MONITORING MISSION CRITICAL DATA FOR INTEGRITY AND AVAILABILITY

Michael Gertz

Department of Computer Science, University of California at Davis, USA
gertz@cs.ucdavis.edu

George Csaba

IPLocks Inc., Santa Clara, USA
gcsaba@iplocks.com

Abstract: Protecting the integrity, confidentiality, and availability of mission critical data is one of the primary objectives of IT departments in industry, government, and research. Standard techniques to realize these objectives are often confined to network- and host-based intrusion detection systems, which are known to be inappropriate for handling security threats caused by insiders. This paper introduces the concept of data monitoring systems as an additional line of defense against external and internal security threats. These systems, which are closely coupled with a database managing mission critical data, provide IT personnel with effective means for specifying, detecting, and responding to anomalous behavior of data and data accesses caused by users and applications.

Key words: data auditing, data integrity, monitoring, anomaly detection

1. INTRODUCTION

The ever-growing dependence of businesses on data and increasing customer demands for more functionality to operate on and utilize the data

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35693-8_16](https://doi.org/10.1007/978-0-387-35693-8_16)

M. Gertz (ed.), *Integrity and Internal Control in Information Systems V*

© IFIP International Federation for Information Processing 2003

has become a major challenge for IT personnel in charge of protecting the data against various security threats. A primary security objective in such businesses is to guard mission critical data against breaches of integrity, availability, and confidentiality, and thus to prevent serious financial losses, damage to the reputation of the business, and legal or regulatory problems. In this paper, data integrity is concerned with actions against data that might result in incorrect or incomplete data. The focus in the context of data availability is to ensure that data is available when needed. Finally, data confidentiality is concerned with mechanisms aimed at preventing the release of and access to critical data to non-authorized users.

In order to address these security concerns, which are well understood in practice and theory, today's businesses employ a variety of techniques to ensure the above objectives. Beside role and access based security mechanisms provided by commercial database management systems (DBMS) [3], the most prominent type of technique is the usage of *intrusion detection systems* (IDS) [1,4,6,7]. Such systems typically operate on the network or operation system layer, resulting in so-called *network* and *host-based intrusion detection systems*, respectively. While these systems have gained quite some popularity in recent years, it is also well known that they only offer a part of the solution to the above security problems. In particular, as recent reports clearly indicate [1,9], IDS are an effective means to guard an information system infrastructure against external security threats but they are insufficient when it comes to dealing with internal security threats. That is, in case of authorized users who have legitimate access to data, IDS are typically not capable to detect and/or prevent such users from tampering with the integrity, availability and confidentiality of the data. This problem is well known in the literature as *insider problem* [2,8]. What is needed is a framework that adds another layer of security mechanisms to existing IDS. Ideally, this layer should be as close to the DBMS managing the mission critical data as possible, and it should provide IT security personnel with effective means to guard the data against various security threats. Compared to the above mentioned systems, this layer then is "data centric" in the sense that it observes and analyses the behavior of the data rather than directly the behavior of users. Naturally, such an approach seems reasonable in many practical application scenarios, for example, where applications share database accounts and thus simple auditing/logging at the database level does not provide sufficient information about insider misuse.

In this paper, we present the philosophy and techniques underlying different audit systems developed at IPLocks Inc. The objectives of these systems are to provide security personnel with system components that allow

for the specification and analysis of the behavior of mission critical data managed in a relational DBMS. During normal operation of the DBMS, these systems collect a variety of information about the data managed by the DBMS, establish profiles, and compare these profiles against the behavior of the data in order to determine possible vulnerabilities and security breaches. A major aim of these systems is to minimize the impact of data collection and auditing activities on the operational database and to secure audit system components and collected data in a system separate from the production database.

In Section 2, we describe the data audit system, whose main function is to determine and analyze the behavior and characteristics of mission critical data managed in a DBMS. Section 3 discusses a similar approach in the context of various metadata utilized by the DBMS. Section 4 outlines how information about accesses to the data is collected and analyzed. In Section 5, the technical challenges and possible approaches in realizing the audit systems are described.

