A Dubiety-Determining based Model for Database Cumulated Anomaly Intrusion

Gang Lu
SIST, BUCT
Beijing 100029, China
sizheng@126.com

Junkai Yi
SIST, BUCT
Beijing 100029, China
yijk@mail.buct.edu.cn

Kevin Lü
Brunel University
Uxbridge UB8 3PH, UK
kevin.lu@brunel.ac.uk

ABSTRACT

In this paper, the concept of Cumulated Anomaly is addressed, which describes a new type of database anomalies. A detection model, Dubiety-Determining Model (DDM), is proposed for it. The DDM can measure the dubiety degree of each database transaction quantitatively. We conducted experiments basing on the DDM. In our experiments, the DDM method calculates a real number for each audit record. That number is called dubiety degree, which indicates the possibility of being anomaly for each transaction. The experimental results demonstrate basic features, the feasibility, and the effectiveness of the method.

Categories and Subject Descriptors

H.2.7 [**Database Management**]: Database Administration – *security, integrity, and protection.*

General Terms

Design, Experimentation, Security

Keywords

Database security, Intrusion detection, Anomaly intrusion

1. INTRODUCTION

The database security is becoming a more and more important issue. The number of security-breaking attempts originated inside the organizations is increasing steadily. This kind of attacks is usually made by "authorized" users of the system. Typically, in one type of intrusions, the attacker, is authorized to change the data in small value under certain constraints, deliberately hides or embeds his/her intensions of change data in different operations and different transactions. The amount of money, for example, is allowed to be changed at each time is limited, but if the attacker can manage to change data values for many times, the sum of the data changed would be large and result would be serious. We called this type of intrusions as *Cumulated Anomaly.*

Issues of developing *Intrusion Detection Systems* (IDS) have been considered to increase the defense capacity of an information system [4]. Intrusion detection is the process of monitoring the events and analyzing them for signs of intrusions. On the basis of the collected information, IDS takes decision about the activity as either normal or intrusive. The existing IDS can be grouped into two classes: (1) *misuse detection*, which maintains a database of known intrusion techniques and detects intrusion by comparing behaviors against the database [4]; (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.

There have been a number of studies on *anomaly detection* and *misuse detection* [1][4] for computer, network systems and database systems. [1] proposes an approach to avoid releasing summary statistics that could lead to the disclosure of confidential individual data. However, they only consider the case that the response variable is of nonnegative real type, and only aimed at intrusions carried out by sum-queries in statistical databases. [5] and [6] introduced studies on an algorithm that summarizes the raw transactional SQL queries into compact regular expressions. This representation can be used to match against incoming database transactions efficiently. In general, *misuse detection* model cannot detect new, unknown intrusions [4]. *Anomaly detection* needs to update the data describing 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 making. Furthermore, *anomaly detection* would normally work long after the anomalies had occurred, which may be too late for many applications. If an attacker hides his intention into various operations which match his profiles, *anomaly detection* even may not be able to detect it. As a result, neither *anomaly detection* nor *misuse detection* on database would be able to detect *Cumulated Anomaly Intrusions.* 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 [2] in fuzzy set theory, with their parameters specified into the detection rules, are applied in the model to monitor and specify 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 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. For example, [3] uses a fuzzy Adaptive Resonance Theory (ART) and neural network to detect anomaly

INFOSCALE 2007, June 6-8, Suzhou, China
Copyright © 2007 ICST 978-1-59593-757-5
DOI 10.4108/infoscale.2007.220

intrusion of database operations. But their method monitors the connection activities to a database, but does not check the content of database transactions.

In this study, membership functions are used to carry out the precise measure of the dubiety degrees of database transactions. During the monitor and detection for *Cumulated Anomaly*, the contents database transactions are examined. By this method, the dubiety of various types of database transactions can be denoted in a unified form way quantitatively. By showing the dubiety degrees of database transactions, the model can detect possible anomalies if their dubiety degrees are high.

