

Intrusion Response System for Relational Databases by using Secondary Level Authentication

¹Priscilla. Gummadi, ²Hari Krishna. Deevi, ³Dr. K. Rama Krishnaiah, ⁴G. Sudhir

^{1,2,3,4}Dept. of CSE, NOVA College of Engineering & Tech., Jupudi, Ibrahimpatnam, Vijayawada, AP, India

Abstract

The intrusion response component of an overall intrusion detection system is responsible for issuing a suitable response to an anomalous request. We propose the Secondary Service Link Authenticator (SSLA) to support the intrusion response system. We follow the notion of database response policies to support our intrusion response system tailored for a DBMS. The interactive response policy language makes it very easy for the database administrators to specify appropriate response actions for different circumstances depending upon the nature of the anomalous request. The two main issues that we address in context of such response policies are that of policy matching, and policy administration. For the policy matching problem, we use two algorithms that efficiently search the policy database for policies that match an anomalous request. The experimental evaluation shows that our techniques are very efficient. The other issue that we address is that of administration of response policies to prevent malicious modifications to policy objects from legitimate users. We propose a novel Joint Threshold Administration Model (JTAM) that is based on the principle of separation of duty. The key idea in JTAM is that a policy object is jointly administered by at least k database administrator (DBAs), that is, any modification made to a policy object will be invalid unless it has been authorized by at least k DBAs. The design details of JTAM which is based on a cryptographic threshold signature scheme, and show how JTAM prevents malicious modifications to policy objects from authorized users and follows the SSLA.. We also implement JTAM in the PostgreSQL DBMS, and report experimental results on the efficiency of our techniques.

Keywords

Databases, Intrusion Detection, Response, Prevention, Policies, Threshold Signatures

1. Introduction

RECENTLY, we have seen an interest in products that continuously monitor a database system and report any relevant suspicious activity [1]. Database activity monitoring has been identified by Gartner research as one of the top five strategies that are crucial for reducing data leaks in organizations [2-3]. Such step-up in data vigilance by organizations is partly driven by various US government regulations concerning data management such as SOX, PCI, GLBA, HIPAA, and so forth [4]. Organizations have also come to realize that current attack techniques are more sophisticated, organized, and targeted than the broad-based hacking days of past. Often, it is the sensitive and proprietary data that is the real target of attackers. Also, with greater data integration, aggregation and disclosure, preventing data theft, from both inside and outside organizations, has become a major challenge. Standard database security mechanisms, such as access control, authentication, and encryption, are not of much help when it comes to preventing data theft from insiders [5]. Such threats have thus forced organizations to reevaluate security strategies for their internal databases [4]. Monitoring a database to detect potential intrusions, Intrusion Detection (ID), is a crucial technique that has to be part of any comprehensive security solution for

high-assurance database security. Note that the ID systems that are developed must be tailored for a Database Management System (DBMS) since database-related attacks such as SQL injection and data exfiltration are not malicious for the underlying operating system or the network.

Our approach to an ID mechanism consists of two main elements, specifically tailored to a DBMS: an Anomaly Detection (AD) system and an anomaly response system. We are using Secondary Service Link Authenticator SSLA which incorporates the user authentication info clearly explained in the proposed method. We follow the different types of response actions that we refer to, respectively, as conservative actions, fine-grained actions, and aggressive actions. A tainted request is marked as a potential suspicious request resulting in further monitoring of the user and possibly in the suspension or dropping of subsequent requests by the same user.

The two main issues that we address in the context of such response policies are that of policy matching and policy administration. Policy matching is the problem of searching for policies applicable to an anomalous request. When an anomaly is detected, the response system must search through the policy database and find policies that match the anomaly.

Our SSLA mechanism is a real-time intrusion detection and response system; thus efficiency of the policy search procedure is crucial. Two efficient algorithms that take as input the anomalous request details, and search through the policy database to find the matching policies. We implement our policy matching scheme in the PostgreSQL DBMS [9], and discuss relevant implementation issues. We also report experimental results that show that our techniques are very efficient.

