Database Security Protection based on a New Mechanism

Amira Rezk
Faculty of Computers and
Information Sciences
Mansoura University
Mansoura, Egypt

H. A. Ali
Faculty of Engineering
Mansoura University
Mansoura, Egypt

S. I. Barakat
Faculty of Computers and
Information Sciences
Mansoura University
Mansoura, Egypt

ABSTRACT

The database security is one of the important issues that should take a complete attention from researchers. Although applying the traditional security mechanisms, the database still violate from both of external and internal users. So, the researchers develop a Database Intrusion Detection System (DBIDS) to detect intrusion as soon as it occurs and override its malicious affects. The previous work developed a DBIDS as a third party product which is isolated from the DBMS security functions especially access controls. The lack of coordination and inter-operation between these two components prevent detecting and responding to ongoing attacks in real time, and, it causes high false alarm rate. On the other hand, one of the directions that are followed to build a profile is the data dependency model. Although this model is sufficient and related to the natural of database, it suffers from high false alarm rate. This means that it needs an enhancement to get its benefits and eliminate its drawbacks.

This Paper aims to strengthen the database security via applying a DBID. To achieve this goal it develops an efficient IDS for DB and integrates it with DBMS for cooperation and completeness between the different parts in the security system. The experiments declare that the proposed model is an efficient DBIDS with a minimum false positive rate (nearly zero %) and maximum true positive rate (nearly 100%). Moreover, it is based on a novel method to build an accurate normal user profile and integrate it with access control.

General Terms

Database, Security.

Keywords

Database security, Intrusion detection. Data dependency. Access Control.

1. INTRODUCTION

Having a secure system is an essential target for any organization, to achieve that, a good security policy must be designed carefully and a multilayered approach to security should be deployed. Also, it is important to denote that the database security cannot achieve without a secured environment surrounds it, i.e. secured network, operating system and application.

The most common threats that face the database are less of integrity, less of availability and less of confidentiality. To protect database against these types of threats, it is common to implement four kinds of control mechanisms; Access control, inference control, flow control, and encryption control [1]. But these database security mechanisms are not design to detect intrusions but to avoid it; this means it is not enough to protect the database [2]. Although applying these mechanisms, database still violate from internal and external attackers. So the quick and accurate detection of attacks on a database system is a prerequisite for fast damage assessment and recovery.

In recent years, researchers interest in database intrusion detection system as a second line of defense in database. There are three main reasons for this. Actions deemed malicious for a DBMS are not necessarily malicious for the underlying operating system or the network. Thus designing an IDS for the latter may not be effective against database attacks [3]. In addition, host- or network-based IDSs are mainly focused on detecting external intrusions as opposed to internal intruders, while legitimate users who abuse their privileges are the primary security threat in database systems. Therefore, SQL injection [4] and other SQL-based attack targeted at databases cannot be effectively detected by network- or host-based IDSs. The distinctive characteristics of DBMSs, together with their widespread use and the invaluable data they hold, make it vital to detect any intrusion attempts made at the database level [5].

Database intrusion refers to an unauthorized access and misuse of database systems. DBIDSs identify suspicious, abnormal or downright malicious accesses to the database system from both of internal and external users. These systems aim to detect intrusions as early as possible, so that any damage caused by the intrusions is minimized. Unfortunately, malicious transactions can seriously corrupt a database through a vulnerability denoted as damage spreading [6].

The purpose of an IDS is classify the input correctly as a normal or an intrusive. So That, the IDS output should faithfully reflect the “truth" about the input (i.e., whether an intrusion occurs or not) [7]. Generally, an IDS works in two phases: the training phase and the detection phase. In the training phase, the data that describes the normal pattern (in the case of anomaly based IDS) or the attack pattern (in the case of misuse based IDS) are collected, pre-processed, and used to create a profile which is stored in a data repository. In the detection phase, the current event is compared with the profile in the data repository to detect if it is a normal or a malicious, if it is a malicious, an alarm is raised and a response is taken according to the response strategy [8, 9].

In any intrusion detection system, building a profile that represents a subject behavior accurately and consistently, is the main goal. While the main challenge is generate an optimal set of rules that maximizes the detection rate and minimizes the false rate.

This paper introduces an efficient IDS for DB (through building an accurate profile which presents users’ activity, maximizing the detection rate, minimizing the false alarm and optimizing the system overload) and integrates it with DBMS for cooperation and interoperation between the different parts in the security system.

The rest of the paper is organized as follows: Section 2 briefly describes the related work. Section 3 descries a problem statement. Section 4 introduces the proposed model. Section 5 discusses the experimental results. Section 6 Conclusions and future work.

2. RELATED WORK

In the last few years, the researchers began to interest in DBIDS as an additional layer of defense. The researchers follow different directions to build their systems. One of these directions is analyzing the query expressions, which has many approaches. These different approaches include syntax-based, semantic based and data dependency.

Syntax-based: analyzing the SQL-expression syntax of queries. In [10], SQL statements are summarized as regular expressions, which are then considered "fingerprints" for legitimate transactions. In [11, 12, 13], database transactions are represented by directed graphs describing execution paths (select, insert, delete etc.) and these are used for malicious data access detection. This approach made assumptions such as restricting the number of distinct queries possible. However, syntactically similar queries may produce vastly different results, leading the syntax-based engine to generate false negatives.

Semantic-based: interested with what the user is trying to access – the result of the query itself – rather than how he expresses it. [14] Introduces a data centric approach in which a user profile is built on what he accesses (i.e. the semantic of the query). The feature vector is extracted from the result set of a query and used to build the user profile.

