A Framework for Securing Databases from Intrusion Threats

R. Prince Jeyaseelan James
Department of Computer Applications, Valliammai Engineering College
Affiliated to Anna University, Chennai, India
Email: prince_james2000@yahoo.co.in

Abstract- Unauthorized users can make use of the association among data to infer information from a series of data accesses. A violation detection system is suggested in this paper to protect the data content. Based on data dependency, database schema, and inference knowledge, an intrusion detection model is constructed that represents the possible inference channels. When a user submits a query, the detection system inspects the past query records and calculates the probability of inferring information. The query request is denied if the inference probability exceeds the pre-specified limit. In multiuser environment, the query results may be shared among users to increase the inference probability. Based on the query sequences of users, joint inference is evaluated and a model is proposed to prevent multiple users from deriving information.

Index Terms- inference engines; deduction and knowledge processing; security and privacy protection; query processor.

1. INTRODUCTION

Generally access control mechanisms are used to protect users from the disclosure of delicate information in data sources. With the demand for even higher dimensional databases, consisting of hundreds of or more dimensions, earlier security was based on the access control mechanism. Indexing techniques have typically been designed for 30-50 dimensions and fail to improve the performance of sequential scan due to the known “dimensionality curse.”

But such techniques are insufficient since unauthorized users may access a series of information and then employ inference techniques to derive data by using that information.

Managing trust is a problem of particular importance in peer-to-peer environments where one frequently encounters unknown agents. Existing methods for trust management that are based on reputation focus on the semantic properties of the trust model. They do not scale as they either rely on a central database or require to maintain global knowledge at each agent to provide data on earlier interactions.

Address the problem of reputation-based trust management at both the data management and the semantic level by employing at both levels scalable data structures and algorithms that require no central control and allow to assess trust by computing an agent’s reputation from its former interactions with other agents. This method can be implemented in a peer-to-peer environment and scales well for very large numbers of participants. Scalable methods for trust management are an important factor and fully decentralized peer-to-peer systems should become the platform for more serious applications than simple file exchange. The architecture for trust management which relies on all system layers, namely network, storage and trust management, on peer-to-peer mechanisms is shown in Fig. 1.

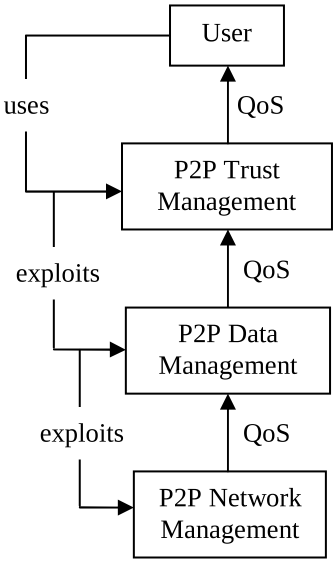


Fig. 1. Different system levels of P2P computing.

A large portion of real-world data is stored in commercial relational database systems. In contrast, most statistical learning methods work only with “flat” data representations. Thus, to apply these methods, it is necessary to convert the data into a flat form, thereby losing much of the relational structure present in the database.

Probabilistic relational models (PRMs) describe how to learn from databases. PRMs allow the properties of an object to depend probabilistically both on other properties of that object and on properties of related objects. Although PRMs are significantly more expressive than standard models, it shows how to extend well-known statistical methods for learning.

Relational models are the most common representation of structured data. Enterprise business information, marketing and sales data, medical records, and scientific datasets are all stored in relational databases. Recently, there has been growing interest in making more sophisticated use of these huge amounts of data, in particular mining these databases for certain patterns and regularities. By explicitly modeling these regularities, we can gain a deeper understanding of the domain and may discover useful relationships. We can also use the model to "fill in" unknown but important information.

To address this inference problem, we develop an intrusion detection system shown in Fig. 2, which resides at the central directory site. Because inference channels can be used to provide a scalable and systematic sound inference, we construct an intrusion detection model that represents all the possible inference channels from any attribute in the system to the set of pre-assigned sensitive attributes.

.

