Implementation of Anomaly Response Architecture for Traditional Databases

S.S. More, Pramod Potdar, Prashant Rangate, Ankit Patil, Prachi Shivne, Priyanka Patil

S.S. More, Professor, Dept. Computer Science and Engineering, Sanjay Ghodawat Institutes, Atigre.
Pramod Potdar, Prashant Rangate, Ankit Patil, Prachi Shivne, Priyanka Patil,
BE Scholars, Sanjay Ghodawat Institutes, Atigre.

---***---

Abstract – *Even though the data stored in database systems is normally protected, existing security modles are not sufficient to prevent bat use, especially from the inside abuse by legitimate users. This is major problem of organizations. So we propose the role based system for detecting the anomaly and also the response system which allows only particular queries to be fired by single user. We enhance the Joint Threshold Administrative Model (JTAM) that is based on duty separation. The idea behind JTAM is that policy object should be jointly administrated by at least n database users.*

***Key Words*: Intrusion, Database, Policy, Administration, Response, Signatures, etc.**

1. INTRODUCTION

Most important asset of organizations is data stored in databases. Many of these data are worth billions of dollars, and organizations have to take great care to control the access to such data, with respect to both internal users of organization and external users, outside the organizations. Data security has a main role in the regards of information systems security.

The development of new Database Management Systems (DBMS) requires a revision of architectures and techniques adopted by traditional DBMS. An important component of this new generation security-aware DBMS is an Anomaly Detection (AD) mechanism. Even though access control mechanism is provided by DBMS, these mechanisms alone don't guarantee data security. They need to be complemented by suitable Anomaly Detection mechanisms. Insider threats are addressed by Anomaly Detection mechanisms, an increasingly important problem in today's organizations for which not many solutions have been provided.

1.1 Problem Statement

To detect and give appropriate response to the anomalous requests in database system those which are not related to given permissions and those which violate the security of the database.

Insider threats is the main problem essentially, that is, how to protect a response policy object from malicious modifications made by a database user that has legitimate access rights to the policy object. To propose an administration model referred to as the JTAM that can prevent malicious modifications to policy objects from authorized users by detection the anomaly.

1.2 Literature survey

The current Data Base Management Systems provides various degrees of security including authorization and authentication. Also the access control mechanisms that are part of the traditional database design provide additional security for the data. Two types of access control mechanisms are provided by the DBMS. Discretionary Access Control Mechanism and Mandatory Access Control Mechanism are the two types.

A RDBMS in which data and the relationships among the data are stored in tables. The data can be accessed in many different ways. For example employee database, student and banking database etc. Security policy specifies the authentication of agents who can access computer system. An intruder may try to violate security policy. The standard database security mechanism like "encryption, authentication" are insufficient to provide security. We use the term anomaly for insiders who violate the organizational policies. When a user's request does not match with normal access of information then anomaly is characterized. An anomaly detection system for relational databases is proposed by Spalka et al[7]. The main focus is on detection of anomalies in a database. They use basic statistical functions to compare reference values for relation attributes which is monitored for anomaly detection.

We are using a role based approach in which system identifies the user based on the role assigned to them. A user's request which is different from normal access of information is detected as anomalous and then proper action is taken. There are three main actions: conservative actions like sending an alert, aggressive actions simply block the anomalous request, Fine grained response actions are neither conservative nor aggressive. These actions may suspend or taint an anomalous request until some specific actions are executed by the user.

2. Design Aspects

The problem definition in Introduction has focused on key issues of security in database systems and the related work. Design Aspects discusses the design of the system to be implemented to work successfully.

2.1 System Architecture

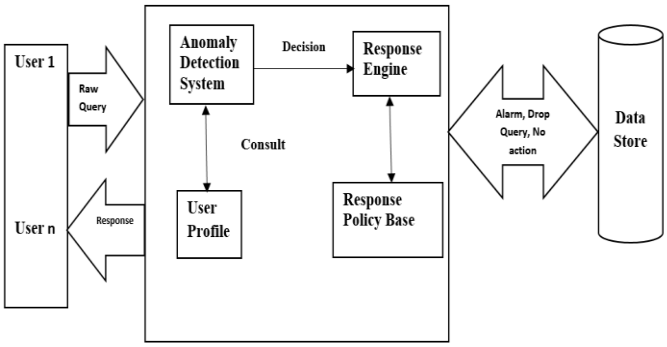


Fig -1: Architecture of System

Figure-1 shows the proposed system architecture for the anomaly detection system. The system has two main components namely Anomaly Detection System and Response Engine. The other components are user interface and the data store.