Data-dependency: This approach mines dependencies among attributes in a database. The transactions that do not follow these dependencies are marked as malicious transactions [15, 16, 17]. In [15], researchers pay attention to the sequences that include write operation and delete the sequences that contain read only operations because a write operation is more critical for the database. In [16] the work in [15] is enhanced by taking the sensitivity of the attributes into consideration in the form of weights. Sensitivity of an attribute signifies how important the attribute is, for tracking against malicious modifications. Researches in [17] also modify the work in [15] by extending the concept of malicious transactions from those that corrupt data to those that either read data or write data or do both without permission.

The data-dependency models in [15, 16, 17] find the association rules between the data items without consideration to who accesses this data. This may generate a rule that conflicts with access control mechanism for some users, and this will cause false positives and violate the availability of the database for these users. On the other hand, the data dependency model that used the association rules, which depend on the principles of support and confidence in their traditional form cause a high false alarm. Also, its performance is affected by the used value of support and confidence. This leads to search for solutions to enhance the system's performance. From another point of view, The previous work that done in DBIDS is built as a third party product which isolated from the DBMS security functions especially access controls. The lack of coordination and inter-operation between these two components prevents detecting and responding to ongoing attacks in real time, and, it causes high false alarm rate.

It is important to highlight that, the DBMS performs authentication and authorization before any transaction execution [18]; this means that, the intruder can submit malicious transactions to the database only by masquerading as a normal user. Access control checks if the transaction attempts to access data that user has authority to access without any intelligence [1]. On the other hand, the IDS check if the behavior of a user while accessing the data is normal or intrusive. However, current database intrusion detection systems work in isolation from access control. The lack of coordination and interoperation between these two components prevents detecting and responding to ongoing attacks in real time, in addition, it causes high false alarm rate. This paper introduces the ideas of merging the DBIDS in DBMS security functions through integrating it with access controls and proposes a DBIDS model based on a modified data dependency model to build an accurate profile for each user. The proposed modification and its integration with access control enhance the performance of the system on the term of true positive rate and false positive rate compared with the rival models

3. PROBLEM STATEMENT

The traditional security mechanisms are fail to prevent the new generation of attacks and the database security is still violated from both of internal and external user. However the researchers develop a DBIDS system to be an additional layer of defense follows the traditional mechanisms, these systems is still suffer from high false alarm rate. On the other hand, current database intrusion detection systems work in isolation from access control. The lack of coordination and interoperation between these two components prevent detecting and responding to ongoing attacks in real time, and, it causes high false alarm rate.

The challenge is building an efficient DBIDS which can detect any malicious transaction with low false positive rate and high detection rate, and integrating it with access control, in order to strength the database security.

4. PROPOSED MODEL

The proposed model integrates the IDS with the DBMS. It uses the access table to determine the user authorization and specify which items he has privilege to read and/or modify. The model mines the dependency among user's data items from his transaction log and generates specific rules, which determine the context in which the user access the data items.

Figure 1 presents the cooperation between the intrusion detection system and access control. When user submits a query to execute, the DBMS checks the user identification and his authorization to execute the query using the access table, if he has authority to access data items in the query, the IDS checks the context in which the user accesses the data items to determine if he follows his normal behavior or not. If there is any deviation from his normal behavior, an alarm is raised, else the query is executed.

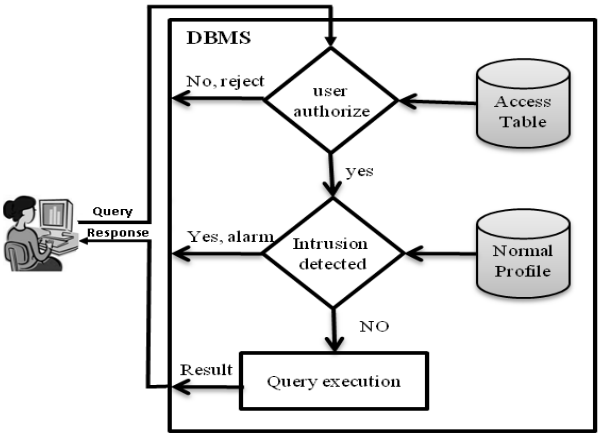

Figure 1: The cooperation between IDS and access control.

Overall, the powerful of the proposed DBIDS can be specified in two points; modifying the data dependency model for generating an accurate and efficient profile, and integrating the DBIDS with access control for minimizing the false rate and strengthening the database security. In order to clarify the benefit of each point separately, the data dependency model as discussed in [15, 17] will be integrated with access control to prove that the integration can refine the performance. Then the modification of the data dependency model will be proposed.

4.1 Integrating Data Dependency Model with Access Control

Data dependency shows access correlations between data items in user's transactions. Transactions that are not compliant to the data dependencies generated are flagged as anomalous transactions. This component of our approach follows the main concept of the data dependencies in [15, 17] with minor modification. For more details and illustrated example review our models details in [19, 20]

4.1.1 Terminology

To discuss the issue of data dependency analysis, it is necessary to introduce the following concepts.

• **Definition 1**: A sequence is an ordered list of read and/or writes operations. A sequence s is denoted by <o(d_1), o(d_2), …, o(d_n)>, where o ϵ {r, w} and d_k is a data item, $1 \leq k \leq n$.

The support for a sequence is defined as the fraction of total transactions that contains this sequence. A subsequence of a given sequence is a sequence formed from the given sequence by deleting some of the elements without disturbing the relative positions of the remaining element.