The main contributions of this study are: (1) address a specific type of anomalies *Cumulated Anomaly*; (2) proposing a method DDM to detect *Cumulated Anomaly*; (3) design a system architecture for database transaction monitoring based on the DDM; (4) the implementation of DDM; (5) experimental studies to verify the effectiveness and efficiency of the DDM.

The rest of the paper is as follows. Section 2 discusses the DDM method. Design and implementation issues are discussed in Section 3. In Section 4, the experimental results are introduced. Section 5 is the conclusion.

2. THE DUBIETY-DETERMINING MODEL

Given a metric for a random variable X and n observations $X_1,..,X_n$, 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 $X_1,..,X_n$ are defined as:

$$avg = \frac{X_1 + X_2 + \cdots + X_n}{n} \qquad (1)$$

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

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*:

$$CI = avg \pm dev \qquad (3)$$

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*.

We use membership functions 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.

A set P containing n observations $X_1,..,X_n$ of a metric for a random variable X, i.e. $P = \{X_n | n = 1,2,..\}$, 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.

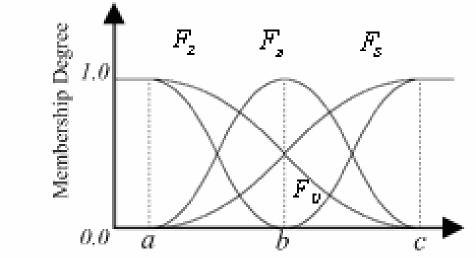

Figure 1. The curves of the membership functions

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

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

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

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

Figure 2. The definitions of the membership functions

Nevertheless, it is recommended that α is not less than 1 too much to keep the result values in $(b,c]$ differentiable.

3. AN ARCHITECTURE BASED ON DDM

3.1 Basic Data Structures

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 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

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 current 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.

3.2 The Architecture

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 Module* selects every new *audit record* from *Audits Base*, and then checks against the detection rules in *Detection Rules Base*. Finally, *Event Analyzing Module* calculates dubiety degree for the audit record, and sends the results to *Dubiety-Determining Result*.

4. EXPERIMENTAL RESULTS

We implement a system based on the architecture introduced in Section 3, which is used to test and verify the DDM method. The experiments are performed on the DBMS is Microsoft SQL Server 2000. The example database *Northwind* of SQL Server is used in this study. It includes trade data records for a company called *Northwind Traders*, engaged in the import and export trade business. *Audits Base* and *Detection Rules Base* are built according to the two basic structures defined. According to Section 3.1, 30,000 typical audit records are generated and 19 detection rules are set up.

*Data***.** AIDs are generated in ascending order of Time_stamp, the values of which are randomly generated precise to second (system clock) in a period of three months (from 2006-07-22 to 2006-10-23). Seven user names appear in the field of UID: *Ann*, *Bob*, *Charles*, *Dennis*, *Eva*, *Fabre*, and *Gama*. The values of fields SQLText are randomly generated as common database operations in the form of SQL statements. The content of SQLText includes selecting data from a table, updating the data in a table, inserting data into or deleting data from a table, executing a procedure, or opening a database.

The *Detection Rules Base* contains 19 typical detection rules. Each rule is specified with one of the four types of membership functions, and the parameters *a*, *b*, and *c* are assigned manually. For instance, as Table 1 shows (in which the column of Enable is not listed to make the table not too wide), we have

R09=< *R09*, *Fabre*, *update*, *order details*, *UnitPrice*,
ProductID=43, *5000*, *S*, < *S*, *5.0*, *10.0*, *32.0* >, 1>.