2. DATA AUDIT SYSTEM

Security breaches are often not detected until a user or customer discovers incorrect, incomplete, or unavailable data during normal operations against a production database. Another observation is that although often databases are designed with sufficient security mechanisms, these mechanisms are not kept up-to-date in the presence of new applications, users, and requirements, thus resulting in vulnerabilities malicious users can exploit. Rather than to monitor the behavior of applications and users in a relatively dynamic environment, thus making any monitoring and auditing strategy inherently complex, the objective of the data audit system is to learn and analyze the behavior of mission critical data as starting point for discovering vulnerabilities and necessary security mechanisms. Deviations of the data characteristics from what has been learned or specified result in user specified actions.

Given a production database, the data audit system is set up as follows. First, the portion of the database containing mission critical data is identified. That is, security personnel, in cooperation with the database administration, identify those relations in the database that play a crucial role in serving applications and business needs and whose integrity cannot be restored that easily. Often such relations can be identified by analyzing mission critical applications or the sensitivity of the data as prescribed by

regulations and laws. The specification of mission critical data can occur at different levels of granularity, ranging from a group of relations to individual attributes in one or more relations. In this initial phase, the next step then is the analysis of the specified relations and attributes in terms of data characteristics. The analysis employs statistical methods such as distribution of attribute values, minimum, maximum and average values for groups of tuples. Characteristics of groups of tuples are based on appropriate grouping criteria (using the SQL group by function). Correlations among tuples from different relations are determined using, e.g., foreign key dependencies.

The outcome of the analysis of mission critical data not only provides for an appropriate setup for the data audit system but it also provides security personnel with important information about the characteristics of the data they are concerned about. For example, the analysis can reveal the existence of data that is “out of bound”, i.e., showing properties that are not conform to what is assumed or expected about the data. Data visualization tools provide security personnel with an important means to inspect such data characteristics. Before initiating the audit cycle on the mission critical data based on the observed (static) characteristics, it is thus very important to make sure that the data to guard against security breaches are *cleaned*. In that respect, the setup for the data audit system also provides for a comprehensive model to identify data that needs to be cleaned. It is important to note that for determining security threats to the data and insider misuse in particular, it is crucial that the data to be monitored are in a “consistent state” and that the characteristics obtained from the data reflect what is expected from the data. Otherwise any data monitoring or auditing approach would operate on incorrect data (characteristics) and thus would probably result in a high number of false positives indicating a security breach.

The properties of the data analyzed are specified in the form of *data profiles* that describe security policies. With each profile, a set of so-called *guard bands* is associated. Guard bands basically describe the admissible behavior of the data (groups) and can be either user-specified or derived during the learn and guard cycle discussed below. For example, a guard band can specify that the minimum value of an attribute in a group of tuples satisfying a grouping criterion may not decrease by more than 5% and not increase by more than 10%. With a guard band an action is associated that is taken once a violation of the policy has been determined during subsequent guard cycles. It is interesting to note that such profiles, including guard bands and actions, can be considered as *soft integrity constraints* on (groups of) data. Unlike hard integrity constraints, which typically specify semantic,

domain, or foreign key constraints on relations, a violation of a soft constraint does not cause an abort of the violating transaction but a user specified action, e.g., an email notification sent to the security personnel.

