Volume 3, Issue 5, May 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Protection of Databases by Using Policy Based Mechanism: A Survey

Abhijeet S. Sartape*
Department of Information Technology,
Sinhgad College Of Engineering,
Pune.41, India.

Prof. B. P. Vasgi
Department of Information Technology,
Sinhgad College Of Engineering,
Pune.41, India.

Abstract— In many organizations Database Security plays an important issue for their safe & secure environment. Performance of the organization or any enterprise should depends on Database Security i.e. Insider attack detection. In this paper mainly insider attack detection technique is studied which is based on user policy. Each user has its own profile data, from that profile data, user wants to activate their policy, but it must be permitted by at least k-DBAs. Other hand any policy modifications are done by any legitimate user, without administration of that policy, then it should be detected as malicious modification done by legitimate user. Once malicious modifications are detected then as per response action database, action should be taken against those DBAs. Here in this perspective of response policies two major issues are discussed such as policy matching and policy administration. To solve this policy matching problem two algorithms are introduced. Another major issue is how to prevent the malicious modifications of policies by legitimate or authorized user. To solve this problem Joint Threshold Administration Model (JTAM) is introduced. JTAM is based on cryptographic threshold signature and also separation of duties.

Keywords— Database Security, Intrusion Detection, Response actions, Threshold signatures, Policy-based user profile.

I. INTRODUCTION

In today's world many organizations or enterprises have huge amount of databases to store its data. But these data storage, maintenance and access are very hard and important issues for organizations. In these organizations there are huge no of employees wants to use their database from their respective departments. In this case each user has their own identity proof as like their userID. So by identifying their userID we detect as person is authorized person or unauthorized person. But this detection is like identify from user name and password so that known as external attack detection. But if in case legitimate or authorized person or employee doing activities that are not fit for their role & it will harmful for their organization, then how we decide that those employee activity is suspicious activity & how we detect that. So data should be stored and accessed by only authorized or legitimate users otherwise this work will be acts as an attack on database. So databases should maintain its own security policies for their safe and secure environment [4].

Gartner research presented that Database transactions activities and behaviors are examined for to detection of data leaks as well as for detection of malicious insider attacks done by legitimate user or authorized user [5],[6]. SQL injection and Data exfiltration attacks are database related attacks this are not issuing regarding operating system or the network [1]. Here each user has its own database access profile, from that profile data, user wants to activate their policy, but it must be permitted by at least k-DBAs. Other hand any policy modifications are done by any legitimate user, without administration of that policy, then it should be detected as malicious modification done by legitimate user. Once malicious modifications are detected then as per response action database, action should be taken against those DBAs. In this system response actions are based on their severities that are Low severity, Medium severity action, and high severity action. These three types of an action are taken against any abnormal transactions or any malicious modifications.

II. RELATED WORK

Yi Hu, Alina Campan, James Walden, Irina Vorobyeva, Justin Shelton [7] used data mining approach for database security. In Log examining approach given transaction of each user should be examined or tested for insider attack.

Sunu Mathew, Michalis Petropoulos, Hung Q. Ngo, and Shambhu Upadhyaya [8] are used semantics of the query for database security, which is more powerful than query syntax. In Query clustering approach external query (outlier) i.e. other than cluster of query, should be detected.

A.Kamra,E.Terziand,E.Bertino (2008) [9] worked on the issues of database management system. Here a SQL queries stored in database audit log files. The result of the mining process is used to form profiles that can be model normal database access behavior and identify intruders.

