Proc. of Int. Conf. on Computer systems, Data Communication and Security, CDCS

Detecting Intrusive Malicious Transactions in Database using Session and Token Management

Indu Singh[1], Tapasya Singh[2] and Tanya Verma[3]
[1]Delhi Technological University, Department of Computer Engineering, New Delhi, India
indu.singh.dtu14@gmail.com
[2]Delhi Technological University, Applied Mathematics, New Delhi, India
tapasya.singh.dtu12@gmail.com
[3]Delhi Technological University, Applied Mathematics, New Delhi, India
vtanya16@gmail.com

***Abstract*—With the advent growth of organizational expectations, the new ability of security systems is revolutionised to secure confidentiality and integrity of the system interface. Unauthorised access of database can result in significant losses for organizations and individuals. With the increased number of attacks, intrusion detection has become vital part of information security. The database intrusion detection model and algorithm proposed in this paper analyses user access patterns using three phases. Learning phase provides security access levels for users to generate valid profile in profile generating phase via session and token manager. In the last phase, valid transactions are committed and an alert is generated for malicious transactions. We have implemented our proposed architecture along with transaction level approach for the detection of malicious transaction incorporating CBF.**

***Index Terms*— Database Security, Anomaly Intrusion Detection, Counting Bloom Filter (CBF), Session and Token management.**

I. INTRODUCTION

Internet today plays a major role in forming the structure of new business avenues and formulating certain advances in them. However, with the development of human brain and certain advances in technology, nothing in the internet is safe these days. Thus, modern needs have motivated business firms and governments across the globe to develop sophisticated and complex information networks. Protection of user information is of utmost importance as the user data travelling along the internet is susceptible to several internal as well as external threats.

Today all organizations rely on database systems as the key data management technology to deploy several tasks ranging from day-to-day operations for critical decision making. Such widespread use of database systems implies that security breaches to these systems affect not only a single user or application, but also may have disastrous consequences on the entire organization. Most operating systems do ensure a certain level of security by offering some protection at the file system level, however only typical files and directories are covered under protection and these protection units are incompatible with database management systems. Database security is the system, processes and procedures that protect a database from unintended activity.

Grenze ID: 02.CDCS.2015.1.506

As discussed by Bertino and Sandhu [13], data security breaches are typically classified as unauthorized data observation, improper data modification, and data unavailability. Unauthorized data observation leads to the improper access to data with certain individuals getting access to data that they are not permitted to. The unauthorized disclosure of personally identifiable data results in privacy breaches that may lead to identity theft and other serious consequences for the individuals.

Improper data modifications tend to result in incorrect data. Improper data modifications can be intentional or unintentional. Incorrect data may also result in heavy losses for organizations. The unavailability of data leads to the unavailability of crucial information required for the proper functioning of an organization. Thus, a complete solution to data protection must meet three key requirements: authenticity (guarantees that a service of information is authentic, **confidentiality**— refers to the protection of data against unauthorized disclosures; **integrity** refers to the prevention of improper data modifications; and **availability** refers to the prevention and recovery from hardware and software errors and from malicious data access denials making the database system unavailable. These three requirements arise practically in all applications [6].

Access control mechanisms are key component for assuring data protection and integrity. When a subject attempts to access some data, the access control mechanism checks whether or not the subject has the authorization, semantic integrity and access control policies to perform the action on the data. Encryption techniques are used to enhance the confidentiality further [2].

Insider Attacks are one of the most dangerous threats organizations face today. An insider attack occurs when a person authorized to perform certain actions in an organization decides to abuse the trust, and harm the organization. These attacks may negatively impact the reputation of the organization, its productivity, and may produce losses in revenue and clients. Avoiding insider attacks is a daunting task. While it is necessary to provide privileges to employees so they can perform their jobs efficiently, providing too many privileges may backfire when users accidentally or intentionally abuse their privileges. Hence, finding a middle ground, where the necessary privileges are provided and malicious usage are avoided, is necessary.