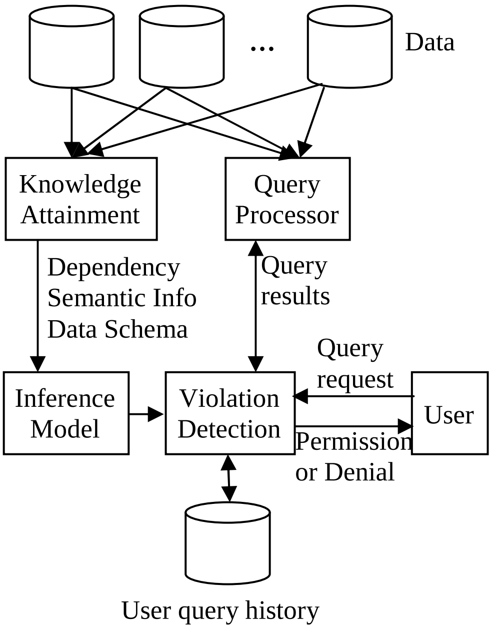


Fig. 2. Intrusion detection system framework.

The intrusion detection model can be constructed by linking all the related attributes, which can be derived via attribute dependency from data dependency, database schema, and semantic related knowledge. Based on the intrusion detection model, the violation detection system keeps track of a user's query history. When a new query is posed, all the channels where sensitive information can be inferred will be identified. If the probability of inferring sensitive information exceeds a pre-specified threshold, then the current query request will be denied.

This intrusion detection approach is based on the assumption that users are isolated and do not share information with one another. But most users usually work as a team, and each member can access the information independently. Afterward, the members may merge their knowledge together and jointly infer the sensitive information. Generalizing from a single-user system to a multi-user system greatly increases the complexity of the intrusion detection system.

2. KNOWLEDGE ATTAINMENT

Since users may pose queries and obtain knowledge from different sources, we need to construct a detection system to track user inference intention. This requires the system to gather knowledge from three entities.

2.1. *Data Dependency*

It represents informal relationships and non-deterministic correlations between attribute values. The dependency between two attributes A and B is represented by the conditional probabilities $p_{i \mid j} = P_r (B = b_i \mid A = a_j)$. There are two types of non-deterministic data dependencies:

2.1.1. Dependency within entity

Let *A* and *B* be two attributes in an entity *E*. If *B* depends on *A*, then for each instance of *E*, the value of attribute *B* depends on the value of attribute *A* with a probability value. The conditional probabilities can be derived via a sequential scan of the table with a counting of the occurrences of A and B and the co-occurrences of A and B.

2.1.2. Dependency between related entities

The parameters of dependency between related entities can be derived by first joining the two entity tables based on the relation *R* and then scanning and counting the frequency of the occurrences of the attribute pair in the joined table. If two entities have an *m*-to-*n* relationship, then the associative entity table can be used for joining the related entity tables to derive dependency between related entities.

2.2. *Database Schema*

In relational databases, database designers use data definition language to define the data schema. The owners of the entities specify the primary key and foreign key pairs that represent a relationship between two entities. If entity E_1 has primary key *pk*, entity E_2 has foreign key *fk*, and $e_1.pk = e_2.fk$, then the dependency between related entities from attribute *A* (in e_1) to attribute *C* (in e_2) can be derived.

2.3. *Domain-Specific Knowledge*

Domain-specific knowledge among attributes is not defined in the database. However, from a large set of queries presented by the users, we can extract the semantic constraints. Domain-specific relationships among attributes and / or entities can supplement the knowledge of unauthorized users and help their inference. Therefore, it is necessary to capture this knowledge as extra inference channels.

3. INTRUSION DETECTION MODEL

The intrusion detection model combines data schema, dependency, and semantic knowledge. This model links related attributes and entities, as well as semantic knowledge needed for data inference. The related attributes (nodes) are connected by three types of relation links:

(1) *Dependency link*: It connects dependent attributes within the same entity or related entities. The conditional probabilities of the child node, given all of its parents, are summarized into a conditional probability table (CPT) that is attached to the child node.