User's transaction may look like:

T1: r(a),r(b),r(e),w(a),w(d),r(m),w(f) commit.

Data dependency mines the context that user used to follow to access an item.

• **Definition 2:** The read sequence RS(x) of a data item x is the sequence with the format <r(d_1), r(d_2)… r(d_n) , O(x)> which represents that the transaction needs to read all data items d_1, d_2, ..., d_n in that order before the transaction reads or writes x.

• **Definition 3:** The pre-write sequence pre-WS(x) of a data item x is the sequence with the format <w(d_1), w(d_2), …, w(d_n), o(x) >, which represents that the transaction needs to write all data items d_1,d_2, . . ., d_n in that order before the transaction reads or writes x.

• **Definition 4:** The post-write sequence post-WS(x) of a data item x is the sequence with the format < o(x) ,w(d_1), w(d_2), …, w(d_n)>, which represents that the transaction needs to write all data items d_1, d_2, . . ., d_n in that order after the transaction reads or writes x.

pre-writ and post-write sequences are defined instead of a write sequence in [15 - 17]. Write sequence in [15] represents that the transaction may need to write all data items d_1, d_2, …, d_n in this order after the transaction updates data item x. while a write sequence in [17] is extended to both read and write operations as well. For example, sometimes after reading the data item x it is necessary to log this event by writing on the data items p and q to indicate that x is accessed by a specific user at a specific time. In other words, after reading x the values of p and q will be updated by normal transactions.

In the proposed model, the write sequence is modified to be pre-write and post-write sequences. It is straightforward to show why the new definition can improve intrusion detection. For example, sometimes the data item x is updated based on the updated value of data item z, i.e. if the updated value of z satisfy certain condition the data item x will be updated. This means that z should be written before write x, w(z) ϵ pre-WS(x). However, it is not necessary to write x after write z. So, w(x) may not be an element of WS(z).

The process of extracting dependency rules is done in three phases: (1) Mining the frequent sequential patterns, (2) Generating the read, pre-write, and post-write sequence sets, and (3) Extracting the dependency rules among data items.

4.1.2 Mining the Frequent Sequential Patterns

Frequent sequential pattern mining is to find all frequent sequential patterns from a given data set with a user specific support threshold [21].

Instead of finding, the frequent sequential pattern from all transactions log [15, 17], the transactions is divided into subset corresponding to each user and then the frequent sequential patterns in each subset are mined.

4.1.3 Read and Write Sequence Set Generation

After mining the frequent sequential patterns, the read, pre-write, and post-write sequences can be extracted by the following procedure:

- For each operation O(d_i) in sequential patterns, add <r(d_{i1}), r(d_{i2}), r(d_{i3}),…, r(d_{in}), O(d_i)> to the read sequence set of data item d_i where {r(d_{i1}), r(d_{i2}), r(d_{i3}),…, r(d_{in})} is the set of all read operations before O(d_i).
- Similarly, add <w(d_{j1}),w(d_{j2}), w(d_{j3}),…, w(d_{jk}), O(dj)> to pre-write sequence set of data item dj where {w(d_{j1}), w(d_{j2}), w(d_{j3}),…, w(d_{jk})} is the set of all write operations before O(dj).
- Also, add <O(d_j), w(d_{j1}), w(d_{j2}), w(d_{j3}),…, w(d_{jk})> to post-write sequence set of data item d_j where {w(d_{j1}), w(d_{j2}), w(d_{j3}),…, w(d_{jk})} is the set of all write operations after O(d_j).

4.1.4 Data Dependency Rules Generation

Data dependency rules are categorized as read rules, pre-write and post-write rules. The following procedure is utilized to generate data dependency rules.

- For all sequential patterns <r(d_{i1}), r(d_{i2}), …, r(d_{in}), O(d_i)> in read sequence set, generate the read rules with the format O(d_i)→ r(d_{i1}), r(d_{i2}), …, r(d_{in}). If the confidence of the rule is larger than the minimum confidence, then it is added to the answer set of read rules, which depicts that before O(d_i), data items d_{i1}, d_{i2}, …, d_{in} must be read by the same transaction.
- For all sequential patterns w(d_{j1}), w(d_{j2}), …, w(d_{jk}), O(d_i) in the pre-write sequence set, generate the pre-write rules with the format O(d_j)→w(d_{j1}), w(d_{j2}), …, w(d_{jk}). If the confidence of the rule is larger than the minimum confidence, then it's added in the answer set of pre-write rules, which depicts before O(d_j), data items d_{j1}, d_{j2}, …, d_{jk} must be updated by the same transaction.
- Similarly, for all sequential patterns O(d_j),w(d_{j1}), w(d_{j2}), …, w(d_{jk}) in the post-write sequence set, generate the write rules with the format w(d_{j1}), w(d_{j2}), …, w(d_{jk})→O(d_j). If the confidence of the rule is larger than the minimum confidence, then it's added in the answer set of post-write rules, which depicts after O(d_j), data items d_{j1}, d_{j2}, …, d_{jk} must be updated by the same transaction.

The confidence of every rule is calculated similarly to the standard model of mining association rules. So, the confidence of this rule is equal to the following equation.

$$\text{Confidence} = \text{Support}(<r(d_{i1}), r(d_{i2}), \ldots, r(d_{in}), O(d_i)>)/\text{Support}(<O(d_i)>)$$

If the confidence of this rule is higher than the user specified minimum confidence, this rule will be added to the set of read rules.

