ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

Intrusion Detection System for Database with Dynamic Threshold Value

KHOMLAL SINHA
Khomlal_sinha@yahoo.co.in

TRIPTI SHARMA
triptisharma@csitdurg.in

ABSTRACT: ***In this paper, we propose an approach for database intrusion detection. Database management system are key component in the information field of most organization now days so security of DBMS has become more important several mechanism needed to protect data such as Authentication user privileges[1], encryption and Auditing have been implemented in commercial DBMS but still there are s various ways through which system may be effected by malicious transaction one of the mechanism to these databases is to use an IDS, in every database having a some attributes or columns that are more important to be sensed for malicious as compared to the other attributes in our work. initially we calculate support and dynamic threshold value and create some read and write rules among important data item in a database management system. . Any transaction that does not follow rules are identified as malicious . We also generate report for Audit log file to check how much transaction is malicious or non malicious.***

KEYWORDS:- Database security , Read and Write rule, Threshold value, Intrusion ,Intrusion detection,

I. INTRODUCTION:

In the last few years, Database systems form an indispensable component of the information system infrastructure in most organizations and are adopted as the key data management technology for daily operations and decision making in businesses. Data constituting such databases may contain invaluable sensitive information and organizations manage access to these data meticulously, with respect to both internal as well as external users. Any breach of security or intrusion in these databases perturbs not only a single user but can have devastating consequences on the entire organization [2]. Data Mining is one of the mechanism which refers to a collection of methods by which large sets of stored data are filtered, transformed, and organized into meaningful information sets [3][4]. It also applies many existing computational techniques from statistics, machine learning and pattern recognition.

According to a computer crime and security survey conducted by the Computer Security Institute (CSI) [5] in 2005, approximately 45% of the inquired entities have reported increased unauthorized access to information. The 2007 CSI computer crime and security survey [6] proclaimed financial application fraud as the leading cause of financial loss and found it more than doubled as compared to the loss estimated in the previous year. Another significant cause of loss was system penetration by outsiders. Additional findings in the current year's survey include the fact that, 59% of all respondents outlined insider abuse as the most prevalent security problem. These figures show the growing sophistication and stealth of attacks on databases. In addition to the substantial financial losses, intrusive attacks can also tarnish reputation of organizations, cause loss of customer confidence and even litigations. Thus, database intrusion detection is a significant problem which needs to be addressed urgent Security of Database system is compromised when an intrusion take place .An intrusion can be defined as " any set of action that attempt to compromise the integrity, confidentiality or availability of resource ".Intrusion detection analyzes unauthorized accesses and malicious behaviors and finds intrusion behaviors and attempts by detecting the state and activity of an operation system to provide an effective means for intrusion defense. Intrusion Detection Systems (IDSs) are the software or hardware products that automate this monitoring and analysis process[7]. In general, there are two types of attacks (i) inside and (ii) outside. Inside attacks are the ones in which an intruder has all the privileges to access the application or system but he performs malicious actions. Outside attacks are the ones in which the intruder does not have proper rights to access the system. He attempts to first break in and then perform malicious actions. Detecting inside attacks is usually more difficult compared to outside attacks. Intrusion detection systems determine if a set of actions constitute intrusions on There are mainly two models, namely, anomaly detection and misuse detection. The anomaly detection model bases its decision on the profile of a user's normal behavior. It analyzes a user's current session and compares it with the profile representing his normal behavior. An alarm is raised if significant deviation is found during the comparison of session data and user's profile. This type of system is well suited for the detection of previously unknown attacks. The main disadvantage is that, it may not be able to describe what the attack is and may sometimes have high false positive rate[10].