An intrusion detection system gathers and analyzes information from various areas within a computer or a network to identify possible security breaches. In other words, intrusion detection is the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a system/network. Traditionally, intrusion detection systems have been classified as signature detection systems, anomaly detection systems or a hybrid/ compound detection systems. A signature detection system identifies patterns of traffic or application data presumed to be malicious while anomaly detection systems compare positive rate. The major drawback of the signature detection approach is that such systems typically require a signature to be defined for all of the possible attacks that an attacker may launch against a network. The biggest advantage of anomaly detection systems is that profiles of normal activity are customized for every system, application and/or network, and therefore making it very difficult for an attacker to know with certainty what activities it can carry out without getting detected. However, the anomaly detection approach also has its share of drawbacks: the intrinsic complexity of the system and the difficulty of associating alarms with the specific events that triggered those alarms. A hybrid intrusion detection system combines the techniques of the two approaches [5].

In this paper we have proposed an innovative approach for the detection of malicious transactions in database using intrusion detection system .Our proposed technique is enhanced to incorporate CBF, session and token management to define user behavior of operation as legitimate or illegitimate.

Rest of this paper is organized as follows. In section II, we discuss related background. In section III, architecture of our proposed database intrusion detection system is defined. In section IV, our proposed algorithm for detecting malicious transaction is given. In section V, the performance evaluation and analysis of the proposed system is given.

II. Related Work

The widespread proliferation of computer networks usage has resulted in the increase of attack on information systems. These attacks are used for illegally gaining access to unauthorised information or misuse of data. This results in huge financial losses to companies besides losing their goodwill to customers as their customer services are disrupted. These attacks are increasing at a staggering rate. Thus there is a need for complete protection of organizational computing resources which is driving attention of people towards intrusion detection and prevention systems.

Flavio Bonomi et al. [1] proposed a simple hashing-based technique based on *d*-left hashing called a *d*-left CBF (dlCBF). This technique offers the same functionality as a CBF, but uses less space. They have described the construction of dlCBFs, providing analysis, and demonstrated their effectiveness experimentally.

For the prevention and detection of SQL injection at the database layer of an application, Cristian Pinzón et al. [2] incorporated a CBR(Case-based reasoning) mechanism whose main characteristics involved a mixture of neural networks, multi agent systems, learning capabilities of CBR systems and distributed artificial intelligence technique that carry out the task of filtering attacks.

In [9] a Misuse Detection System for Database System (DEMIDS) has been proposed by Chung.*et al* especially for relational database systems. The authors derived profiles describing behaviour of various users in DBMS from audit data log for detection of misuse behaviour against observed events. Also description of similar working area of the user by searching frequent item sets was proposed.

Lee et al. [10] used time signatures in discovering the intrusions in real-time database systems. If a transaction tried to attempt to update a temporal data item which is already updated within certain period, the systems detects it as an anomaly. Real-time database systems have a deal with data that changes its value with time. These temporal data objects are used to reflect the status of object in the real world.Whenever the value of a real world object changes, the data describing this object should change as well. Their intrusion detection model observed the database behaviour through sensor. An alarm is raised if a transaction attempts to update a temporal data which has already been updated in that period.

B.Panda et.al [7] presented a model for addressing the problem of insider threat at database level by mapping several transactions on the basis of different user tasks. An implementation oriented approach was developed for validating these tasks after execution by authenticated users.

Jose Fonseca et.al [6] proposed a mechanism that allows concurrent detection of malicious data access through the online analysis of Database Management Systems (DBMS) audit trail. Profiles of valid transactions were represented by directed graphs to detect illegal accesses to data consisting of unauthorized sequences of commands.

For detecting the intrusions Yi et.al [11] uses mining of data dependencies and correlations from the database log. Classification rules were generated to identify the intrusive activities. The transactions which were not following these rules were flagged as anomalous transactions.

M.Doroudian et al. [12] experimentally evaluated the approach describing intrusion detection at database transaction and inter-transaction level for analysing malicious behaviour by different users. In this method anomaly based detection collaborated with data mining was given for discovering temporal rules of relations among identified authenticated transactions at user task level.

