Mechanisms for Database Intrusion Detection and Response

Michael Sintim - Koree
SE 521
March 6, 2013.
Title: Mechanisms for Database Intrusion Detection and Response

Authors: Ashish Kamra, Elisa Bertino, and Guy Lebanon

Conference: 2nd SIGMOD PhD workshop on Innovative database research, 2008.
Data represents today an important asset for companies and organizations. Some of these data are worth millions of dollars.

Organizations are taking great care at controlling access to these data for both Internal and external users.
Recently, there has been an interest in database activity monitoring solutions that continuously monitor a Database Management System (DBMS) and report any relevant suspicious activity.

This is due to Government regulations such as:
- SOX,
- PCI,
- GLBA,
- HIPAA, .... and so forth concerning data management.
Current attack techniques are more sophisticated, organized and targeted than broad based hacking days of the past.

With Greater data integration, aggregation and disclosure, preventing data theft has become a big challenge (from both Outside and within).

Standard database security controls are not of much help when it comes to preventing data theft from insiders.
  ○ Access controls mechanisms,
  ○ Authentication
  ○ Encryption Technologies
What are the new Strategies

There are two:

1. Monitor database activity using Third-party to detect any potential abnormal behaviour.

2. Revise Architectures and techniques adopted by traditional DBMS.
   ○ Intrusion Detection (ID) mechanism.

However, there few ID systems specifically tailored to a DBMS exist. Hence the goal of this work is to develop efficient and effective algorithms for detecting intrusive user behavior in a context of a DBMS, and augment that with an intrusion response engine.
Problem Statement

● Creating profiles that succinctly represent user/application-behavior interacting with a DBMS

● Developing efficient algorithms for online detection of anomalous database user/application behavior.

● Developing strategies for responding to intrusions in context of a DBMS.

● Creating a system architecture for database intrusion detection response as an integral component of a DBMS, and a prototype implementation of the same in PostgreSQL relational database.
The Approach

- System Architecture
- Detecting Anomalous Database Requests
- Detecting SQL Injection Attacks
System Architecture

Consist of three main components:

- The conventional DBMS mechanism that handles query execution process
- The database audit log files
- The ID mechanism

These components form the new extended DBMS that is enhanced with independent ID system operating at database level.

The flow of interactions for the ID process is shown in Figure 1.

Every time a query is issued, it is analyzed by the ID mechanism before execution.
System Architecture

Figure 1: System Architecture
Detecting Anomalous Database Request

In order to identify anomalous user/role access to a DBMS:

○ Use the database audit files for extracting information regarding user's actions.
○ The audit records, after being processed, are used to form initial profile representing acceptable actions.
○ Every entry in the audit file is represented as separate data unit; these units are then combined to form the desired profile.

Two Scenarios while addressing this problem:

● Role Based Access Control (RBAC) model.
  ● False Alarms

● DBMS without any role definitions.
  ● Missed Alarms
Detecting Anomalous Database Request

Specific Methodology used for ID task:

- Partition the training data into clusters using standard clustering techniques.
- Maintain a mapping for every user to its representative cluster.
- The representative cluster for a user is the cluster that contains the maximum number of training records for that user after the clustering phase.
- For every new query under observation, its representative cluster is determined by examining the user-cluster mapping.

Note: The assumption is that query is associated with a database user.
Detecting Anomalous Database Request

For the detection phase:

- Applying the naive bayes classifier in a manner similar to the supervised case, to determine whether the user associated with the query belongs to its representative cluster or not.

- A statistical test is used to identify if the query is an outlier in its representative cluster.

If the result of the statistical test is positive, the query is marked as an anomaly and an alarm is raised.
Detecting SQL Injection Attacks

- SQL Injection is an attack exploiting applications that construct SQL statements from user-supplied input.
- When an application fails to properly validate user-supplied input, it is possible for an attacker to alter the construction of backend SQL statements.
- The modified SQL queries are executed with the privileges of the application program. An attacker may abuse the privileges of the program in a manner unintended by the application program designers.
- An SQL Injection attack typically involves malicious modifications to this input either by adding additional clauses or by changing the structure of an existing clause.
- The projection clause of the query, however, is not modified and remains static because it is not constructed at run time.
Detecting SQL Injection Attacks

The two rules derived:

- The first set consists of rules binding the projection attributes of the query to the attributes used in the where clause.
- The second set of rules represent the relationship among the attributes in the where clause of the query.

These two sets of rules together form the profile of an application. To detect an attack query, we check if the relationship among its query attributes can be inferred by the set of rules in the application profile. If not, our system raises an alarm.
Our Contributions

1. We propose novel encoding schemes for SQL queries to extract useful information from them.

2. We present an approach for fingerprinting database applications based on the SQL queries submitted by them to a database.

3. We then present an anomaly detection model to detect anomalous behavior of database applications.

4. We further demonstrate how this model can be used to detect SQL Injection at the database level.
Related Work

- Database Activity Monitoring
- Database Intrusion Detection
- SQL Injection Attacks
Database Activity Monitoring

There are two categories of database activity monitoring products:

● Network - application - based
  ○ Detected hardware application that taps into a network to monitor network traffic to and from the data center.
    ■ Passive mode - Observes the traffic and report suspicious behaviors
    ■ Inline mode - All traffic to the data center passes through them

● Agent - based
  ○ A software component installed in the database server that interacts with the DBMS in order to monitor them.
Several approaches dealing with ID for operating systems and networks have been developed. However, they are not adequate for protecting databases.

The reason is that actions deemed malicious for a database application are not necessarily malicious for the network or the operating system; thus ID systems specifically designed for the latter would not be effective for database protection.

An abstract and high-level architecture of a DBMS incorporating an ID component has been recently proposed by several authors.

However, this work mainly focuses on discussing generic solutions rather than proposing concrete algorithmic approaches.

Our approach, on the other hand, builds profiles using syntactic information from SQL queries appearing in the database log, which makes our approach more general than others.
SQL Injection Attacks

- To date, there have been few approaches that apply anomaly detection techniques to identify anomalous database application behavior.

- One such technique is by F. Valeur, D. Mutz, and G. Vigna. (A learning-based approach to the detection of sql injection attacks.)

- It builds a number of different statistical query models using a set of typical application queries, and then intercepts the new queries submitted to the databases to check for anomalous behavior.

- Our work is similar in spirit to their work but uses association rule mining techniques to build application query profiles.
Future Works

● Intrusion Response

  ○ ID mechanism should have the capability to decide which response action to take under a given situation. We envisage future ID systems to be augmented with an intrusion response mechanism that is responsible for providing the end-user with a framework for defining appropriate response actions to intrusive behavior.

  ○ As part of our future work, we propose to design and implement an intrusion response mechanism within the context of a DBMS.

● Prototype Implementation
Future Works

- Prototype Implementation

A major component of this thesis is to illustrate the feasibility and efficiency of our methods. For this purpose, we intend to implement a prototype intrusion detection mechanism integrated with the PostgreSQL database server.

The main objectives of this implementation are as follows:

- To illustrate efficient implementation strategies for developing an intrusion detection and response mechanism inside a DBMS.

- To measure the overhead on normal query processing introduced by the intrusion detection and response mechanism.
Thank You

Questions ?