The second issue that we address is that of administration of response policies. Intuitively, a response policy can be considered as a regular database object such as a table or a view. Privileges, such as create policy and drop policy, that are specific to a policy object type can be defined to administer policies. However, a response policy object presents a different set of challenges than other database object types. Recall that a response policy is created to select a response action to be executed in the event of an anomalous request. Consider the case of an anomalous request from a user assigned to the DBA role. Since a DBA role is assigned all possible database privileges, it will also possess the privileges to modify a response policy object. Now consider a scenario, where organizational policies require auditing and detection of malicious activities from all database users including those holding the DBA role. Thus, response policies must be created to respond to anomalous requests from all users. But since a DBA role holds privileges to alter any response policy, it is easy to see that the protection offered by the response system against a malicious DBA can trivially be bypassed. The fundamental problem in such administration model is that of conflict-of-interest. The main issue is essentially that of insider threats, 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 address this issue, we propose an administration model that is based on the well-known security principle of separation of duties

(SoD)[a]. approach instead applies the technique of threshold cryptography signatures to achieve SoD. administration model as the Joint Threshold Administration Model (JTAM) for managing response policy objects. To the best of our knowledge, ours is the only work proposing such administration model in the context of management of DBMS objects. The three main advantages of JTAM are as follows: First, it requires no changes to the existing access control mechanisms of a DBMS for achieving SoD. Second, the final signature on a policy is nonrepudiable, thus making the DBAs accountable for authorizing a policy operation. Third, and probably the most important, JTAM allows an organization to utilize existing man-power resources to address the problem of insider threats since it is no longer required to employ additional users as policy administrators.

The main contributions of this paper can be summarized as follows:

1. The system follow the previous system response policies in the context of a DBMS and a novel administration model called JTAM for administration of response policies.
2. We propose Secondary Service level Authenticator SSLA for user authentication to eliminate intrusion.

II. Related Work

The concept of database response policies was first introduced by us [7]. The current paper is a major extension of our previous work. The policy matching algorithms in the current paper take into account arbitrary predicates .while the scheme in [7] only considers equality predicates. Also, the JTAM policy administration model presented in this paper is a novel contribution. The concept of fine-grained response actions such as suspend, and taint has been introduced by us [8]. However, the work in [8] presents the design and implementation of an access control model that is capable of supporting such fine-grained response actions. Another approach toward addressing the problem of insider threats from malicious DBAs is to apply the principle of least privilege. The principle dictates that a user must be assigned only those privileges that are necessary to serve its legitimate purpose. This effectively means to restrict the privileges of the DBAs, and to create new roles for administration of response policy objects. Such approach is followed by Oracle Database using the concept of a protected schema for the administration of the database vault policies [18]. Database vault is a mechanism introduced by Oracle Database to reduce the risk of insider threats by using policies that prevent the DBAs from accessing application data. A discussion of the related work on threshold signature schemes can be found in [12]. To the best of our knowledge, ours is the first work that applies the technique of threshold signatures for the administration of DBMS objects. The policy matching problem is similar to the event matching problem in content based publish-subscribe (pubsub) systems [19] An algorithm for event-matching based on the concept of subscription trees is described in context of the GRYPHON project [20]. Many algorithms for content-based event matching are described by Pereira et al. [21]. The focus of their main algorithm is to improve the cache hit ratio of main memory access, which is not our main concern since we store the policies in the system catalogs, the contents of which are cached by the DBMS in its main memory. base policy matching algorithm is similar to the counting algorithm proposed by Yan et al. [22]. However, we provide an extension to the counting algorithm by proactively eliminating predicates that no longer need to be evaluated. An algorithm for matching predicates in database rule systems using a interval binary tree is proposed by Hanson et al. [24]. The focus

of the algorithm is on equality and inequality predicates on totally ordered domains, whereas our policy matching problem need to support arbitrary predicates. Event matching is also related to the problem of continuous query processing in streaming databases [25]. In continuous query processing, the problem that is addressed is matching multiple streaming tuples, belonging to different relations, to the stored queries. This is different (and much harder) from the policy matching problem in which we only need to match a single tuple (anomaly assessment) to the stored queries (policy conditions).

