Statistical and Fuzzy Approach for Database Security

Gang Lu, Junkai Yi
School of Information Science and Technology
Beijing University of Chemical Technology
Beijing 100029, China
sizheng@126.com
Kevin Lü
Brunel University, Uxbridge UB8 3PH, UK

Abstract

A new type of database anomaly is described by addressing the concept of Cumulated Anomaly in this paper. Dubiety-Determining Model (DDM), which is a detection model basing on statistical and fuzzy set theories for Cumulated Anomaly, is proposed. DDM can measure the dubiety degree of each database transaction quantitatively. Software system architecture to support the DDM for monitoring database transactions is designed. We also implemented the system and tested it. Our experimental results show that the DDM method is feasible and effective.

1. Introduction

The number of security-breaking attempts originated inside an organization is increasing steadily [5][8]. These attacks are usually made by "authorized" users of the system. Typically, in one type of intrusion, an attacker who is authorized to modify data records under certain constraints deliberately hides his intentions to change data beyond constraints in different operations and different transactions. Often, in this type of attack, each individual transaction is legitimate; however, the accumulated results of the attacker's operations are malicious.

The existing *Intrusion Detection Systems* (IDS) can be grouped into two classes: (1) *misuse detection*, which maintains a database of known intrusion techniques or behaviors and detects intrusions by comparing users' behaviors against the database [7][8]; (2) *anomaly detection*, which analyzes user behaviors and the statistics of a process in a normal situation, and checks whether the system is being used in a different manner [3][9].

In general, *misuse detection* model cannot detect new, unknown intrusions [7]. *Anomaly detection* needs to maintain the records of users' behaviors and the statistics for normal usages, which is referred to as "profiles". The profiles tend to be large. That makes detecting intrusion needs a large amount of system resources, and delays detection decision makings. If attackers hide their operations into other places, *anomaly detection* may not even be able to detect them. It is fair to say that neither *anomaly detection* nor *misuse detection* would be able to effectively detect *Cumulated Anomaly.* New techniques need to be investigated.

In this study, we investigate *Cumulated Anomaly* and propose a model for detection. In this model, the detection rules are set up manually based on the statistical properties of intrusions amongst the normal transactions. In addition, membership functions [5] in fuzzy set theory, with their parameters specified into the detection rules, are applied in the model to monitor and present the possibility of intrusions in real time. Membership functions assist detection rules to indicate the likelihood of a transaction being intrusive. If a transaction is identified by a detection rule as a "possible" intrusion, it is said that the rule "matches" the transaction. An indicator (degree) within the interval $[0, 1]$ will be calculated. This indicator is used to represent the dubiety degree of a transaction. Therefore, this model is named as *Dubiety-Determining Model* (DDM). In this method, the dubiety of various types of database transactions can be quantitatively denoted in a unified form way. By showing the dubiety degrees of database transactions, the model can detect possible anomalies if their dubiety degrees are high.

The rest of the paper is as follows. Section 2 reviews some related work briefly. Section 3 describes the DDM method. Design and implementation issues are discussed in Section 4. In Section 5, the experimental results are introduced. Section 6 is the conclusion.

2. Related work

The characteristics of widespread used databases with the invaluable data held in them make it vital to detect any intrusion or intrusion attempts made at the databases. Therefore, basing on the development of intrusion detections on computer systems, intrusion detection for databases is becoming imperative needs. Besides access policies, roles, administration procedures, physical security, security models, and data inference, *misuse detection* and *anomaly detection* at databases have been focused on. Christina Yip Chung, Michael Gertz and Karl Levitt developed DEMIDS, which is a misuse detection system for database systems tailored to relational database systems [2]. Francesco M. Malvestuto, Mauro Mezzini and Marina Moscarini propose an approach to avoid releasing summary statistics that could lead to the disclosure of confidential individual data in [4]. In [8] and [10], Sin Yeung Lee, Wai Lup Low and Pei Yuen Wong describe an algorithm that summarizes the raw transactional SQL queries into compact regular expressions. All of them have pointed out that the content of transactions can be used to abstract the users' profiles, which will be used during *misuse detection* or *anomaly detection*. However, to make the detection results more precise, some quantitative approaches should be employed.

