Decentralized Computation of Attack Discovery using Relational Databases

S. Jeya,
Department of Computer Applications
PET Engineering College, Vallioor
Tamil Nadu, India
jeyapetmca@gmail.com

S. Muthu Perumal Pillai,
Department of Computer Applications
PET Engineering College, Vallioor
Tamil Nadu, India
smpp_india@yahoo.co.in

Abstract

Intrusion detection system for relational database is responsible for issuing a suitable response to an anomalous request. We propose the notion of database response policies to support our intrusion response system tailored for a DBMS. Our 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 are that of data matching, and data administration. 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. We present 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. We also implement JTAM in the PostgreSQL DBMS, and report experimental results on the efficiency of our techniques.

Keywords: Databases, intrusion detection, response, prevention, threshold signatures

1. Introduction

In this research, we have described the response component of our intrusion detection system for a relational database. 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. 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. 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 infiltration 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.

There are four fundamental functions of classification detection system: Monitoring, Analysis, Response, and Generating Reports [1]. The different sources of event information can be drawn from different levels of the system, with network, host, and application monitoring system. Analysis makes sense of the events derived from the information sources, deciding when those events indicate that intrusions are occurring or have already taken place. Responses can be generated involving some automated intervention on the part of the system, and involving reporting IDS findings to humans, who are then expected to take action based on those reports.

The main contributions of this research can be summarized as we present a framework for specifying intrusion response policies and monitoring analysis of system events and user behaviors; We present algorithms to efficiently search the policy database for policies that match an anomalous request and testing the security states of system configurations; We extend the data stream classification with our response policy mechanism, managing operating system audit and logging mechanisms and the data they generate and conduct an experimental evaluation of our techniques [25].

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 provides an overview of Policy language 4 discusses our approach in detail. Section 5 then describes the experimental result of our technique. Finally, Section 6 concludes with directions to future works.

.

2. Related Work

The current paper is a major extension of our previous work. Novelty detection is also closely related to outlier detection techniques. There are many outlier detection techniques available [6, 11]. Some of them are also applicable to data streams; this paper significantly extends our previous work on IDS using GA. First, in our previous work, we did not consider the time constraints, these time constraints impose several restrictions on the classification algorithm, making classification more challenging. We encounter these challenges and provide efficient solutions. Second, it adds considerable amount of crossover and mutation analysis over the intrusion.

In this way [3, 14, 9], a normal profile can be built by learning the patterns of the short system-call sequences. Programs that show sequences that deviate from normal sequence profiles are considered to be victims. Namely, intrusion detection by learning the behavior of a program can be transformed to the problem of learning and classifying temporal sequence data [15, 24]. Machine-learning techniques that are known for good solutions for this kind of problem have been applied. ADAM (Audit Data Analysis and Mining) is an online network based IDS which uses association rules algorithm in detection [10, 21].

Learning the behavior of a program is a well-known and widely used intrusion-detection paradigm. Several kinds of intrusion can occur by inducing program misuse that exploits the bugs of the specific program [12, 18]. Victim programs behave differently from normal execution programs [16]. Therefore, learning normal behaviors and recognizing significant deviations can be efficient for anomaly detection. Though there are many ways to observe the behavior of a program, capturing the system call sequences is a typical and efficient method [20].

Various techniques have been developed to extract comprehensible classification rules from time series. Many of them share some characteristics, such as segmenting the original time series and then extracting features from these sections [2, 8, and 13]. Besides using the raw values of the time series, static features such as maximum, average, or signal to noise ratio are used to describe the characteristics of the time series [7, 17]. Traditional stream classification techniques also make impractical assumptions about the availability of labeled data. Most techniques assume that the true label of a data point can be accessed as soon as it has been classified by the classification model [23]. Thus, according to their assumption, the existing model can be updated immediately using the labeled instance. In reality, we would not be so lucky in obtaining the label of a data instance immediately, since manual labeling of data is time consuming and costly. Our technique is related to both data stream classification and novelty detection. Data stream classification has been an interesting research topic for years, and many approaches are available [22, 19].