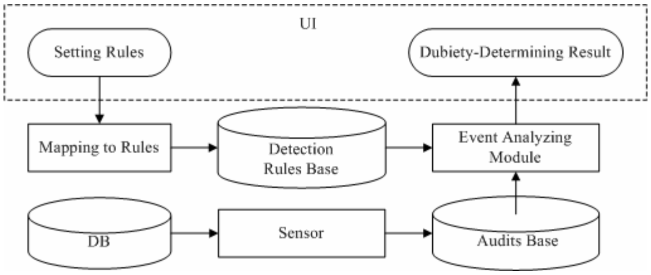

Figure 3. The Architecture for Database Transaction Monitoring Based on DDM

Table 1. Detection rule R09

RID	UID	Action	Obj1	Obj2	Condition	Time_window	Mon_type	FID	A	B	C
R09	Fabre	update	order details	UnitPrice	ProductID =43	5000	S	S	5	10	32

Table 2. 5 example results of Test 1

AID	RID	FID	A	B	C	X	Result
1118	R18	S	50	200	400	192	0.329208
1124	R02	S	10	50	72	41	0.5
1126	R18	S	50	200	400	194	0.338547
1127	R18	S	50	200	400	195	0.343265
1128	R07	Z	5	40	100	26	0.8120439

Table 3. The summarized data of the two tests

Test		Total	UID						
			Ann	Bob	Charles	Dennis	Eva	Fabre	Gama
1	All results	9971	1463	1431	1383	1357	1422	1549	1366
	Not 1	1380	219	247	130	186	275	152	171
2	All results	1811	152	307	136	64	170	982	2
	Not 1	558	66	207	0	63	153	69	0

That means *R09* is used to monitor the audit records where UID is *Fabre*, *update [order details] set UnitPrice=p where ProductID=43* as SQLText, and *p* is a number. The data items before and after update operation are recorded in the fields Data1 and Data2. When an audit record of that type occurs, *R09* seeks the audit records of that type which have occurred over the past *5000* hours, and sums up the difference between each pair of Data1 and Data2 in each audit record. Then, the sum is substituted into the $F_S(x, 5.0, 10.0, 32.0)$ defined in *R09*. Finally, a result value of the function is assigned as the dubiety degree of that audit record. As this is a real-time process; an audit record has been examined as soon as it has arrived.

*Results***.** The experiment consists of two tests. In Test 1, all of the 19 detection rules are enabled. 9971 of the 30000 audit records are detected as dubious or anomalous. The rest are regarded as normal, as these audit records do not match any of the 19 rules. Among the 9971 results, there are 1380 ones with results being neither *0* nor *1*. The rest ones are either *0* or *1*. Table 2 lists 5 examples. In Table 2, all of the dubiety degrees of these audit records are between *0* and *1*. That means they are "dubious": not completely acceptable or unacceptable. By the "degree", we know *how* dubious a record is.

For the record of AID *1118*, RID is *R18*, while X is *192.0*. When *R18* is matched again in the record of AID *1126*, X is *194.0*. This can be explained because both *R02* and *R18* matched the AID *1124*. For *R18*, its X is *193.0*, while for *R02*, X is *41.0*. Therefore, AID *1124* has *0.5* as result by *R02* and *0.333861* as result by *R18*. Because the result of *R02* is greater than that of *R18*, the audit record is more dubious as measured by *R02* than by *R18*. As a result, *R02* is selected for AID *1124*.

In Test 2, 6 rules including *R02* are disabled. In the results, all the records picked up by *R02* in Test 1 are now picked up by *R01* (in Test 1, R02's dubious degree is higher than R01's). This is because these records are matched by both *R01* and *R02.* When *R02* is disabled in Test 2, *R01* is used where *R02* was selected before. The results of these two tests are summarized in Table 3. For the limitation of space, more details of the experiments are not stated.

5. CONCLUSIONS

This study investigated a new type of intrusion *Cumulated-Anomaly*. A new detection method *Dubiety-Determining Model* (DDM) has been proposed. Based on DDM, a database transaction monitoring system has been designed and implemented. This system has been tested using an SQL server. Tests have been performed to verify the effectiveness of our newly proposed DDM method and show the basic features of it. The results suggest that our methods are capable of identifying suspicious user behaviors.

We are currently working on more details of DDM, such as improving the performance of the algorithms, and constructing the entire detection system.

6. REFERENCES

[1] 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.

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

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

[4] 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.

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

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