In the existing database intrusion detection researches, fuzzy set theory is mainly used with other theories such as neural network in building profiles for anomaly detection [1][9][11]. For example, [6] uses a fuzzy Adaptive Resonance Theory (ART) and neural network to detect anomaly intrusion of database operations, by monitoring the connection activities to a database.

As a result, we have a motivation of integrating fuzzy set theory and intrusion detection technique to deal with *Cumulated Anomaly* in databases precisely in real time.

3. Dubiety-Determining Model (DDM)

Given a metric for a random variable X and n observations $X_1,\ldots,X_n$, the purpose of the statistical sub-model of X is to determine whether a new observation X_{n+1} is abnormal with respect to the previous observations. The mean avg and the standard deviation $stdev$ of $X_1,\ldots,X_n$ are defined as:

$$avg = \frac{X_1 + X_2 + \cdots + X_n}{n} \tag{1}$$

$$stdev = \sqrt{\frac{\sum_{i=1}^{n}(X_i - avg)^2}{n}} \tag{2}$$

A new observation X_{n+1} is defined to be abnormal if it falls outside a *confidence interval* that is standard deviations from the mean, which is denoted by *CI*:

$$CI = avg \pm dev \tag{3}$$

where $dev = d \times stdev$ with d as a parameter. Note that 0 (or null) occurrences should be included so as not to bias the data. This model can be applied to variant cases such as event counters accumulated over a fixed time interval. Therefore, it would apply for the case of *Cumulated Anomaly*.

Membership functions are used to "measure" the dubiety degrees for each transaction. For each transaction, a value of variable X can be observed. It can be mapped into the interval $[0,1]$ by a membership function. We define 0 means *completely acceptable*, and 1 implies anomaly or *completely unacceptable*. The values between 0 and 1 are called *dubious degree*. In this way, the dubiety of transactions can be denoted in a unified form.

An appropriate membership function is the basis of quantitative analysis on fuzzy attributes and plays a key role in fuzzy mathematics. The most widely used functions include *S*-shaped functions (F_S), *Z*-shaped functions (F_Z) and π-shaped functions (F_π). With *U*-shaped functions (F_U) defined as complementarities of π-shaped functions, as Figure 1 shows. In Figure 1, we assume that $a \le b \le c$. It is straightforward to prove that when $a = b = c$, F_S and F_Z both have only two values which are 0 and 1, while F_π only has 0 and F_U only has 1 as their values. By adjusting the values of *a*, *b* and *c*, the shapes of F_π and F_U can be changed.

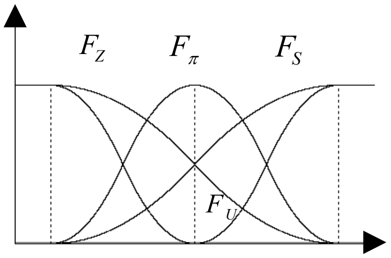

Figure 1. **The curves of the membership functions**

A set P containing n observations $X_1,\ldots,X_n$ of a metric for a random variable X, i.e. $P=\{X_n \mid n=1,2,\ldots\}$, can be obtained. In P, there must be a minimum X_{min} and a maximum X_{max}. The mean of all the elements in P is avg as (1) defines. It is defined that $CI = [X_{min}, X_{max}]$. Thus, by assigning X_{min}, avg and X_{max} to the parameters of membership functions a, b and c, respectively, any observation of a metric for a random variable X can be mapped to a real number in $[0,1]$. This real number denotes the dubiety degree of an observation X_n. The values of X_{min}, avg and X_{max} can be obtained by existing approaches. Because X_{min} and X_{max} are both in CI, $F(X_{min}) < 1$ and $F(X_{max}) < 1$ must stand (meaning X_{min} and X_{max} do not cause anomaly), where $F \in \{F_Z, F_S, F_\pi, F_U\}$. As a result, we have the definition of the four types of membership functions shown in Figure 2. The parameter α can be assigned a proper value by users according to the applications. Nevertheless, it is recommended that α is not less than 1 too much to keep the result values in $(b,c]$ differentiable.

4. Architecture based on DDM

The architecture for database transaction monitoring based on DDM is designed as shown in Figure 3.