An algorithm for event-matching based on the concept of subscription trees is described in context of the GRYPHON project. The algorithm preprocesses the set of subscriptions to build a subscription tree such that each node of the tree is an elementary test on an event attribute. The leaves of the subscription tree are the actual subscriptions. The matching algorithm walks through the subscription tree to find the set of matching subscriptions. Since no analysis of the preprocessing algorithm is provided, it is not clear if the order according to which subscriptions are chosen affects the size of the subscription tree. Also, the scheme is formulated only for elementary predicates, and it has been optimized only for the equality predicates. However, for the policy matching problem, we need to consider arbitrary predicates.

3. 3. Policy Language

A policy can be specified taking into account the anomaly attributes to guide the response engine in taking a suitable action. An Event Condition Action (ECA) language for specifying response policies

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

For example,
if {the connection has following information: source IP address 209.11.??.??; destination IP address: 130.18.176+?.??; source port number: 42335; destination port number: 80; connection time: 482 seconds; the connection is stopped by the originator; the protocol used is TCP; the originator sent 7320 bytes of data; and the responder sent 38891 bytes of data }then {stop the connection}

The *condition* usually refers to a match between current network connection and the rules in IDS, such as source and destination IP addresses and port numbers used in TCP/IP network protocols, duration of the connection, protocol used, etc., indicating the probability of an intrusion.

The *act* field usually refers to an action defined by the security policies within an organization, such as reporting an alert to the system administrator, stopping the connection, logging a message into system audit files, or all of the above.

Current system experimenting to evolve a comprehensive set of data based on the significance of monitored parameters at the various levels. Accordingly, it predefined many different action types, but it is the classifier system, which will choose the appropriate action depending on the input message. The decision is based on the following actions: Take no action, Informing the system administrator via e-mail or messaging system, Change the priority of user processes, Change access privileges of certain user, Block a particular IP address or sender, Disallow establishing a remote connection request, Termination of existing network connection, Restarting of a particular machine, etc.

3.1. Policy Condition

Let PA = {A1; A2 . . .An} be the set of anomaly attributes where each attribute Ai has domain Ti of values. Let a predicate Pr be defined as Pr: Ak ɵ c, where Ak Є PA, ɵ is a comparison operator in {>; <; >=; <=; =; ! =; like; IN; BETWEEN}, and c is a constant value in Tk. The condition of a response policy Pol is defined as Pol(C) : Prk and Prl and . . . and Prm where Prk; Prl . . . Prm are predicates of type Pr.

4. Proposed System

We present our 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. Fig. 1 illustrates the architecture of proposed system as well as policy matching, policy administration, database management system and JTAM.

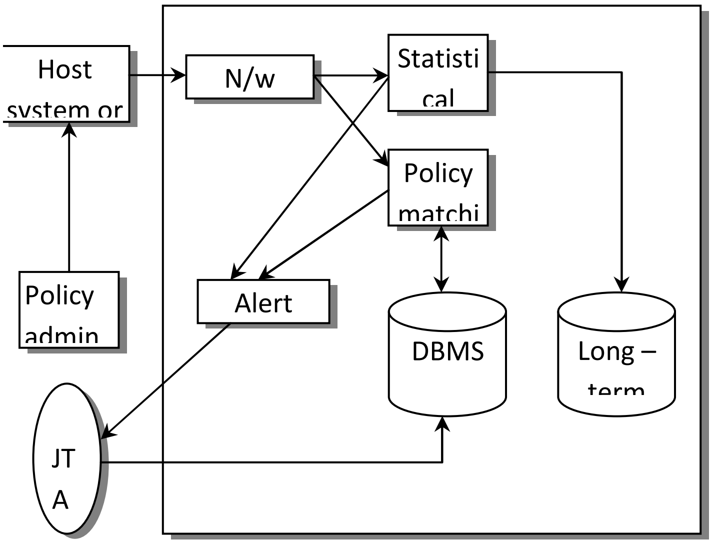

Fig. 1. Architecture of Proposed system

4.1. Joint Threshold Administration Model (JTAM)

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. We present 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. Most existing intrusion detection systems either use packet-level information or user activities to make decisions on intrusive activities. It described an intrusion detection system that can simultaneously monitor network activities at different levels such as packet level, process level system level and user level, it can detect both inside misuse and outside attacks. The main emphasis of this work is to examine the feasibility of using JTAM for robust intrusion detection. It has some unique features of simultaneous monitoring at multi-level to detect both known and unknown intrusions and generate specific response. The developed system will perform real-time monitoring, analyzing, and generating appropriate response to intrusive activities.