The result data dependency rule is used t define the normal user activity. Composing the transactions into subsets according to the user who execute it mines new rules for each user. Also it avoids the rules that may conflict with access control, which increases the detection rate and reduces the error rate. However, the data dependency model has some defects, such as:

- Some items do not have dependency rules, because: (1) its appearance in the user log file is rare (less than minimum support). (2) Its correlation with other items is weak (less than minimum confidence); this means that the users can access them in any context. However, these items may be sensitive.
- On the other hand, the item, which has a rule with confidence less than 100%, appears in another context sometimes equal to (100 - minimum confidence). However, the dependency model rejects any transaction conflict with the dependency rules; this means that, some normal transactions will be detected as malicious (false positive error).

These defects lead to search for other solutions rather than the association rules, which depend on the principles of support and confidence in their traditional form

4.2 The Enhanced Data Dependency Model

Transaction could be defined as an executing program that forms a logical unit of database processing. It includes one or more database access operations, these can include insertion, deletion, modification, or retrieval operations [8]. Based on this definition, transaction can be described by the number of its operations, read data set, write data set, and dependency rules. Suppose that, each write with read immediately before it forms an operation. But, the select query does not have a write operation, so if there is no write operation in the transaction, the number of operations will set to 1. However the transaction may contain more than select statement, it will be considered as one operation for simplicity, and the model will concern of the correlation between the accessed items.

Although, the read data set and write data set determine the correlation between data items; two transactions may have the same sets. So, data dependency rules are required to determine the manner of each one to access these data items. However, the data dependency means as in the previous models can be achieved if the read and write data sets save the order of items as it appears in transaction. Additional data dependency rules are required to specify the transaction well done.

After describing the transaction based on its operations' numbers, read data set, write data set and dependency, this description will be used to define the user normal profile. For each user, each data item he has authority to access it will have an access profile. To minimize the number of rules that describes the user behavior, an optimized algorithm will be applied to remove the redundancy and keep the number of rule as minimum as can.

Now, it is important to test if the enhanced dependency model can override the drawback of the dependency model or cannot. This will be declared in the next section.

5. EXPERMINTAL RESULT

In order to demonstrate the efficiency of the proposed model, we develop a set of simulations with different parameters as input to the algorithm. The simulation program was written in Java programming language.

Two different database logs were generated. The first log comprises normal transactions and the second log comprises intrusion transactions. To create the normal transactions we generate read and write operations based on a dependency factor. The dependency factor defines the average number of read operations immediately before a particular operation O(x) or the average number of write operations immediately after that operation. To be consistent with the experiments presented in [15, 17], the maximum number of sequential read or write operations in the generated transactions is five. To generate intrusion transaction logs we assume that hackers have no information about the dependency rules among the data items and hence the read and write operations of intrusion transactions are randomly generated.

The evaluation will be done in two phases. First, the data dependency model as discussed in [15, 17] will be integrated with access control to prove that the integration can refine the performance. The results will compare with [15, 17] approaches. Then the modification of the data dependency model will be test compared with the results of the first phase. Several experimental will be done in each phase to get an accurate evaluation of the proposed model performance.

5.1 Integrating Data Dependency Model with Access Control

Several experiments are conducted for evaluating the performance of the proposed model in terms of false positive and true positive rates. False positive rate (FP), which is the probability that the IDS outputs an alarm when there is no intrusion, and true positive rate (TP), which is the probability that the IDS outputs an alarm when there is an intrusion [7]

To evaluate the false positive rate of the proposed method, we use the normal transactions' log as both training and test transactions. So, the more an approach identifies transactions as malicious, the higher the false positive rate. On the other hand, the normal and intrusion transaction logs for training and testing respectively are used to evaluate the true positive rate.

However, the proposed model in this phase introduces simple modifications on the data dependency model. These modifications are:

- Modifying the write set to be pre-write and post-write set.
- Decomposing the transaction log file to sub-files according to user (sub-transaction log file for each user)
- Integrate a model with access control

In the following sub-sections, each one of these modifications will be evaluated separately to get a complete analysis.

5.1.1 Modifying the Write Set

The aim of this stage is to evaluate the algorithms' false positive rate and true positive rate based merely on the dependency rule, the proposed model is applied on the transaction log file without decomposing it according to users to reflect the output of the first modification and eliminate the effect of the other modifications.

In order to evaluate modifying the write set to be pre-write set and post-write set, a set of experiments is done. The true positive rate and false positive rate is examined for proposed model and model in [15, 17] according to minimum support values set {10, 15, 20, 25} and minimum confidence values set {70, 75, 80, 85}. Figure 2 to Figure 5 show the results of FP rate while Figure 6 to Figure 9 show the results of TP rate.