A.Rezk et al. [8] introduced traditional database security and network security mechanisms such as firewall and network intrusion detection. An enhancement for the data dependency model has been proposed and integrated with access control techniques to override the high rate of false alarm and increased detection rate.

U.P.Rao et al.[3] demonstrated the novel database intrusion detection approach which reduces the false positive and false negative rates after matching user profiles. This includes two methods ie allowance of subset of attribute access pattern of user profile and altered sequence of consecutive select commands.

A strong incentive was provided by Ricardo Jorge Santos et al. [4] to deal with the main DIDS(database intrusion detection techniques) currently available and discussed the issues concerning their application at the database server layer. The weak points were identified showing that most DIDS inadequately deal with many characteristics of specific database systems, such as *ad hoc* workloads and alert management issues in data warehousing environments. The main finding is that despite the importance of the role played by benchmarks in testing and comparing systems, until this moment no benchmark has been proposed for evaluating the performance of DIDS at the database level.

III. Proposed Approach

Nowadays database security has become an important aspect in information systems engineering. Malicious Intrusive penetrations into the computer systems are possible by legitimate users through unauthorized anomalous access patterns, perquisite misuse, users expeditional nature and sequence of queries issued by internal and external intruders. Therefore to grasp the dynamic nature of user, we have proposed a model for database using intrusion detection system.

A. Architecture

The management of network security and prevention of intrusive activity is a task which causes a lot of concern. However, this task is significantly simplified if the proposed system has a strong architectural

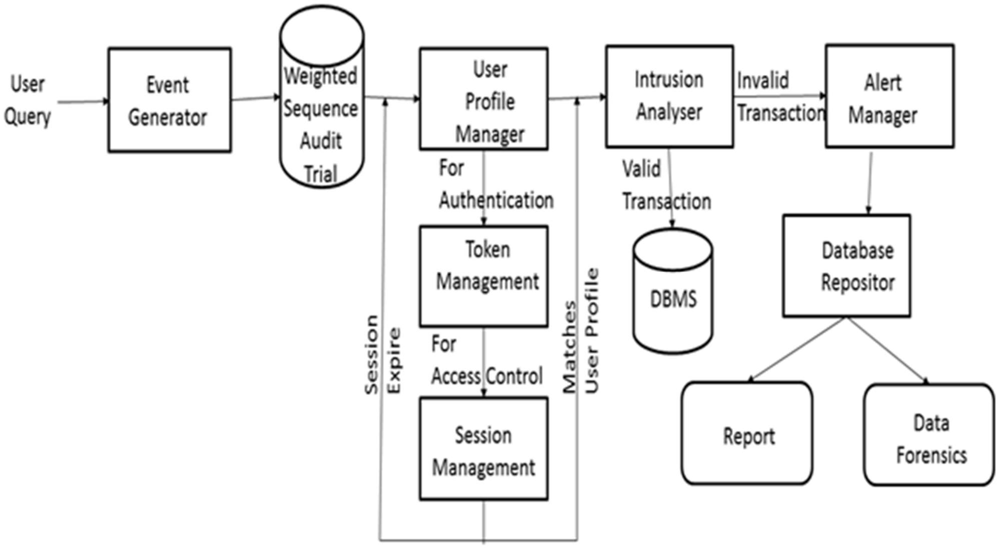

Fig1.Architecture of Database Intrusion Detection Model

foundation. Hence, the architecture of our proposed model ie in Fig.1 is formed by laying emphasis on certain crucial processes such as Event Generator, User Profile Manager, weighted sequence Audit trail, a strong Intrusion Analyser , and for providing authentication and access control through session and token management and final module is of Alert Manger which captures invalid transaction.

The management of network security and prevention of intrusive activity is a task which causes a lot of concern. However, this task is significantly simplified if the proposed system has a strong architectural foundation. Hence, the architecture of our proposed model ie in Fig.1 is formed by laying emphasis on certain crucial processes such as Event Generator, User Profile Manager, weighted sequence Audit trail, a strong Intrusion Analyser , and for providing authentication and access control through session and token management and final module is of Alert Manger which captures invalid transaction.