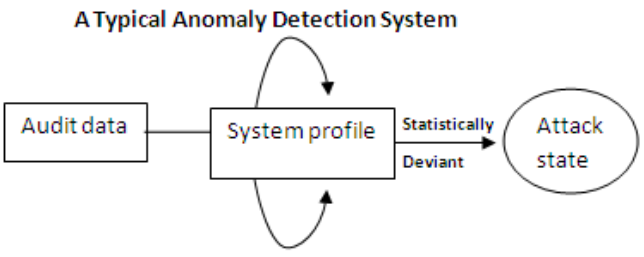

fig1. Anomaly detection system

In contrast, a misuse detection model takes decision based on comparison of user's session or commands with the rule or signature of attacks previously used by attackers. For example, a signature rule for the guessing password attack can be "there are more than 6 failed login attempts within 4 minutes". The main advantage of misuse detection is that it can accurately and efficiently detect occurrence of known attacks. However, these systems are not capable of detecting attacks whose signatures are not available[7][8].

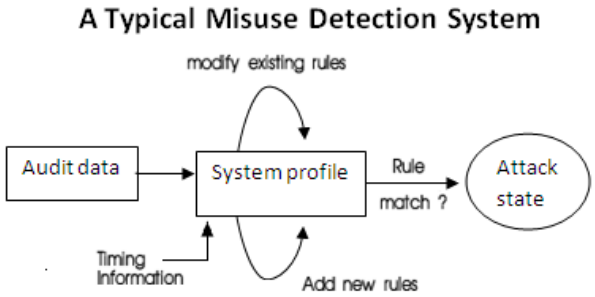

fig2. Misuse detection system

In this paper, we propose a new approach for database intrusion detection using a data mining technique which takes 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. This approach dependency among attributes in a database. The transactions that do not follow these dependencies are marked as malicious transactions.

The rest of the paper is organized as follows. In Section II, , We discussed the Motivation part . In Section III We described methodology and Section IV. We discussed results. In Section V. we discussed conclusion.

II. MOTIVATION

In now days ,database system is very important for most of the organization where information play a important role. database size has grown in different ways: the number of records, or objects in the database and the number of fields or attributes per object. As a result, it is difficult for administrators to keep track of whether the attributes are being accessed only by genuine transactions or not. By categorizing the attributes into different types based on their relative importance, it becomes comparatively easier to track only those few attributes whose unintended modification can potentially have larger impact on the database application security..Categorization of attributes enables the administrator to check only the alarms generated due to unusual modification of sensitive data instead of checking all the data attributes. Since the primary aim of a database intrusion detection system is to minimize the loss suffered by the database owner, it is important to track high sensitive attributes more accurately. It should be noted that, for tracking malicious modifications of sensitive attributes, we need to obtain some rules for these attributes. If there is no rule for an attribute, it cannot be checked. Since high sensitivity attributes are usually accessed less frequently, there may not be any rule generated for these attributes. The motivating factor for dividing database attributes into different sensitivity groups and assigning suitable weights to each group, is to bring out the read and write rules for important but possibly less frequent attributes. Once rules are generated for sensitive attributes, it is possible to check against these rules for each transaction. If any transaction does not follow the created rules, it is marked as malicious.

III. METHEDOLOGY

We divided our approach in three step :

A. Generate frequent Pattern from sensitivity attribute.
B. Check Write operation in frequent pattern.
C. Apply read and write rule .