III. Policy Language

The detection of an anomaly by the detection engine can be considered as a system event. The attributes of the anomaly, such as user, role, SQL command, then correspond to the environment surrounding such an event. Intuitively, a policy can be specified taking into account the anomaly attributes to guide the response engine in taking a suitable action. Keeping this in mind, we propose an Event-Condition-Action (ECA) language for specifying response policies. Later in this section, we extend the ECA language to support novel response semantics. ECA rules have been widely investigated in the field of active databases [10]. An ECA rule is typically organized as follows:

A. ON {Event} IF {Condition} THEN {Action}

As it is well known, its semantics is as follows: if the event arises and the condition evaluates to true, the specified action is executed. In our context, an event is the detection of an anomaly by the detection engine. A condition is specified on the attributes of the detected anomaly. An action is the response action executed by the engine. In what follows, we use the term ECA policy instead of the common terms ECA rules and triggers to emphasize the fact that our ECA rules specify policies driving response actions. We next discuss in detail the various components of our language for ECA policies.

B. Anomaly Attributes

The anomaly detection mechanism provides its assessment of the anomaly using the anomaly attributes. We have identified two main categories for such attributes. The first category, referred to as contextual category, includes all attributes describing the context of the anomalous request such as user, role, source, and time. The second category, referred to as structural category, includes all attributes conveying information about the structure of the anomalous request such as SQL command, and accessed database objects. Details concerning these attributes are reported in Table 1. The detection engine submits its characterization of the anomaly using the anomaly attributes.

C. Response Actions

Once a database request has been flagged off as anomalous, an action is executed by the response system to address the anomaly. Table 2 presents a taxonomy of response actions supported by system. A tainted request is simply marked as a potential suspicious request resulting in further monitoring of the user and possibly in the suspension or dropping of subsequent requests by the same user.

Table 1: Anomaly Attributes

Attribute	Description
CONTEXTUAL	
User	The user associated with the request.
Role	The role associated with the request.
Client App	The client application associated with the request.
Source IP	The IP address associated with the request.
Date Time	Date/Time of the anomalous request.
STRUCTURAL	
Database	The database referred to in the request.
Schema	The schema referred to in the request.
Obj Type	The object types referred to in the request such as table, view, stored procedure
Obj(s)	The object name(s) referred in the request
SQLCmd	The SQL Command associated with the request
Obj Attr(s)	The attributes of the object(s) referred in the request.

We refer the reader to [7], for further details on request suspension and tainting. Note that a sequence of response actions can also be specified as a valid response. For example, Table 3, describes two response policy examples. Interactive ECA Response Policies.

Table 2: Taxonomy of Response Actions

Action	Description
CONSERVATIVE: low severity	
NOP	No Operation. This option can be used to filter unwanted alarms.
LOG	The anomaly details are logged.
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.
REVOKE	A subset of user-privileges are revoked.
DENY	A subset of user-privileges are denied.

Table 3: Response Policy Examples

Policy 1 ON ANOMALY DETECTION IF Role $\neq$ DBA and Obj Type = table and Objs IN dbo.* and SQLCmd IN {Select} THEN DISCONNECT
Policy 2 ON ANOMALY DETECTION IF Role = DBA and Source IP IN 192.168.0.0/16 and Date Time BETWEEN 0800 - 1700 THEN NOP

D. Policy Administration

the main issue in the administration of response policies is how to protect a policy from malicious modifications made by a DBA that has legitimate access rights to the policy object. To address this issue, an administration model referred to as the JTAM The threat scenario that we assume is that a DBA has all the privileges in the DBMS, and thus it is able to execute arbitrary SQL insert, update, and delete commands to make malicious modifications to the policies. Such actions are possible even if the policies are stored in the system catalogs.3 JTAM .protects a response policy against malicious modifications by maintaining a digital signature on the policy definition. The signature is then validated either periodically or upon policy usage to verify the integrity of the policy definition. we employ such scheme in the design of JTAM.