We can describe our architecture model in three phases:

Phase I : *Learning Phase*

In this phase, event generator is used to monitor the operation of database in real time through analysing security access levels of users and it restricts malicious users from inferring prohibited data through amalgamation of queries.

Phase II : *Profile Generator*

On the basis of the event triggering, event generator checks for previously visited tables and session thresholds and then features will be fed into the weighted sequence audit trail. Weights are assigned by the database administrator during profiling phase through random number generator and authorised transactions are represented as sequence of valid commands. These commands would be sent to user profile manager to check validity of transaction using CBF. Token manager provides dynamic Token ID to valid generated profiles on the basis of user privileges and roles for authentication. Now, session manager will issue session ID corresponding to the generated dynamic Token ID so that it can’t be used again in the future for security purpose. Upon the expiring of the session, the user privileges can be revoked by user profile manager.

Phase III :*Alert Manager*

Intrusion Analyser use anomaly detection where user activities are compared with the current user profiles to determine any abnormal deviation beyond the threshold limits to validate the user query. If transaction is declared valid then it is committed into database otherwise the alert manager would raise an alert and store alerts in the database repository for receiving at a later time. Reports are created for summarizing alert activities. Data forensics are used for further investigating long term trends .

B. Design and implementation

Here we are using sample dataset of transactions and database with table warehouse , district , customer , neworder , orders and stock. Our audit log contains information like user ID, transaction ID (TID), session ID

(SID), sequence number (Seq No), operation and entity for transaction performed with database. Table I is used as input for detecting malicious transactions based on the records of audit log files using intrusion analyser.

TABLE I : AUDIT LOG

User ID	T ID	S ID	Seq No.	Operation	Entity
U1	1	1	1	Select	Warehouse
U1	1	1	2	Insert	District
U1	1	1	3	Insert	Customer
U1	1	1	4	Select	Neworder
U1	1	1	5	Insert	Orders
U1	1	1	6	Update	Stock

TABLE II : DETECTION OF MALICIOUS TRANSACTION

User ID	T ID	S ID	Seq No.	Operation	Entity
U1	1	1	1	Select	Warehouse
U1	1	1	2	Insert	District
U1	1	1	3	Insert	Customer
U1	1	1	4	Select	Neworder
U1	1	1	5	Insert	Orders
U1	1	1	6	Update	Stock

Table II shows the result to examine unauthorised transaction profile on the basis of weights which are randomly assigned by the Database Administrator (DBA) and hash values calculated by hashing functions. We find filter counter values step by step using CBF. Finally, as resultant counter bits are not equal to zero, transactions generated are said to be Malicious.

IV. OUR PROPOSED ALGORITHM

The formal algorithm of database intrusion detection model is presented as follows:-
TID : Token ID
SID : Session ID
CBF : Counting Bloom Filter
UC: User counter
Input : user query , user_entry[ ] , user_entry_weight , CBF

1. Initialize user counter UC=0;
2. Automatic generated random value by a function i.e. (public key) V(x)=F(R(x))
3. Enter equivalent value to generated value i.e. E(x)
4. {
5. Calculate V'(x)=F'(x)
6. If (V(x)==V'(x))
7. Enter in user input log
8. initialise entry = user_entry[ ]
9. set ew = user_entry_weight[ ]
10. initialise TID = new Token_ID

```
11. do while user still enters input
12.       set entry = user input ;
13.       if ( user_role = = expected _role )
14.                     {
15.                 set SID = new Session_ID ;
16.                 assign ew = weight;
17.                 set CBF = CBF – 1;
18.                      }
19.       Else
20.                       {
21.                 invalid TID;
22.                 revoke ( );
23.                 rollback ( );
24.                        }
25. if any CBF ! = 0 then rollback
26. else commit ( ) and generate new profile from SID
27. }
28. Else
29. {
30. UC++
31. If(UC>3)
32. {
33.  exit()
34. }
35. Else
36. {
37. Go to step no.2
38. }
```

