It uses relation analysis to compute dependencies or relations among user actions or accessed data to find out which columns, rows, tables, etc. or commands are usually issued or processed together. For anomaly detection when the database has role-based users a Naïve Bayes Classifier (NBC) is used as follows: for all queries in the audit logs, and for each role, the classifier is built. For each submitted query, if any of its classifiers is different from the ones in its roles, the action is considered an intrusion and an alert is generated. If role-based access policies are not implemented in the database, they propose unsupervised anomaly detection. Every user is mapped to the representative cluster and Naive Bayes classifier is applied in a manner similar to before mentioned, to determine whether the user associated with the query belongs to its representative cluster or not or a statistical test is used to identify if the query is different to representative cluster, if yes, anomaly is true and intrusion is confirmed and alarm is raised.

The intrusion detection model suggested in [3] considers query semantics to identify intrusion while approach in [1] uses a data-centric approach that focus on its semantics, considers the amount of sensitive information the data result of a query result contains.

3.2 Using Time signatures

The approach proposed in [4] uses time signatures to identify intrusion focusing on temporal features such as time span between user actions and duration of those actions. The approach uses a mean and standard deviation model built from time signatures to detect attacks from a defined range in real time database systems. This solution considers a transaction as a set of read and/or write actions for each data object which is executed in predefined update time periods. Since data keeps changing rapidly in real time database, the time taken usually to complete the processing is

sensed as time signature and for every update in database, the update time is checked against the time signature i.e. update rate and rejects the update if they don't match.

3.3 Using User behavior analysis

The model proposed in [9] detects anomalous events based on user behavior analysis through usage patterns, user profiles and session management. Detects abnormalities and malicious intrusions based on time varying patterns, user profiles, user query sequences and access pattern. User profiles such as sessions per user, Idle time, connection time, failed login attempts, privileges. An event generator monitors the transaction of operation, access rights of users, operating type used. Different patterns of queries issued by users is analyzed for similarities between different sessions that can be similar with respect to intention of user while executing the session.

Our approach is an anomaly based detection scheme that analyses the transaction signatures in terms of both syntax and data to identify malicious transactions. To limit the effect of the intrusion detection system on the performance of DBMS, we include a data usage-based detection that classifies data according to its usage.

4. PROPOSED APPROACH

The key idea in the proposed anomaly detection approach to build user profiles based on the query syntax and the data accessed by the queries and use these profiles for detecting malicious actions that are performed in the database.

The proposed approach operates in two phases, the training phase and the detection phase. During the training phase, user profiles are built based on past logs of user's activities. Profiles consist of query features extracted from the queries that are considered legitimate for that role. In the detection phase, features are extracted for the incoming query and compared with the defined profile for the user role that has requested for the query execution. Also, data-centric approach is used in addition to deal with the false positives generated in the first stage. Extraction of both syntax-based features and data-based features would hamper the detection performance of the intrusion detection system, so we come up with two steps of data centric detection that has a minimal effect on the detection performance while providing a better efficiency of anomaly detection.

i. Training phase:

First, the features are extracted that represent the syntax of the query and data in the query result. For this, legitimate records of user actions are taken from the audit logs. A normal transaction is between BEGIN and END statement in the transaction and contains clauses like select, insert, update, Delete and attributes on which the operations are performed. Features that are extracted from the SQL command include the SQL operations, attributes, user role, number of commands and command execution time and kept it in dataset. A transaction signature is created for each

query available in the audit log based on the extracted features.
Consider a database with tables Employee, Sales, Finance and set of operations on each of these tables
Ex: SELECT Name FROM Employee WHERE id=1
SELECT * FROM Sales
SELECT Profit FROM Finance
UPDATE Employee SET Name='test' WHERE id=3

For the above transaction, a signature is created of the form (UR1, Select(Employee), Select(Sales), Select(Finance), Update(Employee), 500)
UR1 denotes the user role and 500 denotes the execution time in millisecond.

An automatic transaction signature generation algorithm is used for generating transaction signatures for all the available transactions in audit logs. These signatures are classified according to the user roles and stored in a dataset which is referred to as 'User Profile'. The user profile varies for each user role.
For each user role, features are extracted from the audit logs, signatures are formed, and profile is constructed for each role. These profiles are compared against the profile of the incoming query in the detection phase.
As we previously mentioned about the limitation of syntax-based approach, we perform a data-centric detection in addition to the syntax-based user profile to limit the false positives generated by the profile comparer and make the overall detection system efficient.

From the audit logs, the data returned by all transactions i.e. attributes and tables used by a legitimate user are extracted and classified according to their role and usage.

The extracted attributes are classified as follows:

1. Classification based on user role – Attributes that are permitted for each user role are grouped accordingly
2. Classification based on usage – Attributes are grouped according to their frequency of usage for each user role.

An attribute usage calculation algorithm is used to compute the percentage of usage of each attribute according to the user role and the resulting dataset is stored as 'Data Profile'.
Data Signature: Attribute (Percentage of usage)
Data Profile: Set of Data signatures

Ex: Consider the table Employee

Employee_ID	Name	Designation	Salary

Consider for user role UR1, Employee_ID and Name has been the most returned data followed by designation and salary.