E. Policy Matching

Algorithms for finding the set of policies matching an anomaly. Such search is executed by matching the attributes of the anomaly assessment with the conditions in the policies. Base Policy Matching, Ordered Policy Matching. Response Action Selection.

III. Proposed Model

The problem we focus in this paper is the following: once an anomaly has been detected, what actions should the DBMS perform to address the anomaly? there is no single universal correct response measure to all anomalous events. The response must be tailored to the context and the details of the detected anomaly. To address such requirement, our detection engine submits to the response engine an anomaly characterization along with anomaly indication. Our approach thus supports the definition of response policies, taking into account the anomaly characteristics, for guiding the actions of the response engine.

The proposed approach is a secondary Service Level Authenticator SSLA, a scheme which includes the previous explained policies, response actions .If the overall process of these selections is once completed the system again prompts the level link. The link would maintain user credentials, user details(user creation Date and Time). Once the user response to the Link details the DBMS system automatically assigns the user specific rights. the Block diagram shows that system work nature.

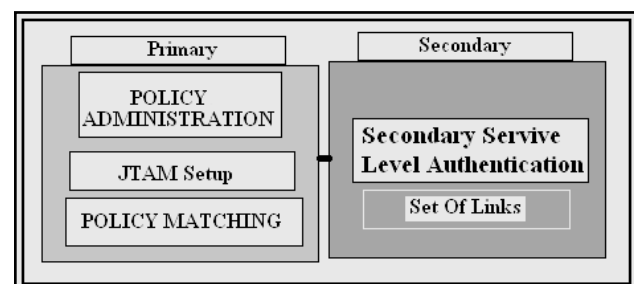


Fig. 1:

SSLA assigns links when user has been created the link works with user sessions and maintains log in database .The secondary policy enhances the primary policy by prompting the user once logged into the DBMS database system.

The link updates the details every time before login into the system. If the user provides the prompted details by selecting two links in set of link. The system automatically assigns the user rights, privileges, and activates the user schema. If the user does provide the secondary level details the system aborts the user.

IV. Experiments

The goal of the experimental evaluation is to measure the overhead incurred by the base policy matching, and the ordered policy matching algorithms. We also report experimental results on the overhead of the signature verification scheme in JTAM. In what follows, we first describe the experimental setup, and then report the evaluation results.

The experiments were conducted on a Intel(R) Core(TM)2 Duo CPU @ 2.33Ghz machine with 4GB of RAM. The operating system was OpenSuse 10.3. We extend the previous system work nature and proposed system increase the scalability of the previous system performance. The following figure shows the overall performance.

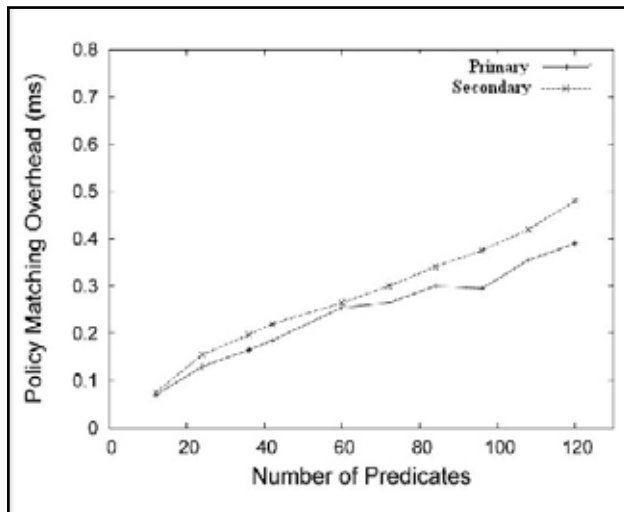


Fig. 2:

V. Conclusions and Future Work

In this paper, we have described the response component of our intrusion detection system for a DBMS. The response component is responsible for issuing a suitable response to an anomalous user request. We proposed the notion of database response policies for specifying appropriate response actions.