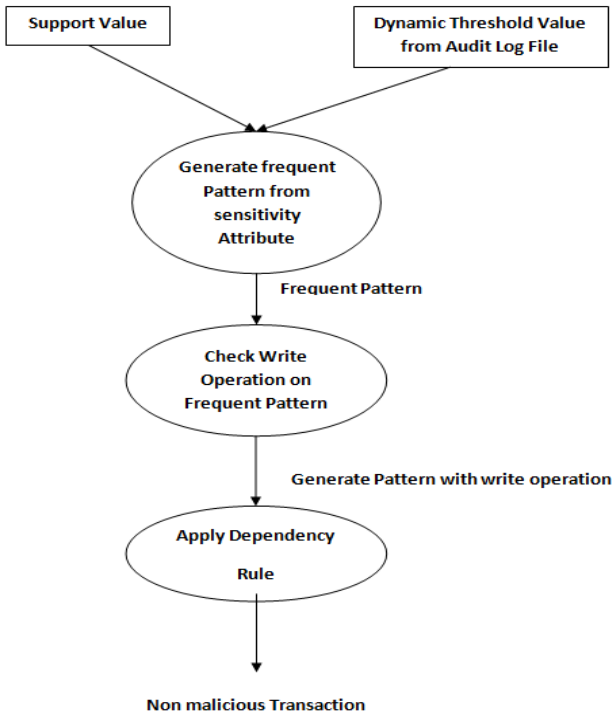

Fig. 3 Data flow diagram

We have taken a bank database schema and divided their different attribute on their uses basis . categories attribute in different importance level. and assign different weight on their importance means Higher importance value attribute having high weights. We have categorized the attributes in three sets : High Sensitivity (HS) attribute set, Medium Sensitivity (MS) attribute set and Low Sensitivity (LS) attribute set. The sensitivity of an attribute is dependent on the particular database application. For the same attribute say x, if $x \in HS$ then $W(xw) > W(xr)$, where W is a weight function, xw denotes writing or modifying attribute x and xr denotes reading of attribute x.*we also assigned the extra weight for write operation . we have taken bank database schema .*

TABLE I. BANK DATABASE SCHEMA

Table Name	Attribute Name
Customer	Name(9),Customer_id(10),address(4),Phone_no(1)
Account	Account_id(2),Customer_id(3),Status(7),Opening_dt(5),closeing_dt(6),amount(8)
Account_type	Account_type(11),Max-tran_per_month(12),

TABLE II –Different sensitivity of attributes

Sensitivity	Attribute	Weights
LS	1,2,3,4,9,10,11	1
MS	5,6	2
HS	7,8,12	3

A. Generate frequent Pattern .

in this step , we assign weights to each pattern based on the sensitivity groups of the attributes present in the pattern . The weight assigned to pattern the same as the weight of the most sensitive attribute present in that pattern. The weight assigned to each sequence also depends on the operation applied on the attributes. after assigned weight we calculate the support value for given pattern by following formula.

support value for current transaction = (n * wp) / N

where wp is a weight of pattern
n is a no. of same pattern in N
N is a total transaction in audit log file

To calculate Dynamic Threshold value from the audit log file . Dynamic threshold value depends the size of audit log file . we have design an algorithms for it . algorithm is given below

Algorithms for dynamic threshold value:

```
call DynamicThreasholdValue()
{  //check the size of log file ,if it is initial
 do size = 0
//otherwise calculate the size and initilize into variable size
   if(size = = 0 )
  // do start incrementing the value then do threshold value low
If (size > 0 && size < normal)
//if this condition satisfy make this variable constant
 //the check the threshold value if it is more then 50% then make a value final
if(size> normal)
   // do start decrementing then decrementing the threshold value
   }
```

now we generate frequent pattern, If support value is minimum to dynamic threshold value means it is a normal transaction and do not need to go next step. but support value is maximum to the Dynamic Threshold value then given pattern is a frequent pattern and forward it to next step .

B. Check Write Attribute in frequent pattern

frequent pattern must contain at least one write operation. All the pattern that do not have any attribute with write operation, are not used for next pattern generation. A pattern that contains a single attribute does not contribute to the generation of dependency rules and hence will be ignored too.

C. Applied Dependency rule

There are two types of rules, namely, read rules and write rules.A read rule of the form $ajw \rightarrow a1r, a2r ..., akr$ implies that attributes $a1$ to ak are read in order to write attribute aj . Write rule of the form $ajw \rightarrow a1w, a2w, ...akw$ implies that after writing attribute ajw, attributes $a1w, a2w, ...akw$ are modified. we create some rule on their use basis **.**

TABLE III - Read and Write Rules

Name(w)→customer_id(r),amount(w)→status(r),status(r),account_id(r),address(w)→balance(r),address(w)→status(w),name(w)→status(w)

The generated rules are used to verify whether an incoming transaction is malicious or not. If an incoming transaction has a write operation, it is checked if there are any corresponding read or write rules. If the new write operation does not satisfy any of these rules, it is marked as malicious .