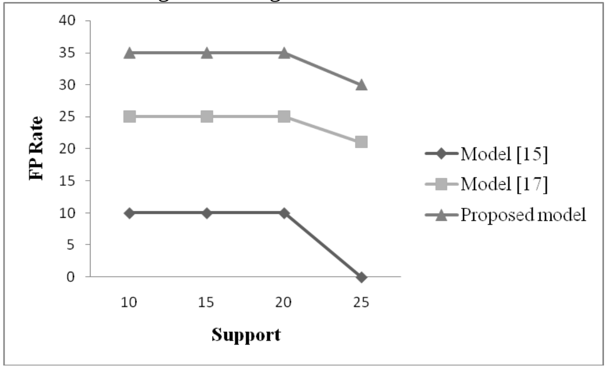

Figure 2: FP rate of minimum Confidence 70

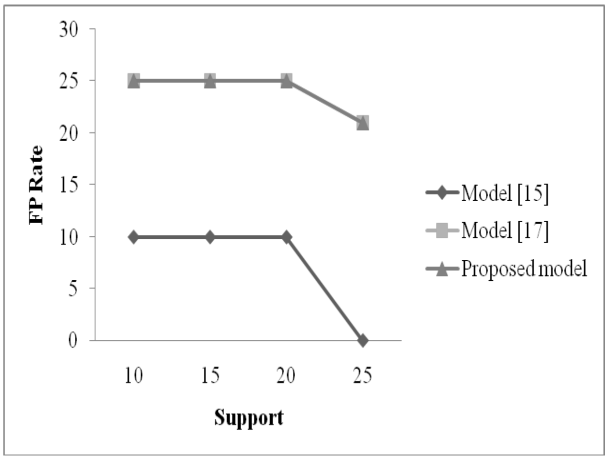

Figure 3: FP rate of minimum Confidence 75

Analyzing Figure 2 to Figure 5 shows that, the false positive rate of the rival methods depends on the number of extracted rules and their confidences. Whenever the total number of dependency rules increases, the probability becomes higher that one of them is violated by a normal transaction, in which case the normal transaction is identified as an intrusion. The false positive rate increases when the number of rules increases (with minimum confidence < 100). As for the rate reduction in the FP, it is because the correlation between data items increases (confidence increase). As a result, the rules attain a higher confidence improve the false positive rate.