III.POLICY BASED MECHANISM FOR ANOMALY DETECTION AND RESPONSE

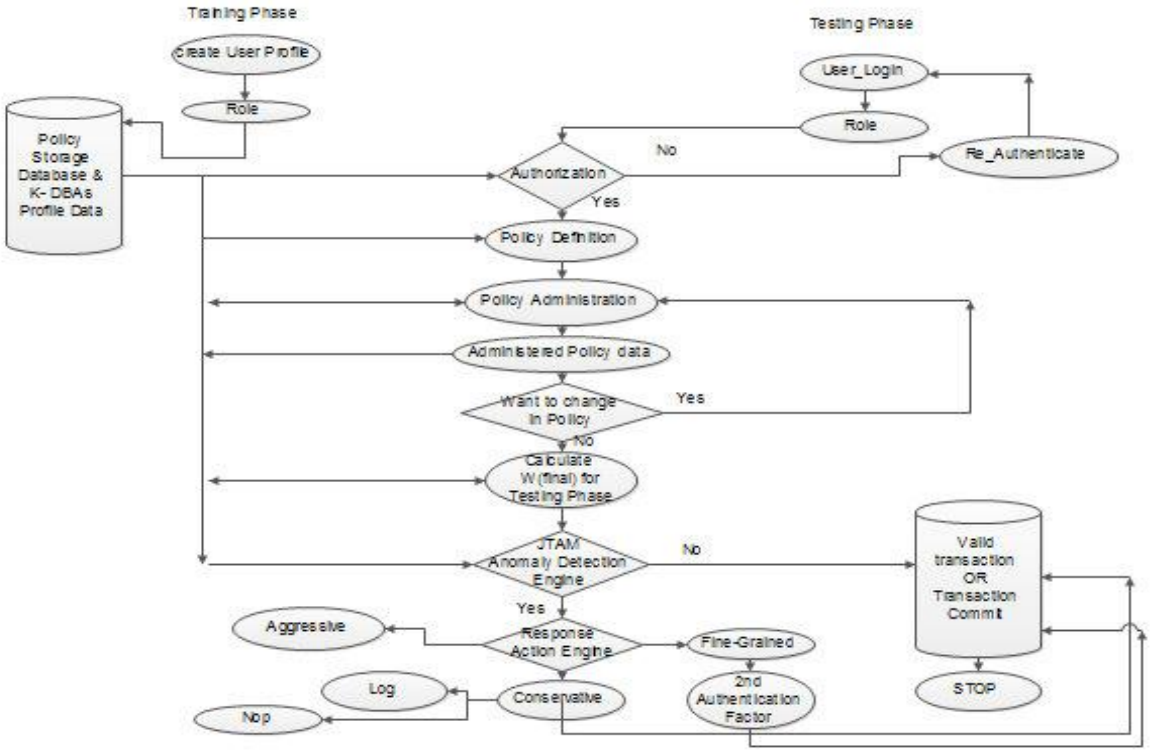

Fig. 1 Flow of an approach[2]

A. Working of an approach

In policy based mechanism each user has its own policy data. It is based on separation of duties principle i.e. fine grained data access control for different DBAs. Here each DBAs have its own policy data, then by using secret key with respective DBAs generate hash of that policy. Then by combining all the number of DBAs hash it will generate final valid signature. Then take this final valid signature as sample benchmark in training phase. By comparing with final valid signature in training phase with final valid signature of testing phase if both are match then and then only considered as, given transaction is done by normal DBAs. Otherwise it should have abnormal in nature [2].

IV.POLICY LANGUAGE

Here policy language is used to construct a system; so this policy language is used in the form of ECA model. Here event is considered as an anomaly, i.e. when system cross the normal database access profile and violated then event will be generated. ECA model is represented as follows,

ON {Event} If {Condition} THEN {Action}.

When any anomaly will be detected in the system then event will be generated. After that system will check for condition, i.e. policy respected condition will be true or false. If condition will be true then system will take an action against that profile. Here actions nothing but the response actions [1].

A. *Anomaly Attributes*

Anomaly attributes are mainly divided into two categories, contextual attributes and syntactical attributes. Contextual attributes are related with the context of the anomalous request such as user, source, role, date and time. Structural attributes are syntactical in nature, so they have fixed syntax such as database, schema, object type and SQL commands.

B. *Policy Creation*

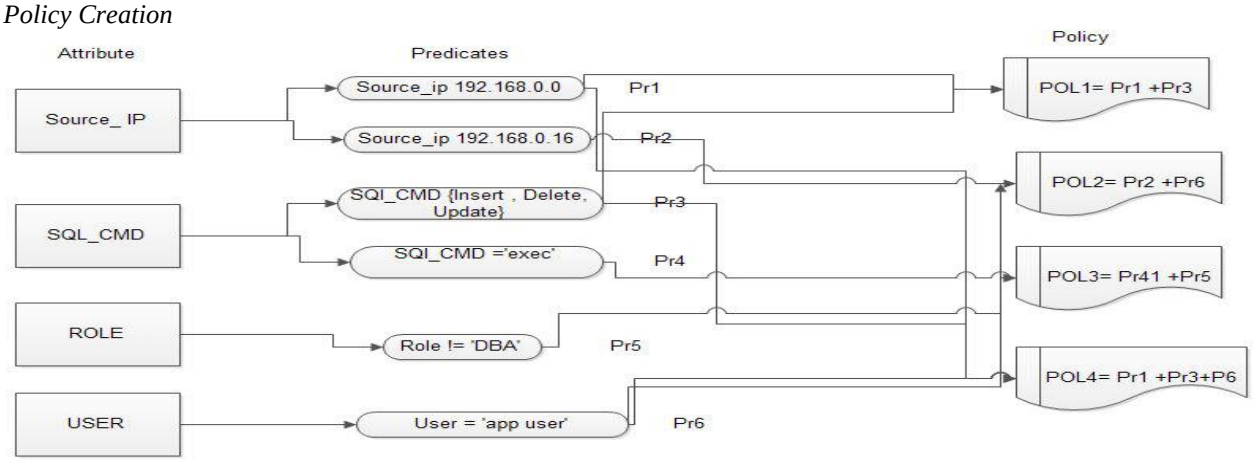

Fig. 2 Policy Creation Examples [2]

Here this attributes can be taken as contextual as well as structural. From that attributes system will generate some predicates, as per requirement. So policy condition or profile restrictions are formed by conjunction of predicates where each predicate is specified against a single anomaly attribute [3].

