Application-Level Misuse Detection in Relational DBMS

Christina Chung, Michael Gertz, Karl Levitt
University of California, Davis
{chungy|gertz|levitt}@cs.ucdavis.edu

INTRODUCTION

The security goals of a computer system can be specified explicitly by *security policies* - what actions a subject can or cannot perform on objects under specified conditions. The security policies are enforced by *security mechanisms.* Compromising the security mechanisms is *intrusion* whereas violating the security policies without compromising the security mechanisms by legitimate users is insider abuse. Misuse includes both intrusion and insider abuse.

Misuse can been handled by *signature based* and *anomaly based* approaches [CFMS95]. In signature based detection, audit logs are matched against a database of known attack patterns. In anomaly based detection, it is assumed that the set of intrusive and misuse events equals the set of anomalous events and such anomalies can be detected in the audit records. *Profiles* are derived from audit logs and policies to characterize the normal system usage by users. If the new audit records deviate from the profiles, alarm is raised.

We are interested in detecting misuse in relational database systems at the application level using techniques from both approaches. Our research is motivated by the fact that the majority of applications running on computer and network systems nowadays are information systems where the valuable data stored is vulnerable to various security threats. However, previous misuse detection systems reside at the system or network layer ([CFMS95]). Audits at this level are not suited for misuse detection at the information level because the semantics of the applications are not reflected in the low-level audit logs. Unlike these systems, our approach focuses on the *application layer* which exploits the database schema underlying the applications. That is, the relation schemas (and attributes), relationship between attributes and the intergrity constraints.

SYSTEM ARCHITECTURE

System Description We use the relational model as the underlying data model for the database schema. We take into consideration both the *structure* and *semantics* of the database application. This is achieved by defining a set of aggregate features which are abstract concepts derived from primitive features. Primitive features are features that are directly measurable by the auditing system. Examples are login/logout time of the users, access frequencies of attributes of the schema, delta values for updates on attributes of tables, or the time of occurrence of the events. Aggregate features are defined in terms of primitive features and the database schema. For example, the number of times attributesA, B are accessed, $Freq(A), Freq(B)$, are primitive features. Suppose the schema gives us a measure, $Dist(A, B)$, of how far away these attributes are in the corresponding entity-relationship diagram. We can define an aggregated feature *shortest semantic distance* between A, B as $\frac{Freq(A)-Freq(B)}{|max(Freq(A),Freq(B))|}Dist(A, B)$.

Components The proposed detection system is located between the DBMS and the database application. It consists of the following components: (1) a data collection and processing module, (2) a security policy specification module, (3) a misuse detection module.

Data Collection and Processing Module In this module, the auditing functionality of the DBMS is used to audit the relevant primitive features. The collected audit data are then pre-processed into a format understandable by other modules of the system. This includes handling missing and incorrect data, type mapping, and representation condensing.

Security Policy Specification Module A security policy language will be developed to aid the security officers in specifying security policies and domain knowledge such as known misuse patterns. The language is designed with the following criteria in mind. (1) It should support temporal constructs since time is an important feature. It should be able to express, for instance, the point in time relative to other events, the temporal relationship between two events, or the nth occurrence of an event during an interval. (2) It should be easy to write and understand since the security officers may not be computer experts. (3) It can be efficiently implemented so that the detection system can respond to malicious behavior in a timely manner. (4) It can be easily converted to the knowledge representation used by the misuse detection engine.

The objects of the rule-based language are (1) users or user groups, (2) tables and attributes of the database schema, and (3) events. An event states which user performs what operations on which table and attributes. The operations are operations supported by the data manipulation language, such as select, insert, update, and delete. Events can be primitive events, or composite events (which are disjunctions, conjunctions or sequences of primitive events).

Misuse Detection Module The misuse detection engine accepts audit data from the data collection module from which normal user profiles are determined. Based on the policies specified by the policy specification engine and the user profiles generated, alarm is raised if new audit records do not match the user profiles. We intend to use data mining techniques to discover the profiles for the users. The profiles based on the security policy language are logical, rule-based specifications that may contain temporal constructs, primitive and aggregated features. It is believed that the normal usage patterns form *working scopes*, i.e., clusters of regions the user usually works with. Clustering techniques ([Eve73]) can be used to discover these clusters where the similarity function can be based on the aggregated features. Deviation from the profiles is detected if new audit records fail to be classified into one of the discovered clusters. Other techniques used in knowledge discovery systems ([DF95], [FPSSU96], [PSF91]) can be borrowed to discover the usage patterns from the set of aggregated and primitive features.

FRAMEWORK

The interactions between different modules in our system are depicted as follows:

1. The system description defines a set of aggregated features from the primitive features selected by the data collection module and the given relation schemas of the database application.

2. Given a database schema for a database application, the set of users and the set of operations audited, the data collection module collects the data of the selected primitive features for events in the database system. The audit data are preprocessed (e.g., aggregated) and fed into the misuse detection module.

3. Based on the primitive and aggregated features defined, the security officers specify the domain knowledge and security policies using the security policy specification module. These are converted to a internal knowledge representation understandable by the misuse detection module.

4. The misuse detection module determines the profiles for the users and user groups from the audit data. Anomalies are detected based on the profiles and the security policies specified.

FUTURE WORK AND CONCLUSION

An extension of the system would be a response module that can react to malicious activities detected. This includes generating a visual presentation to aid the security officers in identifying source of anomalies and to carry out counter-measures against malicious activities. The system can be adapted to a distributed environment such as a federated database (an integration of heterogeneous databases) over a network. Aggregation of information across individual DBMS can lead to potential improvement in performance. Issues that need to be addressed include resolving structural and semantic conflicts among security policies of individual DBMS and aggregation of data from multiple heterogeneous systems.

We plan to conduct a detailed study of misuse detection in relational database systems using the anomaly based approach. A prototype of the misuse detection module has been implemented and preliminary results show our approach is feasible. Our focus at the application layer is novel and we expect to show improvement in performance over existing misuse detection systems.

References

[CFMS95] Silvana Castano, Maria Grazia Fugini, Giancario Martella, and Pierangela Samarati. *Database Security.* Addison-Wesley, 1995.

[DF95] Karsten M. Decker and Sergio Focardi. Technology overview: A report on data mining. Technical Report CSCR TR-95-02, Swiss Scientific Computing Center, 1995.

[Eve73] Brian Everitt. *Cluster Analysis.* John Wiley & Soons - New York, 1973.

[FPSSU96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. *Advances In Knowledge Discovery And Data Mining.* AAAI Press/The MIT Press, 1996.

[PSF91] Gregory Piatetsky-Shapiro and J. William Frawley. *Knowledge Discovery In*

Databases. AAAI Press / The MIT Press, 1991.