Example Data profile: (Employee_ID (40), Name (38), Designation (20), Salary (2))

Employee_ID – Attribute
40 – Percentage of usage

The Percentage of usage changes according to the usage of the attributes later during every transaction inspection depending on the frequency of usage of the existing attributes or addition of new attributes.

The 'User Profile' and 'Data Profile' are computed for each user role for all legitimate transactions available in the audit logs during the training phase.
These profiles are used in the detection phase for detecting the anomalous transactions.

ii. Detection Phase:

In the detection phase, all the incoming transactions are checked for anomaly before passing it to the DBMS for execution.
Detection of anomalous transactions is carried out in three stages

Stage 1 – Syntax based detection

Features of the incoming queries are extracted, signatures are generated and User Profile is built.

The User profile of the incoming query is compared against the pre-calculated User profile (training phase) for that particular user role.

For a legitimate transaction, the signatures i.e. user profiles match. In this case, no anomaly alarm is raised and the queries are passed to the DBMS for execution.

If the user profiles do not match, the queries are marked as a 'temporary anomaly' and undergo further processing in Stage 2 before being declared as benign or malicious.

Stage 2 – Data usage-based detection

For queries that are marked as 'temporary anomaly', the attributes projected in the query, i.e. attributes that appear in the query result are stored and the 'Data Profile' is generated. This data profile is compared against the data profile containing the previous legitimate usage percentage defined for his/her role.
If the incoming query Data profile matches with the pre-defined data profile and the percentage of usage for the requested attributes is above 10%, then the transaction is marked as 'legitimate' irrespective of the result in Stage 1 and sent to the DBMS for execution.

If the incoming query Data profile doesn't match with the pre-defined data profile, it has to undergo further detection process at Stage 3

Stage 3 – Data sensitivity-based detection

Only if a transaction is not declared legitimate in Stage 2 i.e. the attribute is having a percentage of usage below 10 for a specific user role or the attribute is newly added to the database (percentage of usage would be 0 in this case), it passed to the Stage 3 for further detection analysis.

We employ the detection scheme based on the attribute sensitivity score proposed by Md.Saiful Islam[1]. This approach considers the amount of sensitive information a query result contains to detect an anomaly. A sensitivity score is calculated for each query submitted to the database using various pattern matching techniques based on a combination of each individual attribute's score.

During training phase, sensitivity scores taken from a valid query of the associated role are considered and a sensitivity score is determined for each user role. During detection, sensitivity score is the calculated for incoming queries using the Hidden Markov model. If the calculated sensitivity score is greater than a particular threshold score, anomaly alarm is raised. The transaction is flagged as an anomaly and the respective administrator is notified.

4.1 Research tasks

- Select suitable tools to extract features of SQL queries
- Implement an automatic transaction signature generation algorithm for generating transaction signatures for all the available transactions in audit logs
- Implement an attribute usage calculation algorithm to compute the percentage of usage of each attribute according to the user role
- Implement a signature comparer that compares the signature of the received query with the pre-defined signatures.
- Implement Hidden Markov model to calculate sensitivity score of the SQL queries.

5. CONCLUSION

In this paper, we have proposed an approach to detect the anomalous transactions in databases in order to protect highly sensitive data from insiders. The proposed approach is capable of detecting the changes in access patterns based on the syntax of the input queries and the data in the query results with higher accuracy and reduced false alarm rate.

REFERENCES

[1] Mohammad Saiful Islam, Mehmet Kuzu, Murat Kantarcioglu, “A Dynamic Approach to Detect Anomalous Queries on Relational Databases”. 5th ACM Conference on Data and Application Security and Privacy 2015.

[2] Syed Rafiul Hussain, Asmaa M. Sallam, Elisa Bertino, “DetAnom: Detecting Anomalous Database Transactions by Insiders”. 5th ACM Conference on Data and Application Security and Privacy 2015.

[3] Kamra, A., Terzi, E. and E. Bertino, “Detecting Anomalous Access Patterns in Relational Databases”. Springer VLDB Journal, 17, 2008.

[4] VCS Lee, JA Stankovic, SH Son, “Intrusion Detection in Real-time Database Systems Via Time Signatures”. Sixth IEEE Real-Time Technology and Applications Symposium.

[6] Jay Kant Pratap, Devottam Gaurav “A Novel Approach to Database Intrusion Detection” International Journal of Computer Applications, 2017.

[7] AM Mostafa “False Alarm reduction scheme for database intrusion detection system”, Journal of Theoretical and Applied Information Technology 2018.

[8] Sunu Mathew, Michalis Petropoulos, Hung Q. Ngo, and Shambhu Upadhyaya.”A Data-Centric Approach to Insider Attack Detection in Database Systems”. International Workshop on Recent Advances in Intrusion Detection RAID 2010.

[9] Indu Singh, Manoj Kumar, “A Proposed model for Data Warehouse User Behaviour using Intrusion Detection System” ACM 2013.