We presented an interactive Event-Condition-Action type response policy language that makes it very easy for the database security administrator to specify appropriate response actions for different circumstances depending upon the nature of the anomalous request. The two main issues that we addressed in the context of such response policies are policy matching, and policy administration. For the policy matching procedure, we described algorithms to efficiently search the policy database for policies matching an anomalous request assessment. We extended the PostgreSQL open-source DBMS to implement our methods. Specifically, we added support for new system catalogs to hold policy related data, implemented new SQL commands for the policy administration tasks, and integrated the policy matching code with the query processing subsystem of PostgreSQL. The experimental evaluation of our policy matching algorithms showed that our techniques are efficient. The other issue that we addressed is the administration of response policies to prevent malicious modifications to policy objects from legitimate users. We proposed a JTAM, a novel administration model, based on Shoup's threshold cryptographic signature scheme. We presented the design and the implementation details of JTAM, and reported experimental results on the efficiency of the policy signature verification mechanism.

We plan to extend our work on the following lines. An interactive response policy that requires a second factor of authentication will provide a second layer of defense when certain anomalous actions are executed against critical system resources such as anomalous access to system catalog tables. This opens the way to new research on how to organize applications to handle such interactions for the case of legacy applications and new applications. In the security area there is a lot of work dealing with retrofitting legacy applications for authorization policy enforcement [26]; we believe that such approaches can be extended to support such an interactive approach. For new applications, one can devise methodologies to organize applications that support such interactions. Notice that, however, because our approach is policy-based, the DBAs have the flexibility of designing policies that best fit the way applications are organized. We are currently in the process of implementing

the intrusion detection algorithms in the PostgreSQL DBMS as part of our overall intrusion detection and response system in a DBMS. As part of the future work, we intend to report results on the overhead of the entire system on the transaction processing capabilities of the DBMS. We introduced the Secondary Service Level Authenticator for secondary authentication model to enable the authentication at database level.

References

- [1] A. Conry-Murray, "The Threat from within. Network Computing", [Online] Available: [www.http://www.networkcomputing.com/showArticle.jhtml?articleID=166400792](http://www.networkcomputing.com/showArticle.jhtml?articleID=166400792), July 2009.
- [2] R. Mogull, "Top Five Steps to Prevent Data Loss and Information Leaks", Gartner Research (July 2006)", [Online] Available: <http://www.gartner.com>, 2010.
- [3] M. Nicolett, J. Wheatman (2010), "Dam Technology Provides Monitoring and Analytics with Less Overhead", Gartner Research, [Online] Available: <http://www.gartner.com>
- [4] R.B. Natan, "Implementing Database Security and Auditing", Digital Press, 2005.
- [5] D. Brackney, T. Goan, A. Ott, L. Martin, "The Cyber Enemy within. Countering the Threat from Malicious Insiders", Proc. Ann. Computer Security Applications Conf. (ACSAC). pp. 346-347, 2004.
- [6] A. Kamra, E. Terzi, E. Bertino, "Detecting Anomalous Access Patterns in Relational Databases", J. Very Large DataBases (VLDB), Vol. 17, No. 5, pp. 1063-1077, 2008.
- [7] A. Kamra, E. Bertino, R.V. Nehme, "Responding to Anomalous Database Requests", Secure Data Management, pp. 50- 66, Springer, 2008.
- [8] A. Kamra, E. Bertino, "Design and Implementation of SAACS: A State-Aware Access Control System", Proc. Ann. Computer Security Applications Conf. (ACSAC), 2009.
- [9] PostgreSQL 8.3., "The PostgreSQL Global Development Group" [Online] Available: <http://www.postgresql.org/>
- [10] J. Widom, S. Ceri, "Active Database Systems: Triggers and Rules for Advanced Database Processing", Morgan Kaufmann, 1995.
- [11] "Oracle Database Concepts 11g Release 1 (11.1)", [Online] Available: http://download.oracle.com/docs/cd/B28359_01/server.111/b28318/datadict.htm, July 2009.
- [12] V. Shoup, "Practical Threshold Signatures", Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT), pp. 207- 220, 2000.
- [13] R. Gennaro, T. Rabin, S. Jarecki, H. Krawczyk, "Robust and Efficient Sharing of RSA Functions", J. Cryptology, Vol. 20, No. 3, pp. 393-400, 2007.
- [14] D. Kincaid, W. Cheney, "Numerical Analysis: Mathematics of Scientific Computing", Brooks Cole, 2001.
- [15] "Openpgp Message Format. rfc 4800", [Online] Available: <http://www.ietf.org/rfc/rfc4880.txt>, 2009.
- [16] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, "Handbook of Applied Cryptography", CRC Press, 2001.
- [17] C.K. Koc, "High-Speed RSA Implementation", Technical Report tr-201, Version 2.0, RSA Laboratories, 1994.
- [18] "Oracle Database Vault Administrator's Guide 11g Release1(11.1)", [Online] Available: http://download.oracle.com/docs/cd/B28359_01/server.111/b31222/toc.htm, 2009.
- [19] F. Fabret, F. Llirbat, J.A. Pereira, I. Rocquencourt, D. Shasha, "Efficient Matching for Content-Based Publish/Subscribe Systems", technical report, India, 2000.