In this above algorithm the user is allowed to enter commands but at particular time interval user can execute only one transaction. The same user performing transactions from one client machine cannot execute two transactions in parallel. For identifying authenticated user we are using password generated by Random Key Generator. Always different Random number key is generated for the user by the system and then key is inserted into polynomial function for verification. We have used a polynomial function which will be acting as public key of the user ie. $F(x)=x^2+2x+1$, this f(x) will be known to authorised user only and then the random key generated by the system will be inserted into password function V(x) ie $V(x)=f(x)+3$. Then polynomial output is displayed on the screen to user and now user will enter its equivalent polynomial value ie V(x) that will be known to authorised user only. If input value is matched equal to polynomial value then only user will be allowed to proceed further. And also if user is not able to enter correct polynomial value in 3 attempts then user will be blocked and directed towards administrator. If user has correctly entered into the system after passing through above password method then only we allow user to select a particular transaction consisting of sequence of commands. IDS loads corresponding CBF (Counting Bloom Filter) and a weight_command list. The weight_command list contains weights and corresponding commands allowed in the system. If user's command is valid then counter values in CBF are decremented using weight of the identified command. The user is allowed to enter all the commands one by one. When user stops entering commands the counting bloom filter is checked. If all the counter values in CBF are zero then the transaction is declared as valid and a dynamic Token ID is issued on the basis of user profiles and privileges granted. And then session manager will issue Session ID after matching access security levels of users through role based access control(RBAC) mechanism. Then valid transaction would be committed into database otherwise the transaction will be treated as a malicious transaction.

V. Performance Evaluation and Analysis

Our approach presents the implementation of the intrusion prevention system mechanism. Our experimental result usesthe standard benchmark TPC-C to generate profiles of valid transactions and simulate malicious transactions using randomly generated transactions.

A. Counting Bloom Filter (CBF)

CBF uses 'm' random hash functions, each of which hashes some set elements to one of the n bits array positions. An element can be inserted into a set by passing it into m hashing functions and 'm' index values are obtained. All counters in CBF are incremented. Reverse process is followed to delete an element from the set and corresponding counters are decremented. CBF generalizes a bloom filter data structure by allowing the membership queries and resolves the problem of a standard bloom filter with false positives.[1]
Initial phase of CBF ensures the correctness of the genuine profiles as declared. At the detection phase, database IDS system considers the executable transactions and validate them to ensure that the particular transaction is valid or not.

B. Variation of false negatives percentage with increase in number of hashing function

False negative percentage depends on the number of hashing functions used. We measure the impact on false negatives percentage by using different number of hashing functions keeping constant size of CBF. Results in fig. 2 shows a sudden rise in percentage of false negative when size of CBF used is odd whereas in case when size of CBF is even there is a gradual increase in percentage of false negatives.

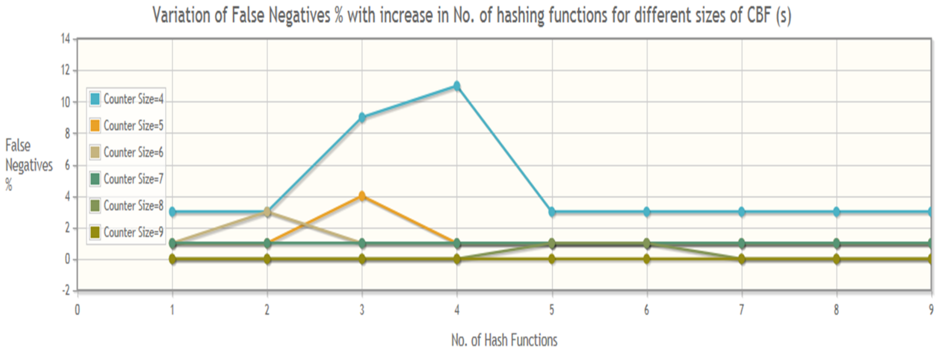

Fig.2 variation of false negatives percentage with increase in number of hashing function

C. Variation of false negatives percentage with increase in size of CBF

As the false negative percentage may depend on size of CBF used we decided to vary size of CBF keeping number of hashing functions constant to observe the change on false negative percentage .Fig. 3 shows the variation in false negative percentage when odd number of hashing function used whereas Fig. 4 shows the variation where even number of hashing functions used. It is seen that false negative percentage decreased with increase in size of CBF used if number of hashing functions is kept constant.