C. *Types of Response Action*

Action	Description
CONSERVATIVE: low severity	
NOP	No Operation , filter unwanted alarms.
LOG	Details are logged.
ALERT	Notification is sent.
FINE-GRAINED : medium severity	
TAINT	Request is audited.
SUSPEND	Put on hold till execution of confirmation.
AGGRESSIVE : high severity	
ABORT	Request is abort.
DISCONNECT	Session is disconnected.
REVOKE	Subset of user-Privileges are revoked.
DENY	Subset of user-Privileges are denied.

Fig. 3 Types of Response Actions [1]

When any anomaly will be detected in the system then event will be generated. After that system will be check for condition, if condition will be true then system will take an action against that profile. These three response actions are based on their severities that are Low severity, Medium severity action, and high severity action. Three types of an action are taken against any abnormal transactions or any malicious modifications.

V. POLICY ADMINISTRATION

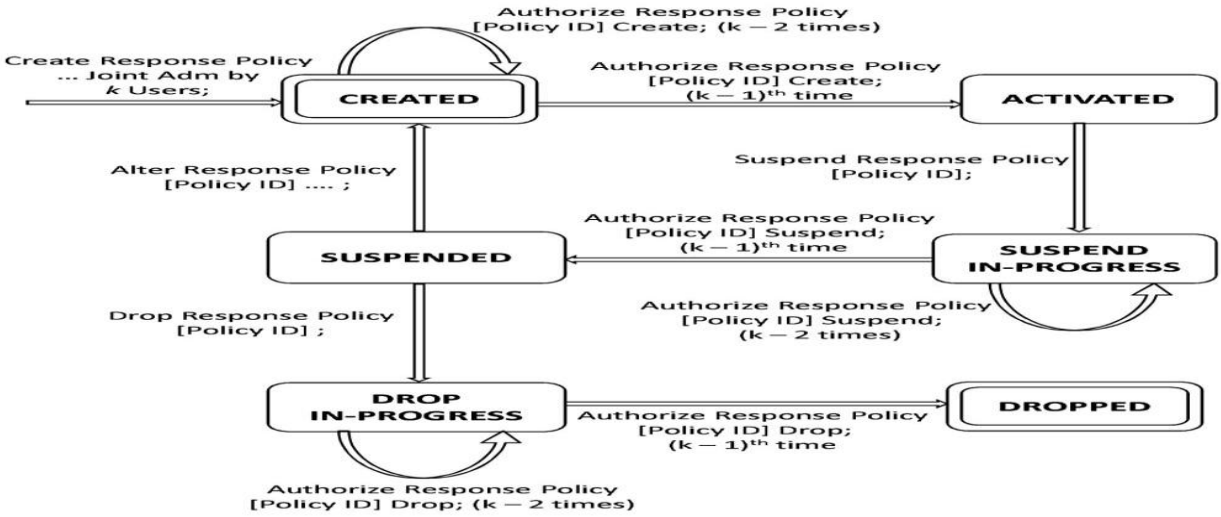

Fig. 4 Policy Administration Lifecycle [1]

Basically policy administration is performed as like any multi hand administration of that policy. System firstly generates some policy data as per user requirements. After that by using its respective secret key, for first time at an individual level system will generate hash code of that policy. Likewise each user generates its policy hash at an individual level. After that System will combine all that hash and performing any logical operation on that hash and get some final valid signature, which is another final hash value. This final valid signature will be represents training data sample. So during policy administration system will depends on at least (k-1) DBAs or user, who wants to give permission for administration of that policy. So system prevents malicious modifications by introducing JTAM (Joint Threshold Administration Model). Suppose in testing phase any malicious modifications will be done by any authorized user, then system will generates its new testing phase hash at an individual level, likewise after that by performing training phase operations, system will generates new final testing phase valid signature. Then by comparing with training phase sample with testing phase sample, system will decide that current activity should be suspicious and it takes some response action against that user.

Policy administration phases are as follows, Created, Activated, Suspend, Drop /Delete.

1) *Created:* It is the first phase of policy administration. Here each user creates its own policy and they have not any permission required for created phase. Here k DBAs can create its own policy.