Once data profiles and guard bands have been specified, the guard and learn cycle starts. During this cycle, modifications to the mission critical data are (periodically) monitored and the modified data (inserted, updated, and deleted tuples) are compared to the previously determined characteristics of the data. There are three important key issues in this cycle. First, the period to monitor or check the data of concern needs to be determined. Different approaches, such as triggers or audit logs, can be employed for this and depend on the impact the data audit system is allowed to have on the performance of the production database. The different approaches supported by the audit systems and their performance impact is discussed in Section 5. Second, the newly computed data characteristics need to be efficiently compared against the data profiles and guard bands previously determined. Data profiles and information about guard bands and actions are kept separately from the production database, not only for security reasons but also for performance reasons. Third and most importantly, data values (attribute values, number of tuples in a group, data distribution etc.) can evolve over time. Thus, guard bands need to be adjusted appropriately to account for data behavior that lies within specified boundaries. For example, if the maximum value of an attribute gradually increases over time within a specified guard band, this boundary value needs to be updated in the data profile. While the guard component of the cycle is responsible for comparing data characteristics, the learn component of the cycle is responsible for adjusting profiles and guard bands. Different statistical analysis modules are employed to distinguish between anomalous data behavior violating (dynamic) band conditions, and dynamics of the data within the specified or recently learned boundaries.

In summary, the data audit system can be understood as a “data centric” anomaly detection system that (1) adjusts automatically to the dynamics of the data characteristics if these characteristics are within learned or specified boundaries, and (2) performs user specified actions in case newly discovered characteristics of the data violate known properties of the data, as specified in the data profiles. During the learn and guard cycle, boundaries and monitoring periods can be adjusted by the user, depending on what has recently been learned about the data and how the workload on the data changes over time.

3. METADATA AUDIT SYSTEM

The data audit system exclusively focuses on detecting anomalous behavior of mission critical data customer or business applications operate on. Although this system builds an important component of the overall architecture, it only partially provides for detecting and responding to security threats and breaches. Administrative personnel such as database administrators and application developers (representing classes of insiders) typically possess a wide variety of different privileges on a production database.

The function of the metadata audit system is to analyze and detect possible security threats and breaches that can arise by administrative personnel operating on the database. The metadata audit system is exclusively concerned with behavioral aspects of the *metadata*. In this context, metadata is understood as all data about the physical and logical layout of the production database, database schemas, stored procedures, and in particular roles, access privileges, and resources. Such metadata is typically managed within the data dictionary of a database and requires special access privileges in order to manipulate the data (typically through data definition language statements). The goal of the metadata audit system is to monitor and analyze the behavior of metadata that relate to possible security threats. For example, a database administrator might –accidentally or maliciously – assign database privileges to a user who then can exploit these privileges to gain access to data he could not access before. A similar scenario occurs when the administrator revokes an access privilege from an application or user, resulting in data being not available to the application. Independent of whether granting such an access was intended or not, observing the behavior of the metadata that record access privileges is helpful in this case. It can detect anomalous events but it can also help the administrator or developer to point out possible vulnerabilities he or she introduced by performing administrative tasks on the database.

The configuration of the metadata management system heavily depends on the underlying production database since different commercial database management systems (DBMS) have different names for data dictionary views and tables and they also differ in terms of what metadata is provided in the data dictionary. Although today's commercial DBMS often have several hundred data dictionary views and tables, only a fraction of these turn out to be relevant for the metadata audit system. These include in particular data dictionary components maintaining information about relations (attributes, data types, integrity constraints), roles (object and

system privileges as well as other roles comprising a role), users (user roles and privileges, profiles, and resource restrictions), audit configurations and general resource management (e.g., buffer sizes, query timeouts etc). Similar to the setup of the data audit system, the security personnel selects respective data dictionary views that are considered to be mission critical. Depending on the underlying DBMS, the metadata audit system provides the security personnel with the respective data dictionary views and tables. Tuples and attribute values in these relations are analyzed and the metadata characteristics obtained by this process are visualized. Similar to the visualization of properties of data, useful information can be obtained from representing metadata characteristics. A typical example is the discovery of too many or untypical access privileges to a role or user, indicating a violation of the principle of minimal access privileges. In such cases, the metadata need to be “cleaned”, meaning that, e.g., access privileges are adjusted to what is currently known about necessary and sufficient privileges for a user or role to operate on the database. Only after such a cleaning, metadata profiles are determined that specify the typical characteristics (signature) of the metadata of interest.