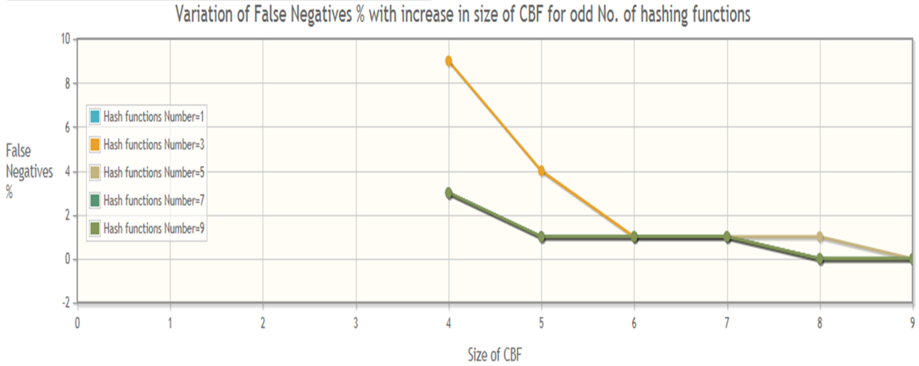

Fig. 3 variation of false negatives percentage with increase in size of CBF for different odd number of hashing functions (hf)

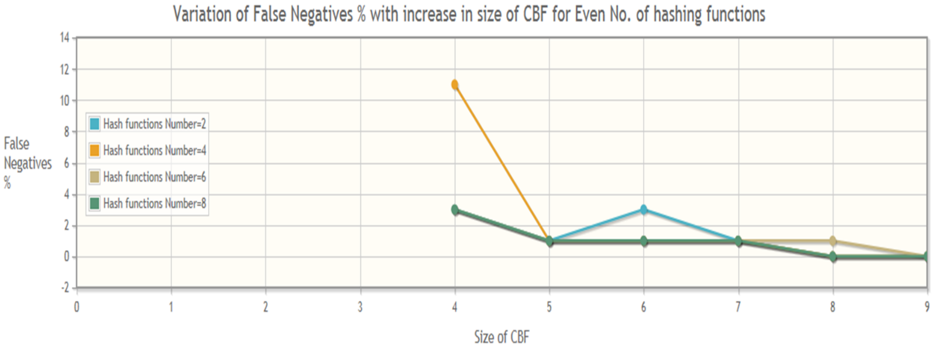

Fig. 4 variation of false negatives percentage with increase in size of CBF for different even number of hashing functions

D. Coverage

The coverage represents number of malicious transactions detected. As already discussed before each pair of size of CBF and number of hashing functions used represents a different configuration of IPS mechanism.10000 malicious transactions are submitted for every configuration of system and number of malicious transactions detected are recorded. The coverage percentage is calculated by using following formula.

Coverage % = ((Number of malicious transactions detected)*100) / (Number of malicious transactions submitted)

For every size of CBF coverage is plotted for five different number of hashing functions. Fig. 5 shows that for size of CBF greater than equal to 7 ; system can achieve coverage of about 9.76% , which is quite good result . Even for size of CBF equal to 6 the coverage is always greater than 95%.