As a security principle, the primary objective of separation of duties SoD is prevention of fraud insider threats, and user generated errors. Such objective is traditionally achieved by dividing the task and its associated privileges among multiple users. However, the approach of using privilege dissemination is not applicable to our case as we assume the DBAs to possess all possible privileges in the DBMS. Our approach instead applies the technique of threshold cryptography signatures to achieve SoD. A DBA authorizes a policy operation, such as create or drop, by submitting a signature share on the policy. At least k signature shares are required to form a valid final signature on a policy, where k is a threshold parameter defined for each policy at the time of policy creation.

The final signature is then validated either periodically or upon policy usage to detect any malicious modifications to the policies. The key idea in our approach is that a policy operation is invalid unless it has been authorized by at least k DBAs. We thus refer to our 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.

4.2. Attacks and defense

We describe possible attacks on JTAM and strategies to protect from them. Recall that the threat scenario that we address is that a DBA has all the privileges in the DBMS, and thus it is able to execute arbitrary SQL commands on the sys_response_policy catalog.

It is possible for a malicious administrator to replace a valid signature share with some other signature share that is generated on a different policy definition. However, such attack will fail as the final signature that is produced by the signature share combining algorithm will not be valid. A malicious administrator can block the creation of a valid policy. We do not see this as a major problem since the threat scenario that we address is malicious modifications to existing policies, and not generation of policies themselves.

A policy may be modified by a malicious DBA using the SQL update statement. However, all policy definition attributes that need to be protected are hashed and signed; therefore any modification to such attributes through the SQL update command will invalidate the final signature on the policy.

5. Experimental Result

We have introduced new commands in PostgreSQL for creation, activation, suspension, and dropping of response policies. We have also added seven new system catalog tables that store the response policy data. We have instrumented the query processing subsystem of PostgreSQL with our anomaly detection and response mechanism. A user request, after being parsed, passes through the detection mechanism. The policy matching procedure is invoked for every request that is detected as anomalous. We then apply the MSP or the LSP option to choose a single policy out of the set of policies returned by the policy matching algorithm. We also report experimental results on the overhead of the signature verification scheme in JTAM.

We use the following seven anomaly attributes for our experimental evaluation: User, Client App, Source IP, Database, Objs, destination IP and SQLCmd. The predicate generation code randomly assigns set valued data to these anomaly attributes to create the policy predicates. The policy generation code randomly assigns such predicates to policy conditions to create the policies. The experiments were run on an Intel P-IV machine with 4 GB memory and 3 GHz dual processor CPU.

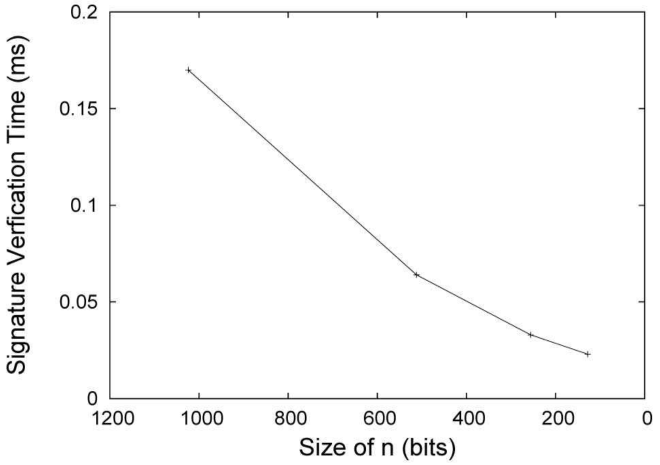

Fig. 2. Single policy signature verification

We now report results on the overhead of the signature verification scheme in JTAM. For this experiment, we set k = 2, l = 5, and e = 17. The size of the RSA modulus, n, is set to 1,024 bits. For such setup, the signature verification overhead for a single policy is approximately 0.17 ms. such overhead value confirms the low computational complexity associated with the RSA signature verification scheme. In figure 2, one approach to reduce the signature verification overhead is by decreasing the size of n such strategy, however, is not recommended since 1,024 bits is the recommended size of n to ensure sufficient security of the RSA algorithm. Therefore, a better strategy is to create a dedicated DBMS process that periodically polls the policy tables, and verifies the signature on all the policies.