With each metadata profile, the security personnel associates guard bands. Clearly, the metadata of concern are less dynamic than the data utilized in customer and business applications. Consequently, not only the guard bands are tighter but also their evolution is much more restrictive. Respective aspects are taken into account when the guard and learn cycle for analyzing and reacting to the behavior of the selected metadata is initiated. In particular, monitoring metadata should preferably occur in real time since anomalous behavior of the metadata can have a much more severe impact on the data managed by the database. We will illustrate the monitoring approach adopted for the metadata audit system in Section 5.

4. ACCESS AUDIT SYSTEM

The audit systems introduced in the previous sections provide security personnel with effective means to associate data anomaly detection mechanisms with a production database. In particular, these systems appropriately address the aspects of data integrity and availability. They do not, however, fully address the aspect of data confidentiality. A typical case where data confidentiality is violated is when a user accesses data he is not supposed to access, although he is authorized to access the data. Even though such cases could be caught during the configuration of the metadata audit system by carefully analyzing what user or application has what access

privileges to mission critical data, this approach is not sufficient. Ideally one would like to have a system that analyzes what user or application operates on what portions of the mission critical data. That is, the system should be able to identify what users contribute to the observed behavior of the data. The behavior does not only include insertions, updates, and deletions of tuples, but also select operations on the data (using the SQL select statement). While data modification can be observed through periodically querying the data, monitoring select operations on data (including their selectivity, i.e., number of result tuples and selection conditions) requires a different mechanism.

It is the purpose of the access audit system to allow security personnel the specification of access monitoring mechanism that observe the access behavior of users on the data and indicate anomalous access behavior. This system thus establishes a connection from the data centric audit systems described in the previous sections to a user or application centric view of security concerns. Compared to other users profiling approaches that have been proposed in the literature (e.g., [5,10]), the initial focus of the system is not a user or group of users but the data users can operate on. The user profiling approach thus is data driven in that first the characteristics of data and their behavior are analyzed and monitored, and then the link to users and applications contributing to the observed (and perhaps abnormal) behavior is established.

In the current version of the access audit system, security personnel has to choose a time window in which accesses to a selected mission critical relation are monitored. Information about database users who are allowed perform operations on the selected relation is obtained from the data dictionary. Information about access pattern during the chosen time window are grouped by user and access pattern. As with other approaches to user profiling, the size of the time window during which information about access patterns is collected is crucial. If it is too large or too small, the information obtained is not meaningful and thus meaningful access patterns cannot be established in order to instantiate appropriate mechanisms that guard against deviations from these access patterns. For the access audit system, it is currently required that security personnel are involved in the learn cycle and appropriately adjust time windows for monitoring access patterns.

The effectiveness of the access audit system to analyze data access behavior and to identify possible access misuse heavily depends on the concept of "user" employed by the production database. Naturally, without correlating the audit information obtained by the above audit systems with

audit information from applications external to the database (e.g., Web-based applications), it is not possible to actually identify application or operating system users who misuse their privileges. The access audit system only correlates access patterns to database users and not users external to the database, an important functionality that is currently under development. If several applications operate on the database and mission critical relations in particular and one or more database user accounts are shared, then the access patterns obtained are probably not very meaningful. The problem in this case is that the configuration of the database already violates the principle of avoiding shared accounts. Therefore, specific operations to relations cannot be correlated to individual applications. Thus, the access audit system can be considered as a layer that can refine and further analyze information about the behavior of mission critical data.

5. CONCEPTUAL AND ARCHITECTURAL ASPECTS

The effectiveness of the audit systems described above naturally depends on the resources and functionality the underlying production database is able to provide. If real-time detection of anomalous data and accesses is the core objective, real-time data content monitoring and analysis is essential. Initially, it might also be sufficient to periodically check for anomalous data and access behavior and to respond retroactively. In order to provide security personnel with a range of options, the audit systems can employ different mechanisms for content monitoring. In the following, we describe these mechanisms and their usage in detail, and we also outline the general architecture of the audit systems in the context of a production database.