The user interface (UI) provides tools for interactions, which includes *Setting Rules* and display *Dubiety-Determining Results*. *Setting Rules* allows users to set up monitoring policies. These monitoring policies are then formatted and transferred into *Detection Rules Base* by *Mapping to Rules*. The information about each database transaction is organized into *Audits Base* by *Sensor*. *Event Analyzing* selects every new audit record from *Audits Base*, and then checks against the detection rules in *Detection Rules Base*. Finally, *Event Analyzing* calculates dubiety degree for the audit record, and forwards the results to *Dubiety-Determining Result*.

Other main components of the architecture are:

Audits Base is built to store the audit records generated by *Sensor*, while *Detection Rules Base* is used to store detection rules defined manually.

Setting Rules, used to define detection rules, specifies which attributes of transactions to monitor, what types of membership functions to use, and what the values of the parameters in membership functions are, etc.

Mapping to Rules. When the information of the monitoring policy and membership function is decided, *Mapping to Rules* translates it into the format of detection rules to store in *Detection Rules Base*.

Sensor. This module monitors the transactions of application databases in real time. By analyzing each transaction processed, it collects information about the transaction, and then stores it in *Audits Base*.

Event Analyzing. This is the centre of the whole architecture. The monitoring algorithm is implemented in this module. For each record in *Audits Base*, *Event Analyzing Module* is processed and matched against the rules in *Rules Base*. The value of the monitored attribute is then obtained. By substituting this value in the membership function defined in the rule, the result of the function is calculated as the degree of dubiety.

There are two basic data structures required in DDM: *Audit Record* and *Detection Rule*. *Audit Record* is for recording the information about each database transaction. *Detection Rule* is the structure for specifying the format of the detection rules. The details of the two structures are defined as follows.

Audit Record. This data structure is 6-tuple recording information of each database transaction:

<AID, UID, SQLText, Time_stampe, Data1, Data2>

where

$$F_S(x,a,b,c) = \begin{cases} 0 & x \le a \\ \frac{1}{2}\left(\frac{x-a}{b-a}\right)^2 & a < x \le b \\ \frac{\alpha+1}{2} - \frac{\alpha}{2}\left(\frac{c-x}{c-b}\right)^2 & b < x \le c, 0 < \alpha < 1 \\ 1 & x > c \end{cases}$$

$$F_Z(x,a,b,c) = 1 - F_S(x,a,b,c)$$

$$F_\pi(x,a,b,c) = \begin{cases} F_S(x,a,\frac{a+b}{2},b) & x \le b \\ F_Z(x,b,\frac{b+c}{2},c) & x > b \end{cases}$$

$$F_U(x,a,b,c) = 1 - F_\pi(x,a,b,c)$$

Figure 2. **The definitions of the membership functions**

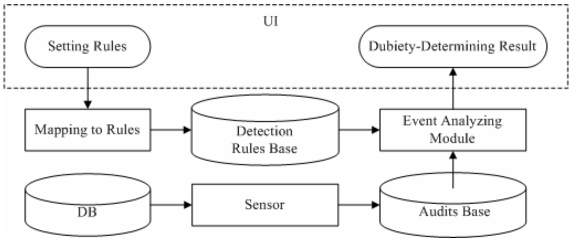

Figure 3. **The architecture for database transaction monitoring based on DDM**

- *AID* is the identifier for each audit record.
- *UID* records the user name of the transaction.
- *SQLText* records the content of the SQL statement of the transaction.
- *Time_stamp* records the time when the transaction is executed.
- *Data1* is the first data field that the transaction relates to. For example, the data value before update.
- *Data2* is the second data field that the transaction relates to. For example, the data value after an update.

To make it clearer, from now on in this paper, we will use the term *audit record* instead of *transaction*.