The user can fire the query through the user interface which is scanned through the Anomaly Detection System using the User Profile as a base. The query is then passed onto the Response System and the actual data store for necessary reply and action to be taken on the given query. The working of each sub system is given below:

I. Anomaly Detection System:

The Anomaly Detection System of the proposed system is responsible for identifying of the anomalous queries from the given queries. The proposed system uses a role based method for detection of irregular queries. When a query is entered by the user the query is first matched against the policy defined for the role.

II. Response Engine:

The response engine is in charge for giving suitable response to the queries acquiesced by the user. Once the query is scanned by the anomaly detection system the anomalous queries will be scaled out and the valid queries will be succumbed to the data store for execution. The anomalous queries are again traced for the type of response to be given to the query conferring to the policy decided for the role of the user. Three types of response policies are being proposed in the system.

Table -1: Anomaly Response System

Action	Description
Conservative: low severity	
NOP	No Operation. This option can be used to filter undesirable alarms
ALERT	A notification is sent
Fine-Grained: medium severity	
TAINT	The request is audited
SUSPEND	The request is put on hold till execution of a confirmation action
Aggressive: high severity	
ABORT	The anomalous request is aborted
DISCONNECT	The user session is disconnected
DENY	A subset of user privileges is denied

2.2 Modules

2.2.1 Anomaly Detection (AD) system

The discovery of an anomaly by the detection engine can be considered as a system incident. The traits of the anomaly correspond to the setting surrounding such an event. A policy can be specified taking into account the anomaly traits to guide the response engine in taking appropriate action. We suggest a role based approach for detection of anomaly activity. Authorizations are specified with admiration to roles and not with respect to discrete users. Our Anomaly Detection system builds a profile for each role and is able to regulate role intruders, that is, individuals that while holding a specific role diverge from the regular behavior of that role.

2.2.2 Anomaly Response System

The database request that has been detected as anomalous has to be given appropriate response. Anomaly Response system gives the response to detected anomaly.

2.2.3 Policy Administration

The chief problem in the administration of response policies is how to protect a policy from malicious alterations made by a Data Base Administrator that has genuine access rights to the policy object. For this purpose exclusive architecture called as JTAM can be used. A k out of 1 threshold scheme is a protocol that permits any subset of k users out of l users to create a valid policy, but that disallows the creation of a valid policy if fewer than k users contribute in the protocol. Once the policy has been created, it must be authorized for activation by at least (k – 1) administrators for the policy to be activated.

3. POLICY MATCHING

For finding the set of policies matching an anomaly we present our algorithms here.

Our approach towards policy storage in the Data Base Management System is presented first in details. Catalog tables in system store's the policies. The reason behind this is PostgreSQL. Database Management System maintains a cache of catalog tables in its buffer. Let us assume a policy database having 4 distinct policies as shown in Table 2. The graph shown in fig 2 gives detailed description of how the policy cache is maintained. Three types of nodes are present in graph: policy nodes, attribute nodes, and predicate nodes. A unique and special start node is also added to graph.

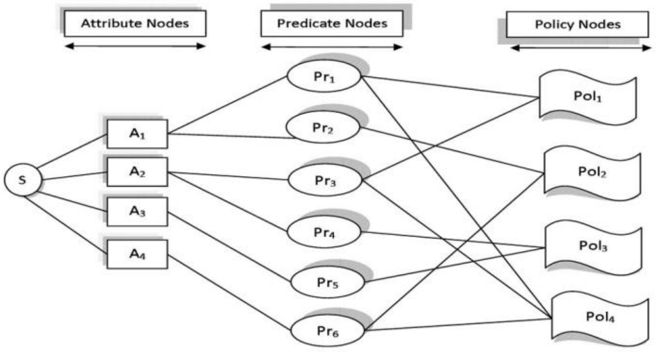


Fig -2: Policy Predicate Graph Example

Now we present our details toward policy matching. Here there are two main variations of our policy matching algorithm. 1st is "Base Policy Matching" algorithm and another is "Ordered policy Matching". Let is first discus Base Policy Matching in details.