The biggest concern businesses have regarding security mechanisms is the (1) performance impact these mechanisms have on the underlying production systems, and (2) violation of autonomy of the production system. The latter case occurs if security mechanisms are directly embedded into the production system and are not realized through means external to the system. As discussed in the previous sections, the main objective of the audit systems is to monitor the behavior of the data and accesses to the data. In the context of databases, several methods and mechanisms exist to realize this objective:

- periodic querying of relations containing mission critical data,
- database triggers that perform specific actions in case an insertion, deletion, or update occurs on a relation containing mission critical data,
- auditing actions on relations using existing database auditing mechanisms.

All types of commercial database management systems (DBMS) provide the above mechanisms. For the data and metadata audit systems, querying critical relations and data dictionary views, respectively, is the first step in determining data characteristics and specifying data profiles. For the subsequent guard and learn cycle, either of the above three mechanisms can be employed. If the major objective is to get a better understanding of the behavior of the data, a periodic querying of the relations of concerns (or subsets thereof) can be sufficient. The time intervals for querying the relations need to be determined based on the workload occurring on these relations. As we will show below, querying of relations occurs from a remote system and thus does not require any modifications to the database. The obvious disadvantage to periodically querying relations and data dictionary views is that a misuse can occur between two points in time in which the data is queried and thus potential security breaches might go undetected. Furthermore, simply querying relations is inappropriate in order to establish a profile for data accesses. That is, there is no way in which accesses to the relations of concerns can be monitored by simply querying these relations. The query approach thus is only an appropriate means if real-time detection of security breaches and access violations are not of concern.

Another approach to monitor changes and accesses to relations and data dictionary views is the usage of database triggers. After data profiles and guard bands have been determined and specified for critical relations through simply querying these relations, database triggers are enabled that react on insertions, deletions, and updates of tuples on these relations. The actions specified in triggers write information about the deleted, inserted, and updated tuples into extra relations that are kept separate from the production database and are used for determining violations of guard bands and learning new guard bands. While the trigger approach requires changes to the production databases, triggers allow for real-time detection of anomalous data behavior and they only require keeping track of changes (deltas) to the mission critical relations. Using triggers, it is also possible to keep track of exact changes to the data. For example, row level triggers can be used to record information about new and old values of attributes that have been modified. Depending on the type of underlying database management system, there are some limitations regarding the usage of triggers. For example, some DBMS do not allow triggers on data dictionary views. In such cases, triggers cannot be used for monitoring anomalous behavior of metadata.

Database auditing together with triggers build the most complete approach for detecting anomalous data and access behavior in real-time.

Since typically only a few relations and data dictionary views need to be audited, the performance impact of enabling database auditing on a production database is minor. There are several advantages to using database auditing. First, auditing provides fine-grained information about relations and data dictionary views in terms of accesses and modifications. That is, in most DBMS, an audit trail records not only when and how a relation has been modified or accessed, it also records detailed information about who executed the access and from where the access occurred (e.g., user id, terminal id, and session information). Second, the audit trail can be kept separately from the production database. That is, the analysis of the audit trail can be done efficiently on a remote system that manages data profiles and guard bands. Since often audit mechanisms record only information about the type of access and not values of tuples being modified, inserted, or deleted, row level triggers on respective relations provide an appropriate means to collect all necessary information about data and access behavior.

In summary, three different types of monitoring mechanisms can be employed. Simple queries are always used to establish data profiles and can also be used to periodically monitor changes to mission critical relations. Depending on the security needs and ability to add mechanisms to the production database, triggers and audit trails can be used to record and analyze changes and accesses to data in real-time. In all cases, the collected information about data and access behavior, data profiles, guard bands, and actions to execute in case of a detected security breach is kept separately from the production database. Figure 1 illustrates the conceptual architecture of the audit systems. Different types of applications are assumed to operate on the production database. The audit system consists of two components: (1) a local database used to manage information about critical relations and data dictionary views, including data profiles, guard bands, notification mechanisms, etc. (2) an analysis component that analyses information about changes to critical relations using the information recorded in the local databases. A simple JDBC connection to the production database is used to initially establish data profiles and, in case the period querying approach is used, to periodically obtain information about changes to relations. In case triggers and/or auditing is enabled in the production databases, information collected by these mechanisms is recorded in the database local to the audit systems.