The conditional probabilities in the CPT can be derived from the database content. The conditional probability P_r $(B = b_i \mid A = a_j)$ can be derived by counting the co-occurrence frequency of events $B = b_i$ and $A = a_j$, and dividing it by the occurrence frequency $A = a_j$.

(2) *Schema link*: It connects an attribute of the primary key to the corresponding attribute of the foreign key in the related entities.

(3) *Semantic link*: It connects attributes with a specific semantic relation. To evaluate the inference introduced by semantic links, we need to compute the CPT for nodes connected by semantic links.

If the semantic relation between the source and the target node is unknown or if the value of the source node is unknown, then the source and target nodes are independent. Thus, the semantic link between them does not help inference.

To represent the case of the unknown semantic relationship, we need to introduce the attribute value “unknown” to the source node and set the value of the source node to “unknown.” In this case, the source and target nodes are independent.

When the semantic relationship is known, the conditional probability of the target node is updated according to the semantic relationship and the value of the source node. If the value of the source node and the semantic relation are known, then the inference probability can be derived from the specific semantic relationship.

3.1. *Evaluating Inference*

From the intrusion detection model, there are many feasible inference channels that can be formed via linking the set of dependent attributes. Therefore, we propose to map the model to a Bayesian network to reduce the computational complexity in evaluating the user inference probability for the sensitive attributes.

For any given node in a Bayesian network, if the value of its parent node(s) is known, then the node is independent of all its non-descending nodes in the network. This independence greatly reduces the complexity in computing the joint probability of nodes in the network.

Furthermore since all attribute nodes in the Bayesian network need to be reevaluated, after posing each query, the time required for inference evaluation is almost constant.

4. VIOLATION DETECTION

For individual users, the intrusion detection model provides an integrated view of the relationships among data attributes, which can be used to detect inference violation for sensitive nodes. Here the values of the attributes are set according to the answers of the previous posted queries. Based on the list of queries and the user who posted those queries, the value of the inference will be modified accordingly. If the current query answer can infer the sensitive information greater than the pre-specified limit, then the request for accessing the query answer will be denied.

Generalizing from the single-user system to the multi-user system greatly increases the complexity and presents two challenges for building the intrusion detection system.

(1) Estimating the effectiveness among the users that involves factors such as authoritativeness, communication mode and honesty.

(2) Integrating the knowledge from the users on the inference channels for the inference probability computation.

For any two users, we can integrate one's knowledge to the other and detect their inference toward sensitive data. When any user poses a query, the system not only checks if the query requester can infer sensitive data above the specified limit with a query answer but also checks the other team members to guarantee that the query answer will not indirectly let them infer the sensitive attribute.

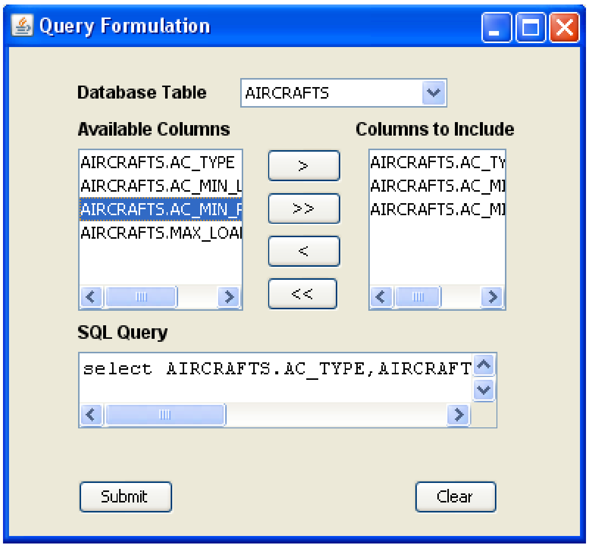
5. IMPLEMENTATION RESULTS

Database intrusions have to address problems of providing protection to database security in spite of the already existing access control mechanisms. Typically, for a given database, there is a structural description of the type of facts held in that database: this description is known as a schema. The schema describes the objects that are represented in the database, and the relationships among them. There are a number of different ways of organizing a schema, known as database models (or data models). The model in most common use today is the relational model, which in layman's terms represents all information in the form of multiple related tables each consisting of rows and columns. This model represents relationships by the use of values common to more than one table. Other models such as the hierarchical model and the network model use a more explicit representation of relationships.