The algorithm was run and returned impressive results. This corresponds to the desired results of a high detection rate and a low false positive rate. Both separate runs of the experiment produced the same final individual, yielding the same statistics. This demonstrated that the best model generated using training data successfully was able to detect unknown attacks over previously unobserved data.

6. Conclusion and Future Work

Intrusion detection is a viable and practical approach for providing a different notion of security in our huge and existing infrastructure of computer and network systems. Intrusion detection system for relational database should be more helpful for identification of network anomalous behaviors. The proposed method not only improved the detection performance but also reduced the time requirements. In this research, we have described the response component of our intrusion detection system depend on DBMS. The response component is responsible for issuing a suitable response to an anomalous user request. We have addressed several real-world problems related to JTAM classification. We have proposed a solution to the concept evolution problem, which has been ignored by most of the existing techniques.

Future work includes creating a standard test data set for the JTAM proposed in this research and applying it to a test environment. It is necessary to find network structures that are particularly good for intrusion detection by analyzing the evolved structures. Also to extend our work on 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. Combining knowledge from different security sensors into a standard test is another promising area in this work.

References

[1] Abdulhadi Shoufan, Thorsten Wink, H. Gregor Molter and Eike Kohnert, “A Novel Cryptoprocessor Architecture for the McEliece Public-Key Cryptosystem”, IEEE Transactions On Computers, Pg.No. 1533-1546, Vol. 59, No. 11, Nov. 2010.

[2] Akrivi Vlachou, Christos Doulkeridis, Yannis Kotidis and Michalis Vazirgiannis, “Efficient Routing of Subspace Skyline Queries over Highly Distributed Data”, IEEE Transactions On Knowledge And Data Engineering, Pg.No. 1694-1708, Vol. 22, No. 12, Dec. 2010.

[3] Ashish Kamra and Elisa Bertino, “Design and Implementation of an Intrusion Response System for Relational Databases”, IEEE Transactions On Knowledge And Data Engineering, Pg.No.875-888, Vol. 23, No. 6, Jun. 2011.

[4] P.A. Bonatti, J.L. De Coi, D. Olmedilla, and L. Sauro, ”A Rule-Based Trust Negotiation System”, IEEE Transactions On Knowledge And Data Engineering, Pg.No. 1507-1514, Vol. 22, No. 11, Nov. 2010.

[5] M. Brian Blake, and Michael F. Nowlan, “Knowledge Discovery in Services (KDS): Aggregating Software Services to Discover Enterprise Mashups”, IEEE Transactions On Knowledge And Data Engineering, Pg.No.889-901, Vol. 23, No. 6, Jun. 2011.

[6] Chun-I Fan, Ling-Ying Huang and Pei-Hsiu Ho, “Anonymous Multireceiver Identity-Based Encryption”, IEEE Transactions On Computers, Pg.No. 1239- 1249, vol. 59, No. 9, Sep. 2010.

[7] Claudia Marinica and Fabrice Guillet, “Knowledge-Based Interactive Postmining of Association Rules Using Ontologies”, IEEE Transactions On Knowledge And Data Engineering, Pg.No. 784-798, Vol. 22, No. 6, Jun. 2010.

[8] Domenico Ficara, Andrea Di Pietro, Stefano Giordano, Gregorio Procissi, Fabio Vitucci, and Gianni Antichi, “Differential Encoding of DFAs for Fast Regular Expression Matching”, IEEE/ACM Transactions On Networking, Pg.No.683-694, Vol. 19, No. 3, Jun. 2011.

[9] Dominik Fisch, Thiemo Gruber, and Bernhard Sick, “SwiftRule: Mining Comprehensible Classification Rules for Time Series Analysis”, IEEE Transactions On Knowledge And Data Engineering, Vol. 23, No. 5, May 2011.

[10] Emilio Miguela, Pedro, Keith E. Brown, Yvan R. Petillot, and David M. Lane, “Semantic Knowledge-Based Framework to Improve the Situation Awareness of Autonomous Underwater Vehicles”, IEEE Transactions On Knowledge And Data Engineering, Pg.No.759-773, Vol. 23, No. 5, May 2011.