Detection Rule. This data structure is 6-tuple defining the format of the detection rules:

<RID, UID, Action, Obj1, Obj2, Condition,
Time_window, Mon_type, Function, Enable>

where

- *RID* starting with the letter *R* is the identifier for each detection rule.
- *UID* indicates which user the rule is aimed at.
- *Action* indicates what type of operations the rule is related to, such as *select, update, delete* and so on.
- *Obj1* and *Obj2* records for which database object (table, view, procedure, and so on) the rule is valid. *Obj1* is the first object that *Action* refers to, such as a table, a view or a procedure. *Obj2* is the second one. If *Obj1* is a table or a view, *Obj2* will be a field name.
- *Condition* indicates the condition of *Action*. Usually it is the condition part (*where* clause) of the SQL statement.
- *Time_window* specifies a number of hours as a time range. The audit records occurred in that time range before the currently being checked one will be sought by the rule.
- *Mon_type* is the type of monitor. It has two values: *C* and *S*. *C* is used for counting numbers and *S* is for recording the sum value.

Function is sub-tuple recording the information of the membership function used by the rule:

<FID, A, B, C>

where

- ■ *FID* specifies which type of membership function to use. It has four values. 'Z' means F_Z. 'S' means F_S. 'P' means F_π, while 'U' means F_U.
- ■ *A*, *B*, and *C* store the values of *a*, *b*, and *c* respectively (definition of membership function).

Enable is a switch. When it is 1, the rule is valid; otherwise, it is not.

5. Experimental results

The experiments are performed on the DBMS of Microsoft SQL Server 2000 on Microsoft Windows Server 2003 SP1, to show whether DDM can discover *Cumulated Anomaly* behaviors. The example database *Northwind* of SQL Server is used in this study. The table *Products* in it stores product-related data, including *ProductID* and *UnitPrice*. Suppose there is a product whose *ProductID* is 9 in *Products*. Assume a member of staff, *Ann,* is authorized to modify *UnitPrice* of *Product* 9. However, if the *UnitPrice* has been changed too much or too often, it could be suspicious. It is defined that *UnitPrice* should not be changed for more than 4 times in 30 days, and the sum of changed value should not be more than 3 pounds in 90 days. *Audits Base* and *Detection Rules Base* are built according to the two basic structures defined.

Data. 30000 normal audit records are stored in the database. Their schema is described in Section 4. They include Time_stamps (system clock) in a period of three months. The values of fields SQLText are common database operations in the form of SQL statements, including selecting data from a table, updating the data in a table, inserting data into or deleting data from a table, executing a procedure, and opening a database. Referring to the above assumptions, 12 additional audit records for *Ann*'s updating *UnitPrice* of *Products 9* are constructed and mixed into the existing 30000 audit records. These 12 records are distributed into the range of three months.

The *Detection Rules Base* (described in Section 4) contains two typical detection rules listed in Table 1 (in which the column of Enable is not listed to make the table not too wide). For example, R02 is used to monitor the audit records with *Ann* as UID, *update [Products] set UnitPrice=p where ProductID=9* as SQLText (where *p* is a number). The data items before and after update operation are recorded in the fields Data1 and Data2. When an audit record R which meets the demand of R02 occurs, the algorithm seeks the audit records meeting the demand of R02 which have occurred 2160 hours before R, and sums up the margins between each pair of Data1 and Data2 in each of them. Then, the summation is substituted into F_U defined in R02. Finally, a result value of the function is calculated as the dubiety degree of that audit record. As this is a real-time process; an audit record will be examined as soon as it arrives.

Table 1. **The two detection rules**

RID	UID	ACTION	Obj1	Obj2	CONDITION	TIME_WINDOW	MON_TYPE	FID	A	B	C
R01	Ann	update	Products	UnitPrice	ProductID=9	720	C	S	0	3	5
R02	Ann	update	Products	UnitPrice	ProductID=9	2160	S	U	-3.0	0	3

It can be seen from Table 1 that the two rules are both designed to monitor *Ann*'s operations of updating *UnitPrice* of *Products 9*. *R01* monitors the number of occurrences of the operation over 30 days (720 hours), while *R02* monitors the accumulated values modified over 90 days (2160 hours).