The inadequacy of schema-level inference is pointed out, and six types of inference rules from the data level that serve as deterministic inference channels are identified viz. hypothetical syllogism, disjunctive syllogism, constructive dilemma, absorption, simplification, and conjunction. In order to provide a multilevel secure database management system, an inference controller prototype was developed to handle inferences during query processing. Rule-based inference strategies were applied in this prototype to protect the security.

Further, since data update can affect data inference, a mechanism is proposed that propagates update to the user history files to ensure that no query is rejected based on the outdated information. Fig. 3 to Fig. 6 shows the query submitted by the user and whether the query results can be shown or denied according to the level of inference. To reduce the time in examining the entire history login computation inference, a prior knowledge of data dependency is used to reduce the search space of a relation and thus reduce the processing time for inference.

Data inference mainly focused on deterministic inference channels such as functional dependencies. The knowledge is represented as rules, and the rule body exactly determines the rule head. Although such rules are able to derive sound and complete inference, much valuable non-deterministic correlation in data is ignored. Further, many semantic relationships, as well as data mining rules, cannot be specified deterministically. To remedy this shortcoming, a probabilistic inference approach is used to treat the query-time inference detection problem.



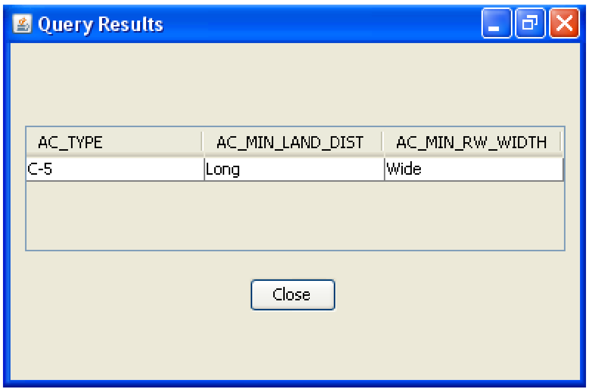
Database Table: AIRCRAFTS

Available Columns:
AIRCRAFTS.AC_TYPE
AIRCRAFTS.AC_MIN_L
AIRCRAFTS.AC_MIN_F
AIRCRAFTS.MAX_LOAI

Columns to Include:
AIRCRAFTS.AC_TY
AIRCRAFTS.AC_MI
AIRCRAFTS.AC_MI

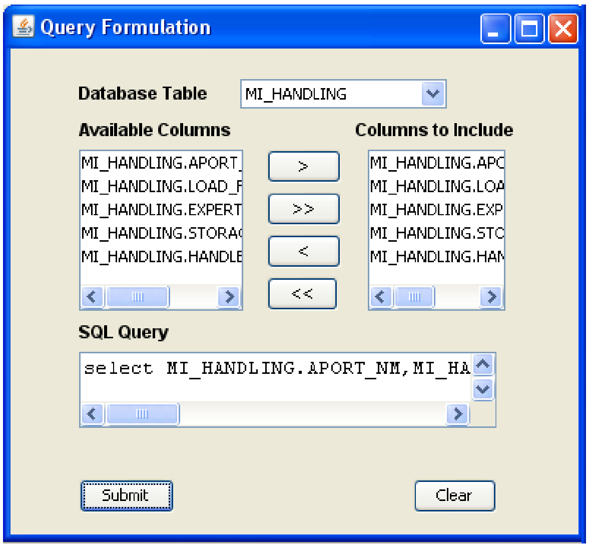
SQL Query:
```
select AIRCRAFTS.AC_TYPE,AIRCRAFT
```

Fig. 3. User submits a query to the database server.



AC_TYPE	AC_MIN_LAND_DIST	AC_MIN_RW_WIDTH
C-5	Long	Wide

Fig. 4. The query result is shown since the inference probability is within the pre-specified limit.