Base Policy Matching

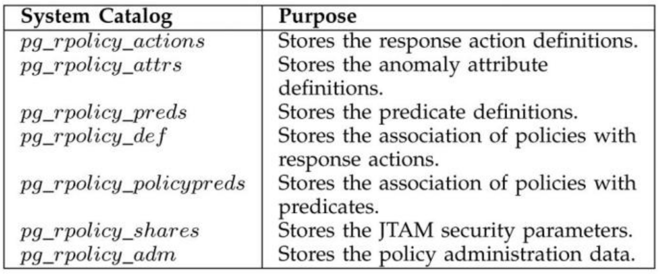
When the response engine receives an anomaly detection the policy matching algorithm is invoked. The predicates defined on A are evaluated for every attribute A in the anomaly assessment. When the evaluation of a predicate is complete, the algorithm increments the "predicate-match-count" of the connected policy nodes by 1. On the other side, connected policy nodes are marked as invalidated if the predicate evaluates to false.

Ordered Policy Matching

According to a fixed order, the search procedure in the base policy matching algorithm does not go through the predicates.

We introduce a heuristic by which the predicates are evaluated in descending order of their policy-count; the policy-count of a predicate being the number of policies that the predicate belongs to. Ordered Policy Matching algorithm is also referred as heuristic. If the correct order of predicates is not chosen it may lead to early termination of the policy.

Table -2: Response Policy System Catalogs



System Catalog	Purpose
pg_rpolicy_actions	Stores the response action definitions.
pg_rpolicy_attrs	Stores the anomaly attribute definitions.
pg_rpolicy_preds	Stores the predicate definitions.
pg_rpolicy_def	Stores the association of policies with response actions.
pg_rpolicy_policypreds	Stores the association of policies with predicates.
pg_rpolicy_shares	Stores the JTAM security parameters.
pg_rpolicy_adm	Stores the policy administration data.

Response Action Selection

It is very important in our system to provide the resolution scheme to determine the response to be implemented. He we define two rank based selection options and these options are based on the level of severity of the response action.

1. MSP (Most Severe Policy). The highest severity level of its response action determines the severity level of a response policy. This Policy is responsible to select the most severe policy form the set of matching policies.
2. LSP (Least Severe Policy). This strategy selects the least severe policy from the set of matching policies.

4. CONCLUSIONS

Vital aspects in today's industrial situation are Database and Data security as per the survey. The work done in the project includes a technique for detection of anomalous queries which are based on a "role based approach". In the database system scenario each user is allocated a specific role to which the user must stand. The old-style security mechanisms of the Data Base Management System packages are not very useful in role based scenario. "The system planned in the project makes it cooler to monitor the working of the users and gears the role based approach for security". The implemented anomaly detection mechanism was able to detect all the types of anomalous queries when tested with example dataset. The response engine in the implemented system was able to reply correctly to the anomalous queries and to the legal queries respectively with the anomaly message and result dataset. The response system sideways with the anomaly detection system makes the proposed system very effective and dependable to use. The two main problems that we talked in the context of such response policies are policy matching and policy administration.

REFERENCES

[1] Ashish Kamra, Elisa Bertino "Design and Implementation of an Intrusion Response System for Relational Databases" *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 6, JUNE 2011.*

[2] *Tran Khanh Dang, Thieu Hoa Le, Duy Tin Truong* "An Extensible Framework for Database Security Assessment and Visualization" *Proceedings of iiWAS2007*

[3] Ashish Kamra · Evimaria Terzi · Elisa Bertino "Detecting anomalous access patterns in relational databases" *The VLDB Journal (2008) 17:1063–1077*

[4] Raji V, Ashokkumar P "Protecting Database from Malicious Modifications Using JTAM" *Journal of Computer Applications*

[5] Ashish Kamra, Elisa Bertino, Rimma Nehme "Responding to Anomalous Database Requests" *SDM 2008, LNCS 5159, pages 50–66, Springer-Verlag Berlin Heidelberg 2008*

[6] Michael Kirkpatrick, Elisa Bertino "An Architecture for Contextual Insider Threat Detection"

[7] A. Spalka and J. Lehnhardt. "A comprehensive approach to anomaly detection in relational databases" In *DBSec*, pages 207-221, 2005

[8] Mohammed Masoud Javidi, Marjan Kuchaki Rafsanjani, Sattar Hashemi and Mina Sohrabi1. " An overview of anomaly based database intrusion detection systems" Indian Journal of Science and Technology

[9] A book "Database System Concepts" by Silberschatz, Korth, Sudarshan.

[10] J. Widom and S. Ceri, Active Database Systems: Triggers and Rules forAdvanced Database Processing. Morgan Kaufmann, 1995.

[11] V. Shoup, "Practical Threshold Signatures," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT), pp. 207-220, 2000.