Figure 1: Coupling of audit and analysis systems with production database

Since the audit and analysis systems are supposed to work with a variety of commercial DBMS, several DBMS specific libraries are available. In these libraries, information about data dictionary structures and names, audit and trigger mechanisms (for automatic trigger generation), and database connectivity information is maintained. On top of the monitoring and analysis system, several tools are available for security personnel to configure and interact with the audit and analysis systems. This includes a data visualization tool for visualizing properties of data in mission critical relations (data distribution, min/max/average values per data groups etc.), a tool to specify guard bands and actions to be taken in case anomalous behavior has been detected, and different report generation tools. The monitoring and analysis components as well as the local database are hosted on a secure platform to avoid that malicious users tamper with the audit systems and collected information about data and access behavior.

6. CONCLUSIONS

In this paper, we have presented the components and functionality of a data content monitoring system that provides security personnel with flexible means to specify, analyze, and react on anomalous data and access behavior in database systems. The proposed system can be used in different contexts, ranging from monitoring mission critical relations to monitoring metadata related to schema and privilege changes. The monitoring mechanisms can be adjusted based on the availability of resources and mechanisms in the underlying production database. The data, metadata, and

access audit systems presented in this paper provide security and IT personnel with an additional layer of mechanisms that guard mission critical data against common security threats, with a particular focus on insider misuse. Currently, several statistical modules are developed and tested in order to provide for accurate and realistic data profiles and results during learn cycles. As a long term goal, the system should not only serve as a database intrusion detection system but also as a tool to detect vulnerabilities in production databases and to suggest repairs of such vulnerabilities in the form of, e.g., modified access privileges and usage of database roles.

REFERENCES

- [1] Ant Allan: Intrusion Detection Systems (IDSs): Perspective. Gartner Research Report DPRO-95367, Technical Overview, January 2002.
- [2] Robert Anderson: RAND Corporation. Research and Development Initiatives Focused on Preventing, Detecting, and Responding to Insider Misuse of Critical Information Systems. Conference Proceedings CF-151-OSD, 1999.
- [3] Silvana Castano, Mariagrazia Fugini, Giacarlo Martella, Pierangela Samarati: *Database Security*, Addison-Wesley, 1995.
- [4] Dorothy E. Denning: An Intrusion-Detection Model. *IEEE Transactions on Software Engineering* 13(2):222-232, 1987.
- [5] Tom Fawcett, Foster J. Provost: Combining Data Mining and Machine Learning for Effective User Profiling. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 1996)*, 8-13, AAAI Press, 1996.
- [6] Wenke Lee, Salvatore J. Stolfo: A Framework for Constructing Features and Models for Intrusion Detection Systems. *ACM Transactions on Information and System Security* 3(4):227-261, 2000.
- [7] John McHugh: Intrusion and Intrusion Detection, *International Journal of Information Security*, 1(1):14-35, 2001.
- [8] Peter G. Neumann: The Challenges of Insider Misuse. Prepared for the *Workshop on Preventing, Detecting, and Responding to Malicious Insider Misuse* 16-18 August 1999, at RAND, Santa Monica, CA, <http://www.csl.sri.com/users/neumann/pgn-misuse.html>
- [9] Richard Power: 2002 CSI/FBI Computer Crime and Security Survey. *Computer Security Issues & Trends*, Vol. 8, No. 1, Spring 2002, Computer Security Institute, 2002.
- [10] Dit-Yan Yeung, Yuxin Ding: User Profiling for Intrusion Detection Using Dynamic and Static Behavioral Models. In *Advances in Knowledge Discovery and Data Mining, 6th Pacific-Asia Conference (PAKDD 2002)*, 494-505, LNCS 2336, Springer, 2002.