- [20] M.K. Aguilera, R.E. Strom, D.C. Sturman, M. Astley, T.D. Chandra, "Matching Events in a Content-Based Subscription System", Proc. Symp. Principles of Distributed Computing (PODC), pp. 53-61, 1999.
- [21] J.A. Pereira, F. Fabret, F. Llirbat, D. Shasha, "Efficient Matching for Web-Based Publish/Subscribe Systems", Proc. Int'l Conf. Cooperative Information Systems (CooplS), pp. 162-173, 2000.
- [22] T.W. Yan, H. Garcí'a-Molina, "Index Structures for Selective Dissemination of Information under the Boolean Model", ACM Trans. Database Systems, Vol. 19, No. 2, pp. 332-364, 1994.
- [23] A. Campailla, S. Chaki, E. Clarke, S. Jha, H. Veith, "Efficient Filtering in Publish-Subscribe Systems Using Binary Decision Diagrams", Proc. Int'l Conf. Software Eng. (ICSE), pp. 443-452, 2001.
- [24] E.N. Hanson, M. Chaabouni, C.-H. Kim, Y.-W. Wang, "A Predicate Matching Algorithm for Database Rule Systems", Proc. ACM SIGMOD, Vol. 19, No. 2, pp. 271-280, 1990.
- [25] H.-S. Lim, J.-G. Lee, M.-J. Lee, K.-Y. Whang, I.-Y. Song, "Continuous Query Processing in Data Streams Using Duality of Data and Queries", Proc. ACM SIGMOD, pp. 313-324, 2006.
- [26] V. Ganapathy, T. Jaeger, S. Jha, "Retrofitting Legacy Code for Authorization Policy Enforcement", Proc. IEEE Symp. Security and Privacy, pp. 214-229, 2006.



Dr. K. Rama Krishnaiah is a highly qualified person holding M.Tech and Ph.D degree in CSE, an efficient and eminent academician. He is an outstanding administrator, a prolific researcher, who has published 21 research papers in various international journals and a forward looking educationist. He worked in prestigious K L University for 13 years and he contributed his service for NBA accreditation in May 2004, Aug 2007 with 'record rating', ISO 9001:2000 in 2004, Autonomous status in 2006, NAAC accreditation of UGC in 2008 and University status in 2009.



G. Priscilla was born in Guntur Dt, Andhra Pradesh, India. She received M.Sc.(I.S) in ANU University, Nambur, Guntur, Andhra Pradesh, India. Presently, she is pursuing M.Tech in CSE Nova college of Engineering, Jupudi, Vijayawada.



Mr. D. Hari Krishna is a qualified person in M.Tech and experienced in presumed engineering colleges which contributed his service and skills to guide the students and their activities. He published in various National journals and workshops with his incredible work to gain the knowledge for future errands.