2) *Activated:* In this phase after creation of any policy, it needs to be at least k-1 DBAs approved this policy. Likewise from suspend phase to activated phase, it needs to be at least k-1 DBAs approved this policy.
3) *Suspended:* In this phase when policy wants to alter, drop or made nonoperational then it needs to be at least k-1 DBAs approved this policy. After that policy will move into created phase, like that when it wants to activate in next time then it moves into activated phase and perform operation as above.
4) *Drop/Delete:* In this phase when any policy is in suspended phase then it wants to Drop or Delete then it needs to be at least k-1 DBAs approved this policy.
5)

VI. POLICY MATCHING

In this section, two algorithms are introduced (Base policy matching OR Ordered Policy matching) for finding the set of policies matching an anomaly from policy predicate table. In base policy matching algorithm, it does not go through the predicates as per the fixed order [3]. But in ordered policy matching algorithm, it goes through the predicates as per the fixed order [3].

VII. CONCLUSIONS

Here in this paper, the intrusion detection system is used for the DBMS has been performing very well. This Database Security System is mainly combination of intrusion detection unit and intrusion response unit. In detection unit it describes which type of attack will be perform against database. And after that response unit will take some actions against those attackers by using reference of response action database, which is very useful for making of trusted environment. So this JTAM based model provides protection of database security from legitimate user who has done suspicious activities. So this approach is very useful making of trusted environment.

REFERENCES

[1] Ashish Kamra and 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] Abhijeet Sartape and Prof. Vasgi B.P.,"Data- Base Security Using Different techniques: A Survey", International Journal of Computer Trends and Technology (IJCTT) - volume4Issue4 –April 2013.
[3] D. Jayanthi and m. Suresh," Intrusion Response System For relational Databases Using Joint Threshold Administration model" International Conference on computing and control engineering (ICCCE 2012) & 12 & 13 April, 2012.
[4] A. Conry-Murray, "The Threat from within. Network Computing (Aug. 2005)," http://www.networkcomputing.com/showArticle. jhtml?articleID=166400792, July 2009.
[5] R. Mogull, "Top Five Steps to Prevent Data Loss and Information Leaks. Gartner Research (July 2006)," http://www.gartner.com, 2010.
[6] M. Nicolett and J. Wheatman, "Dam Technology Provides Monitoring and Analytics with Less Overhead. Gartner Research (Nov. 2007)," http://www.gartner.com, 2010.
[7] Yi Ru, Alina Campan, James Walden, Irina Vorobyeva, Justin Shelton, "An Effective Log Mining Approach For Database intrusion Detection *", 978-1-4244-6588-0/10 IEEE, 2010.
[8] Sunu Mathew, Michalis Petropoulos, Hung Q. Ngo, and Shambhu Upadhyaya, "A Data- Centric Approach to Insider Attack Detection in Database Systems", S. Jha, R. Sommer, and C. Kreibich (Eds.): RAID 2010, LNCS 6307, pp. 382–401, 2010. Springer-Verlag Berlin Heidelberg 2010.
[9] A.Kamra, E. Terzi, and E. Bertino, "Detecting Anomalous Access Patterns in Relational Databases," J. Very Large DataBases (VLDB),vol. 17, no. 5, pp. 1063-1077, 2008.
[10] Srivastava, A, Sural S., and Majumdar, AK.: Database Intrusion Detection Using Weighted Sequence Mining, Journal of Computers, vol. 1, no. 4 (2006).
[11] Agrawal, R., Imieliilski, T., Swami, A: Mining association rules between sets of items in large databases, In Proceedings of the 1993 ACM SIGMOD international conference on Management of data (1993).
[12] Babcock, B., Chaudhuri, S., Das, G.: Dynamic sample selection for approximate query processing. In: SIGMOD Conference, pp. 539–550 (2003).
[13] Chung, C.Y., Gertz, M., Levitt, K.: Demids: a misuse detection system for database systems. In: Integrity and Internal Control Information Systems: Strategic Views on the Need for Control, pp. 159–178. Kluwer Academic Publishers, Norwell (2000).
[14] Fonseca, J., Vieira, M., Madeira, H.: Online detection of malicious data access using dbms auditing. In: Proc. of the 2008 ACM Symposium on Applied Computing (SAC 2008), pp. 1013–1020 (2008).
[15] A. Kamra, E. Bertino, and R.V. Nehme, "Responding to Anomalous Database Requests," Secure Data Management, pp. 50-66, Springer, 2008.
[16] "Oracle Database Vault Administrator's Guide 11g Release (11.1),"http://download.oracle.com/docs/cd/B28359_01/server.111/b31222/toc.htm, Jan. 2009.
[17] R. Gennaro, T. Rabin, S. Jarecki, and H. Krawczyk, "Robust and Efficient Sharing of RSA Functions," J. Cryptology,vol.20,no.3,pp.393-400,2007.