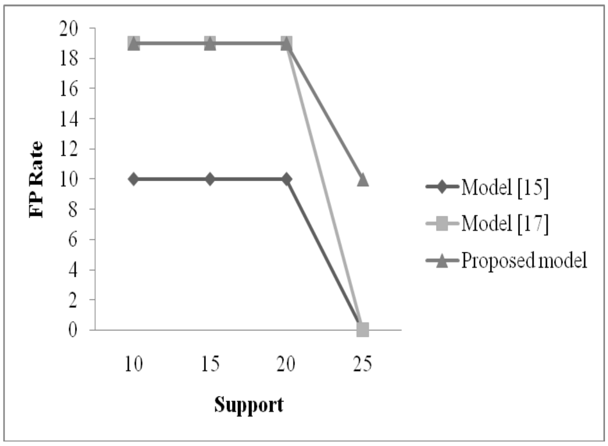

Figure 4: FP rate of minimum Confidence 80

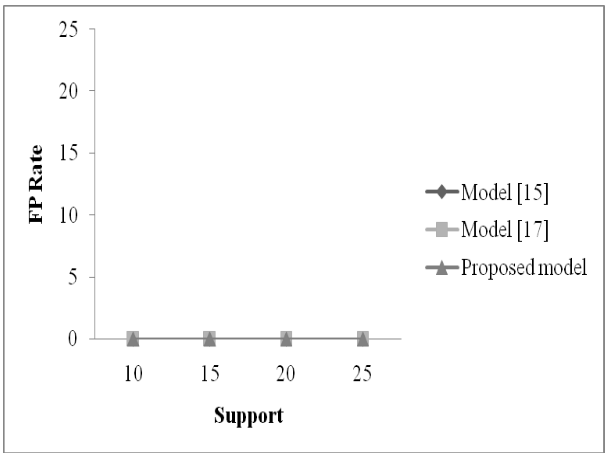

Figure 5: FP rate of minimum Confidence 85

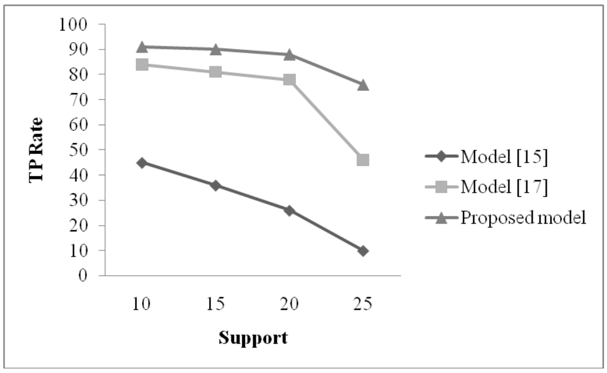

Figure 6: TP rate of minimum Confidence 70

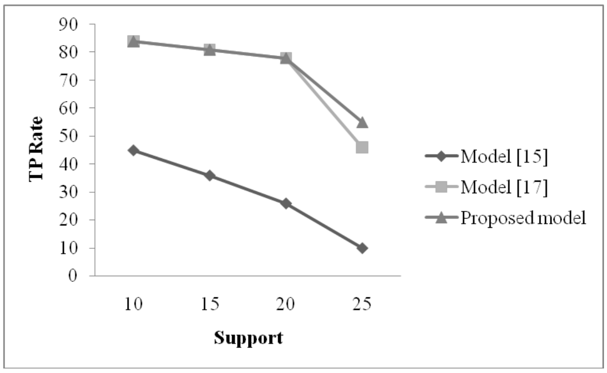

Figure 7: TP rate of minimum Confidence 75

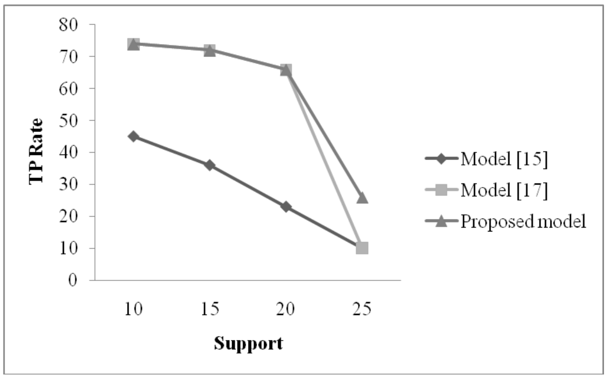

Figure 8: TP rate of minimum Confidence 80

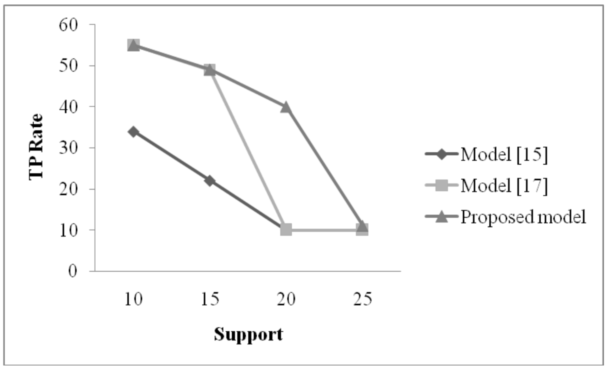

Figure 9: TP rate of minimum Confidence 85

Figure 6 to Figure 9 compare the true positive rate of proposed model against the rival method. It can be observed that with an increase in the dependency rules, algorithms' true positive rates increase. But the proposed approach mostly performs better than the alternative method. This improvement is achieved by using the pre-write and postwrites sequence sets, which take into account the variety of dependencies among operations.

Generally, when the number of dependency rules increases the FP rate and TP rate increase. The dependency rules increases when a minimum support value and minimum confidence value decrease. So, it is important to adjust the value of minimum support value and minimum confidence to balance between the TP and FP rates. In the second phase of the proposed model, the shortage of the model because of the support and confidence values will be overridden with the proposed modification on the data dependency model.

5.1.2 Mining Dependency Rules for Each User

In this stage, a set of experiments is done to evaluate decomposing the transaction log file into sub-log files for each user, and mining the dependency rules for each one separately. So, the Transaction log file is divided into {2, 3, 4} sub-log files to find the affect of the number of users on the model. Figure 10 and Figure 11 show the FP rate and TP rate for each case respectively. The dependency rules are found using minimum support 20% and minimum confidence 75%.

It is observed that the proposed model has a FP rate less than model [17] and more than model in [15]. However it achieves higher TP rate than model [15] and near to model [17]. Decomposing the transactions into subsets according to the user who execute them mines new rules for each user, and avoid the rules that does not reflect his normal activity which cause FP rate when it is violated.

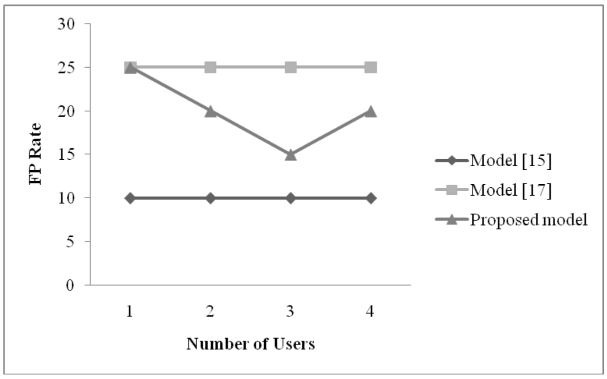

Figure 10: FP rate according to numbers of user (min. sup. 20% and min. conf. 75%)

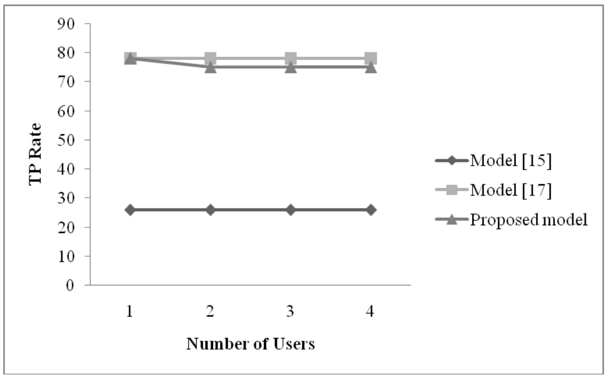

Figure 11: TP rate according to number of users (min. sup. 20% and min. conf. 75%)

5.1.3 Integrating the Proposed Model with Access Control

Finally, the integration of proposed model with access control is examined. In these experiments the authority of each user to access the data item is taken in consideration to simulate the behavior of the access control mechanism. The rival models work as its normality without any change. While the proposed model reforms the malicious transaction and eliminates the transaction that violates the authority of each user. The reminder of the malicious transaction that does not violate the authority is checked using the dependency rules. Figure 12 and Figure 13 dedicate the FP rate and the TP rate respectively when applying the model with minimum support 20% and minimum confidence 75%.