Results. In this experiment, we let $\alpha = 0.9$. As a result, $F_S(X_{max}) = F_S(c) = 0.95$. The experiment contains three tests. In Test 1 only *R01 is* enabled. In Test 2 only *R02* is enabled. Both *R01* and *R02* are enabled in Test 3 to show the combined results. Figure 4 shows all results. Figure 4 (a) shows the value of *UnitPrice* after *Ann* updates it for each time. Figure 4 (b) shows the monitor result of using the rule of *R01*. We can see that the dubiety degree is increasing gradually. However, it does not reach 1 all the while. That means no anomaly occurs by *R01*. Figure 4 (c) shows the results of monitoring the modified *UnitPrice* of *Product 9* over 90 days by *R02*. It is shown that the dubiety degree is more and more close to 1. At the end the dubiety degree reaches 1. According to the definition of DDM, anomalies may occur. When *R01* and *R02* are both enabled in Test 3, the results are shown in Figure 4 (d). Figure 4 (d) also can be regarded as the combinations of Figure 4 (b) and Figure 4 (c) by selecting the point with the higher dubiety degree value between (b) and (c) for each *AID*. In general, when several detection rules are matched to the same audit record, the highest value of dubiety degree amongst these rules will be selected. From the results, we can see *Ann*'s operations cause anomaly.

6. Conclusion

A new type of database anomaly *Cumulated Anomaly* is investigated. A new detection method *Dubiety-Determining Model* (DDM) has been proposed for it. Based on DDM, architecture for database transaction monitoring is designed and implemented.

Tests have been performed to verify the effectiveness of our novel method. The results suggest that our methods are capable of identifying suspicious user behaviors. We are currently considering developing a method based on DDM for general anomaly detection in databases.

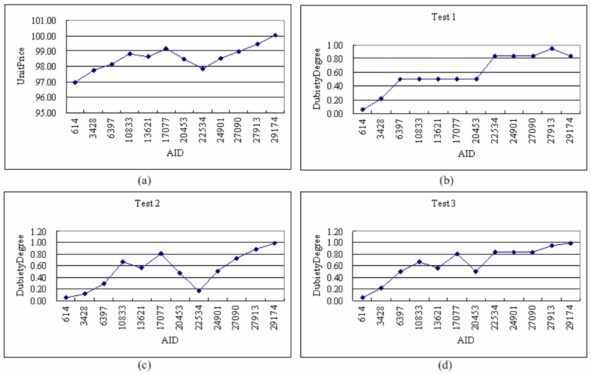

Figure 4. **The result of the experiment**

References

[1] Chan Man Kuok, Ada Fu, Man Hon Wong. Mining fuzzy association rules in databases. SIGMOD Record, 1998, 27(1), 41-46.

[2] Chung C Y,Gertz M, Levitt K. DEMIDS: A Misuse Detection System for Database Systems. In:The Third Annual IFIP TC-11 WG 11.5 Working Conf. on Integrity and Internal Control in Information Systems, 1999

[3] Darren Muts, Fredrik Valeur, Giovanni Vigna. Anomalous system call detection. ACM Transactions on Information and system Seurity, Vol. 9, No. 1, February 2006, 61-93.

[4] Francesco M. Malvestuto, Mauro Mezzini, Marina Moscarini. Auditing sum-queries to make a statistical database secure. ACM Transactions on Information and system Security, Vol. 9, No. 1, February 2006, 31-60.

[5] Pedrycz Witold, Gomide Fernando. An Introduction to Fuzzy Sets: Analysis and Design. Cambridge, Mass. MIT Press, 1998.

[6] Rung Ching Chen, Cheng Chia Hsieh. An anomaly intrusion detection on database operation by fuzzy ART neural network. Proceedings of ICS 2004. 839-844.

[7] Sato I., Okazaki Y., Goto S.. An improved intrusion detecting method based on process profiling. Transactions of the Information Processing Society of Japan vol.43, no.11: Nov. 2002, 3316-26.

[8] Sin Yeung Lee, Wai Lup Low, Pei Yuen Wong. Learning fingerprints for a database intrusion detection system. ESORICS 2002, LNCS 2502, 264-279.

[10] Tian-Qing Zhu, Ping Xiong. Optimization of membership functions in anomaly detection based on fuzzy data mining. Proceedings of 2005 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 05EX1059): (Vol. 4) 1987-92 Vol. 4, 2005.

[11] Wai Lup Low, Joseph Lee, Peter Teoh. DIDAFIT: Detecting intrusions in databases through fingerprinting transactions. ICEIS 2002.