[11] Eric Hsueh-Chan Lu, Vincent S. Tseng, Member, IEEE, and Philip S. Yu, “Mining Cluster-Based Temporal Mobile Sequential Patterns in Location-Based Service Environments”, IEEE Transactions On Knowledge And Data Engineering, Pg.No.914-927, Vol. 23, No. 6, Jun. 2011.

[12] Hannes Frey and Ivan Stojmenovic, “On Delivery Guarantees and Worst-Case Forwarding Bounds of Elementary Face Routing Components in Ad Hoc and Sensor Networks”, IEEE Transactions On Computers, Pg.No.1224-1238, Vol. 59, No. 9, Sep. 2010.

[13] Hua Lu, and Man Lung Yiu, “On Computing Farthest Dominated Locations”, IEEE Transactions On Knowledge And Data Engineering, Vol. 23, No. 6, Jun. 2011.

[14] Hyunjin Lee, Sangyeun Cho and Bruce R. Childers, “PERFECTORY: A Fault-Tolerant Directory Memory Architecture”, IEEE Transactions On Computers, Pg.No 638-650, Vol. 59, No. 5, May 2010.

[15] Ioannis Hatzilygeroudis, and Jim Prentzas, “Integrated Rule-Based Learning and Inference”, IEEE Transactions On Knowledge And Data Engineering, Pg.No. 1549-1563, Vol. 22, No. 11, Nov. 2010.

[16] Irem Y. Tumer, and Carol S. Smidts, “Integrated Design-Stage Failure Analysis of Software-Driven Hardware Systems”, IEEE Transactions On Computers, Pg.No.1072-1084, Vol. 60, No. 8, Aug. 2011.

[17] Javier Carretero, Xavier Vera, Pedro Chaparro, and Jaume Abella, “Microarchitectural Online Testing for Failure Detection in Memory Order Buffers”, IEEE Transactions On Computers, Pg.No. 623-637, Vol. 59, No. 5, May 2010.

[18] Mahesh Balakrishnan, Tudor Marian, Kenneth P. Birman, Hakim Weatherspoon, and Lakshmi Ganesh, “Maelstrom: Transparent Error Correction for Communication Between Data Centers”, IEEE/ACM Transactions On Networking, Pg.No.617-629, Vol. 19, No. 3, Jun. 2011.

[19] Matt Duckham, Doron Nussbaum, “Efficient, Decentralized Computation of the Topology of Spatial Regions”, IEEE Transactions On Computers, Pg.No.1100-1113, Vol. 60, No. 8, Aug. 2011

[20] Mehran Mozaffari-Kermani and Arash Reyhani-Masoleh, “Concurrent Structure-Independent Fault Detection Schemes for the Advanced Encryption Standard”, IEEE Transactions On Computers, Pg.No.608-623, Vol. 59, No. 5, May 2010.

[21] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani Thuraisingham, “Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints”, IEEE Transactions On Knowledge And Data Engineering, Pg.No.859-874 , Vol. 23, No. 6, Jun. 2011.

[22] Panagiotis Papadimitriou and Hector Garcia-Molina, “Data Leakage Detection”, IEEE Transactions On Knowledge And Data Engineering, Pg.No.51-64, Vol. 23, No. 1, Jan. 2011.

[23] Rachid Hadjidj and Hanifa Boucheneb, “Efficient Reachability Analysis for Time Petri Nets”, IEEE Transactions on Computers, Vol. 60, No. 8, Aug. 2011.

[24] Songqing Chen, Shiping Chen, Xinyuan Wang, Zhao Zhang and Sushil Jajodia, “An Application-Level Data Transparent Authentication Scheme without Communication Overhead”, IEEE Transactions On Computers, Pg.No. 943-954, Vol. 59, No. 7, Jul. 2010.

[25] Yu-Wei Eric Sung, Xin Sun, Sanjay G. Rao, Geoffrey G. Xie, and David A. Maltz, “Towards Systematic Design of Enterprise Networks”, IEEE/ACM Transactions On Networking, Pg.No.695-707, Vol. 19, No. 3, Jun. 2011.