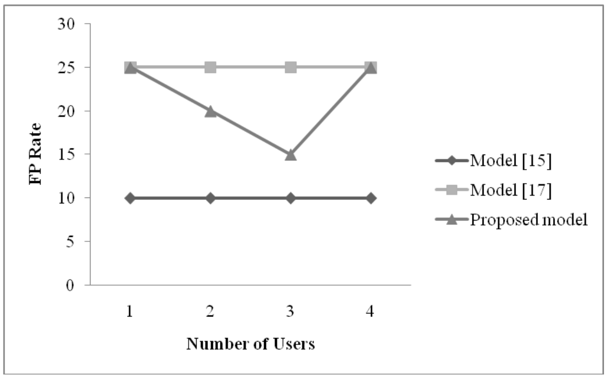

Figure 12: The effect of the integration with Access Control on FP rate (min sup. 20% and min. conf. 75%)

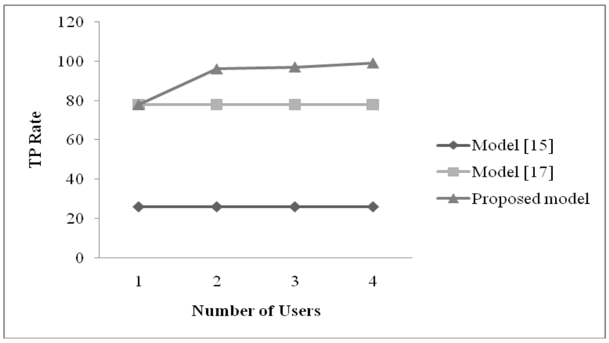

Figure 13: The effect of the integration with Access Control on TP rate (min sup. 20% and min. conf. 75%)

Comparing Figure 12 with Figure 10 and Figure 13 with Figure 11 declares that the FP rate does not change because the dependency rules that describes the normal behavior of the user still as it is and nothing effect on it. While the TP rate of the proposed model increases to be higher than the other models. Because the rival models work in isolation from access control, they checks each submitted transaction. However some of the submitted transactions violate the authorization of the user. On the other hand, the proposed model checks only the transaction that try to access data items which user has authority to access it and any submitted transaction tries to get an illegal access to data item is omitted. This matter reduces the effort of detection process because it reduces the number of examined transaction

Integrating the IDS with the DBMS makes the database more resistant to attacks and detects the malicious transaction before its execution and this avoid the system the cost of the malicious transaction execution and recovery.

5.2 Evaluating the Modified Data Dependency Model.

Although integrating DBIDS with access control enhances the detection rate, the model still suffers from false alarm rate. It is important to note that DBIDS follows other traditional database security mechanisms and network security mechanisms such as firewall and network intrusion detection.

Therefore, it faces the intrusion from internal users or the intrusion that can be passed through other security layers. This means that the number of intrusion event is rare compared to the number of the normal event. So, it is not efficient to raise a large number of false alarms to achieve a high detection rate.

In the Second phase of the proposed model, an enhancement for the data dependency model is introduced to override the high rate of false alarms. In this section a set of experiments is done to evaluate this modification based on the FP rate and TP rate. Figure 14 and Figure 15 present the FP rate and TP rate respectively for the two phase of the proposed model. The first phase represents the last result that is got from integrating the model with access control. While the second phase represents the results of the modified data dependency model.

Figure 14 shows that the FP rate reduces in the second phase to reach a zero%. This is because that the modification, which is done on the dependency model, omits the dependency rules that have a confidence less than 100%, and replaces it with representation of the transaction according to its number of operation, read set, write set and dependency. Also Figure 15 declares that the TP rate in the second phase is better than the first one. As well, it is observed that, the performance of the model is better than the rival model especially when the number of the users increases

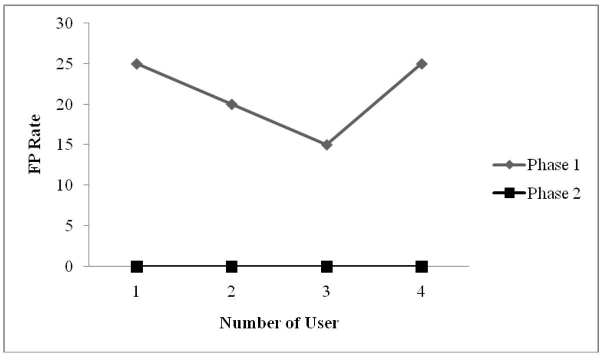

Figure 14: The effect of 2nd Phase's modification on FP rate

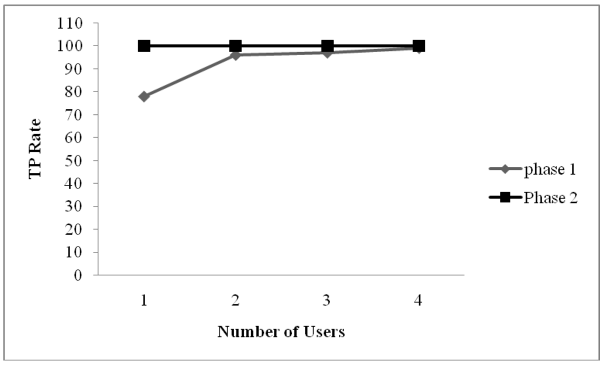

Figure 15: The effect of 2nd Phase's modification on TP rate

Based on all of these experiments, it is obvious that the proposed model is an efficient DBIDS with a minimum FP rate (nearly zero %) and maximum TP rate (nearly 100%). The proposed model introduces a method to build an accurate normal user profile and integrates it with access control. This correlation and cooperation between DBIDS and access control enhance the performance of the system and strengthen the database security.

It is important to report that, the proposed model outputs error only in two cases, first, if the user tries to perform a malicious event using his normal transaction in different context, in this case no alarm rises. This situation can be eliminated using additional factor as update latency, execution frequency, time of execution, and etc. The second case occurs when the user modifies his normal transaction, or defines new ones, and this can be solved by update the user's normal behavior.