Database Table: MI_HANDLING

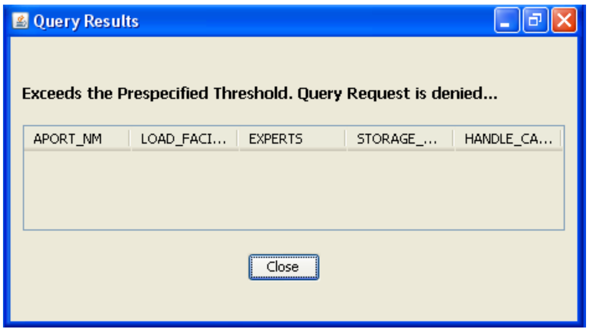
```
MI_HANDLING.APORT_
MI_HANDLING.LOAD_F
MI_HANDLING.EXPERT
MI_HANDLING.STORA(
MI_HANDLING.HANDLE
```

```
MI_HANDLING.APC
MI_HANDLING.LOA
MI_HANDLING.EXP
MI_HANDLING.STC
MI_HANDLING.HAN
```

SQL Query

```
select MI_HANDLING.APORT_NM,MI_HA
```

Fig. 5. User submits a query to the database server.



Exceeds the Prespecified Threshold. Query Request is denied...

APORT_NM	LOAD_FACI...	EXPERTS	STORAGE_...	HANDLE_CA...

Fig. 6. The query request is denied since the inference probability exceeds the pre-specified limit.

6. CONCLUSION

Intrusion detection is the process of extracting possible inference channels from probabilistic data dependency, the database schema, and the semantic knowledge. In this paper, a technique is presented that prevents users from inferring sensitive information from a series of queries submitted by the users. Compared to the deterministic inference approach in previous works, non-deterministic relations into inference channels for query-time inference detection have been included.

For intrusion violation detection, a inference model is developed that combines the users query log sequences into inference channels to derive the inference of sensitive information. A sensitivity analysis of attributes in the Bayesian network can be used for studying the sensitivity of the inference channels. It reveals that the nodes closer to the security node have stronger inference effects on the security node. Thus, a sensitivity analysis of these close nodes can assist domain experts to specify the threshold of the security node to ensure its robustness.

User profiles provide a good starting point for learning the level of sharing among users. However, gathering such information is complicated by the fact that the information may be incomplete and incorrect. In addition, the accuracy of such information is task specific and user-community sensitive. It is also necessary to define the dependency, schema and semantic links in the database appropriately to extract the sensitive attributes involved in the user submitted query. Further research and experiments in generating the training sets to estimate and validate the level of sharing among the users are needed.

REFERENCES

[1] Duma, C; Shahmehri, N; Caronni, G. (2005). *Dynamic Trust Metrics for Peer-to-Peer Systems*, Proc. 16th Int’l Workshop Database and Expert Systems Applications (DEXA ’05), pp. 776-781.

[2] Chavira, M; Allen, D; Darwiche, A. (2005). *Exploiting Evidence in Probabilistic Inference*, Proc. 21st Conf. Uncertainty in Artificial Intelligence (UAI ’05), pp. 112-119.

[3] Chavira, M; Darwiche, A. (2005). *Compiling Bayesian Networks with Local Structure*, Proc. 19th Int’l Joint Conf. Artificial Intelligence (IJCAI ’05), pp. 1306-1312.

[4] Chen, Y; Chu, W. (2006). *Database Security Protection via Inference Detection*, Proc. Third IEEE Int’l Conf. Intelligence and Security Informatics (ISI ’06).

[5] Li, H; Singhal, M. (2007). *Trust Management in Distributed Systems*, Computer, vol. 40, no. 2, pp. 45-53.

[6] Marti, S; Garcia-Molina, H. (2006). *Taxonomy of Trust: Categorizing P2P Reputation Systems*, Computer Networks, vol. 50, no. 4, pp. 472-484.

[7] Winsborough, W; Li, N. (2004). *Safety in Automated Trust Negotiation*, Proc. IEEE Symp. Security and Privacy (SP ’04), pp. 147-160.