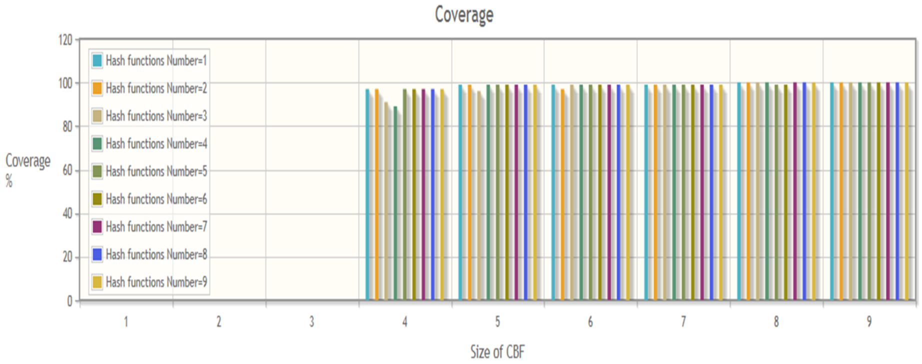

Fig. 5 Coverage

VI. Conclusion And Future Work

The security in the DBMS is one of the main concerns of the researchers now-a-days and there is an interest to develop the possible database intrusion detection systems. Our Proposed Database Intrusion Detection Model analyzes the database user behaviour pattern and detect malicious transactions based on user profiles, CBF, access security levels and time varying patterns through session and token management. We have implemented our proposed approach where valid transaction profile is generated by user profile manager using CBF and any abnormal deviation the threshold limits are detected by the intrusion analyser using anomaly detection.

In our future work, we intend to implement new algorithm with further security features related to access control and inference detection of complex dynamic queries.

References

[1] F.Bonomi, M.Mitzenmacher, R.Panigrahyi, S.Singh, G.Varghese, ”An Improved Constructionfor Counting Bloom Filters”LNCS 4168, pp. 684–695, Springer-Verlag Berlin Heidelberg ,2006.

[2] C.Pinzón, Y. De Paz, R.Cano.”Classification Agent-Based Techniques for Detecting Intrusions in Databases”LNAI 5271, pp. 46–53, Springer-Verlag Berlin Heidelberg, 2008:

[3] U.P.Rao, N.K.Singh, A.R.Amin, K.Sahu” Enhancing Detection Rate in Database Intrusion Detection System:“Science and Information Conference August,London,UK,2014.

[4] R.Jorge, S.J.Bernardino, M.Vieira.”Approaches and Challenges in Database Intrusion Detection:”ACMSIGMOD Record, Vol.43,No.3September ,2014.

[5] I.Singh, M.Kumar, ”A Proposed model for datawarehouse user behaviour using Intrusion detection system”, ACM SIGSOFT software engineering notes, Vol 37,No.6, Nov ,2012.

[6] J.Fonseca , M.Vieira , H.Madeira, “Online Detection of Malicious Data Access using DBMS Auditing”, SAC’08, , Fortaleza, Ceará, Brazil. Copyright ACM 978-1-59593-753-7/08/000 , March 16-20, 2008.

[7] M.Chagarlamudi, B.Panda, Yi Hu, ”Insider Threat in Database Systems; Preventing Malicious Uers Activities in Databases”, Sixth International Conference on Information Technology: New Generations, IEEE , 2009.

[8] A.Rezk, H.Ali, El-Mikkawy, S.Barakat, “Minimize the False Positive Rate In a Database Intrusion Detection System “, International Journal of Computer Science & Information Technology (IJCSIT) Vol 3,No 5, 2011.

[9] C.Chung , M.Gertz, Levitt, K. DEMIDS: “A Misuse Detection System for DatabaseSystems”. In Third Annual IFIP TC-11 WG 11.5 Working Conference on Integrity and Internal Control in Information Systems, Kluwer Academic Publishers, Springer, November , 1999.

[10] V.C.S.Lee, J.A.Stankovic, S.H.Son, ”Intrusion Detection in Real-time Database Systems Via Time Signatures.”In Proceedings of the Sixth IEEE Real Time Technology and Applications Symposium , 2000.

[11] Yi Hu, B.Panda:”A data mining approach for database intrusion detection.” Proceedings ACM Symposium on Applied Computing , NY,USA, pp 711-716 SAC, 2004.

[12] M.Doroudian, H.R.Shahriari ,“Database Intrusion Detection System for Detecting Malicious Behaviours in Transaction and Inter-Transaction Levels”, 7th International Symposium on Telecommunications IST , 2014.

[13] E.Bertino, R.Sandhu, ” Database Security- Concepts, Approaches and Challenges ”, IEEE Transactions on dependable and secure computing, Vol. 2, Mar 2005.