6. CONCLUSION AND FUTURE WORK

The database security is one of the important issues that should take a complete attention from researchers. To get a secured database system, there is no magic technique used to achieve this goal. However there is a set of controls that can be applied in order to develop a secured database system and it should be preceded with a good security plan. Although applying the traditional security mechanisms, the database still violate from both of external and internal users. So, the researchers develop DBIDS to detect intrusion as soon as it occurs to recover its malicious affects.

The previous work that done in DBIDS is built as a third party product which isolated from the DBMS security functions especially access controls. The lack of coordination and interoperation between these two components prevents detecting and responding to ongoing attacks in real time, and, it causes high false alarm rate. On the other hand, one of the directions that are followed to build the profile is the data dependency model. Although this model is sufficient and related to the natural of database, it suffers from high false alarm rate. This means that it needs an enhancement to get its benefits and eliminate its drawbacks.

We present a proposed model that tries to build an accurate profile based on a modified data dependency model and integrate it with access control.

Several experiments are conducted for evaluating the performance of the proposed model in terms of false positive and true positive rates. The experiments are divided into sets to evaluate each parameter of the proposed model compared with the rival models. The experiments declare that the proposed model is an efficient DBIDS with a minimum FP rate (nearly zero %) and maximum TP rate (nearly 100%).

The future work will enhance the model by adding additional features to define the normal behavior of each user, such as the user's identity, the user's role in the database system, the type of connection to the database, the ID of the application which submits transactions to the database server and so on. Incorporating these features into the detection scheme might give us further clues to effectively discover malicious transactions.

7. REFERENCES

[1] Elmasri, R., and Navathe, S. B. 2007. fundamentals of database system, 5th edition, Addison wesley

[2] Vieira, M., and Madeira, H. 2005. Detection of Malicious Transactions in DBMS. In Proceeding of the 11th pacific rim international symposium on dependable computing, 350-357

[3] Kamra, A., Terzi, E., and Bertino, E., "Detecting anomalous access patterns in relational databases". VLDB journal, 17, 5, 2008, 1063-1077

[4] Clarke, J. 2009. SQL Injection Attacks and Defense. Syngress, Burligton, MA.

[5] Jin, X., and Osborn, S. L., 2007. Architecture for data collection in database intrusion detection systems. In

Secure Data Management , Springer-Verlag, Berlin, 96–107

[6] Liu, P. 2002. Architectures for intrusion tolerant database systems. In Proceedings of the Annual Computer Security Applications Conference (ACSAC'02), 311-320

[7] Gu, G., Fogla, P., Dagon, D., Lee, W., and Skoric, B. 2006. Measuring Intrusion Detection Capability: An Information Theoretic Approach. In Proceedings of the ACM Symposium on Information, computer and communications security, 90-101.

[8] Solomon, G., and Chapple, M., 2005. Information Security Illuminated, Jones & Bartlett Learning, USA

[9] Rezk, A., Ali, H. A., Elmikkawy, M., and Barakat, S. "Database intrusion detection system – A short survey", Accepted for publishing in Mansoura Journal of Computer and Information Sciences (MJCIS).

[10] Lee, S.Y., Low, W.L., and Wong, P.Y., 2002. Learning fingerprints for a database intrusion detection system. In ESORICS, LNCS, vol. 2502, Gollmann, D., Karjoth, G., and Waidner, M. Springer, Heidelberg, 264–280

[11] Fonseca, J., Vieira, M., and Madeira, H. 2008. Online detection of malicious data access using DBMS auditing. In Proceedings of the ACM symposium on Applied computing (SAC'08), Brazil, 1013-1020.

[12] Fonseca, J., Vieira, M., and Madeira, H. 2007. Integrated intrusion detection in databases. In Proceeding of Third Latin-American Symposium on Dependable Computing (LADC 2007), Morelia, Mexico, September, 198- 211

[13] Chagarlamudi, M., Panda, B., Hu, Y. 2009. Insider threat in database systems: Preventing malicious users' activities in databases. In proceeding of Sixth International Conference on Information Technology: New Generations, 1616-1620

[14] Mathew, S., Petropoulos, M., Ngo, H. Q. and Upadhyaya, S. 2010. A data-centric approach to insider attack detection in database systems. In Recent Advances in Intrusion Detection (RAID) Symposium, Springer, 382- 401.

[15] Hu, Y., and Panda, B., 2004. A Data Mining Approach for Database Intrusion Detection. In ACM Symposium on Applied Computing, 711 – 716.

[16] Srivastava, A., Sural, S., and Majumdar, A.K. "Database intrusion detection using weighted sequence mining," Journal of Computers, VOL. 1, NO. 4, JULY 2006. 8-17

[17] Hashemi, S., Yang, Y., Zabihzadeh, D., and Kangavari, M. "Detecting intrusion transactions in databases using data item dependencies and anomaly Analysis", Journal of Expert Systems, Vol. 25, No. 5, Blackwell Publishing Ltd, November 2008, 460-473

[18] Lewis, M. 2004. SQL Server Security Distilled, 2nd ed., Apress, New York, NY

[19] Rezk, A., Ali, H. A., Elmikkawy, M., and Barakat, S., Integrating a Database Intrusion Detection System with Access Control. In proceeding of 5th international conference on intelligent computing and information system, Egypt, 2011, 46- 52

[20] Rezk, A., Ali, H. A., Elmikkawy, M., and Barakat, S., "Minimize the False Positive Rate in a Database Intrusion Detection System", International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 5, Oct 2011, (DOI : 10.5121/ijcsit.2011.3503), 29-38

[21] Wang, W., and Yang, J., 2005. Mining Sequential Patterns from Large Data Sets, Springer.