IV. EXPERIMENTAL RESULTS

Our work focused on the development of a database intrusion detection system keeping sensitivity of the attributes into consideration. The system has been developed using

Java(JDK 1.6.0-040) as front end and MYSQL Server 5.0 as the backend database. we have used dynamic threshold value from audit log file so that our result is more flexible and accurate . every time threshold value changed and compare with support value. We defined some dependency rules if any transaction not follow this rule mark as malicious transaction. We also generate a report of audit log file to check how much transaction is malicious or non malicious after every transaction .

name	customr_i	adddress	phone_no
bhupesh	5001	aadarsh nagar durg	8109119061
khomlal	5002	gayanagar durg	9589356960
Sandeep Das	5003	aarya nagar bhilai	9713122558
Abhishek shrivastava	5004	matri kunj bhilai	8817394917
Deman Dhruv Singh	5005	Shankar Nagar, Near	8305053102
Rakesh Yadev	5006	khursipar bhilai	9407676785
Ajay kumar verma	5007	kasaridhih durg	9981329827
Mohnish miri	5008	kelabadi durg	9406207410
Rahul kumar Sharaf	5009	Block-4/B, Sector-6,	9098122351
Shravan Ku. Sahu	5010	Vishrampuri, Thana-	9302332042
Manish Kumar Agrawal	5011	PolshayPara,Durg	88-2210781
Ku.Priya Tiwari	5012	Street- WMR, Sec-4,	9302332042
Ritu Verma	5013	279/C, Risali Sector	9434183444
Nikita Khandelwal	5014	kadambari nagar bhil	9630519809

Fig 4. Snapshot of customer table

account_id	customer_id	status	open_dt	close_dt	balance
49000	5000	true	28/01/2000	30/12/2012	1000
49001	5001	true	29/01/2000	22/01/2013	10786
49002	5002	false	30/01/2000	12/04/2011	0
49003	5003	true	05/02/2000	21/03/2013	1000
49004	5004	true	10/02/2000	24/09/2017	10000
49005	5005	true	20/02/2000	20/03/2013	20000
49006	5006	false	20/03/2000	02/02/2014	0
49007	5007	true	04/04/2000	05/02/2013	10000
49008	5008	true	08/04/2000	06/04/2014	20000
49009	5009	false	05/05/2000	20/04/2011	0
49010	5010	false	08/05/2000	10/05/2009	0
49011	5011	true	10/06/2000	12/12/2012	3000
49012	5012	false	11/07/2000	11/11/2011	0
49013	5013	false	11/07/2000	04/05/2008	0
49014	5014	true	12/07/2000	14/05/2012	500
49015	5015	true	17/07/2000	15/08/2014	25000
49016	5016	false	18/07/2000	14/03/2012	0
49017	5017	true	25/08/2000	15/03/2014	200000
49018	5018	true	25/09/2000	15/03/2014	23000

Fig 5. Snapshot of Account table

Query:-update account set address =gayanagar durg whrere status =true;

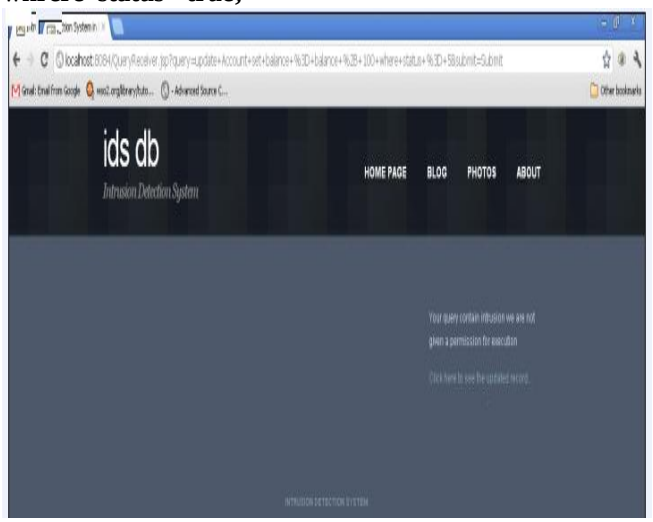

Fig 6. Snapshot of malicious transaction

above transaction is a malicious types of transaction because when we read status attribute we cannot update address . when we put a such type of query on text box after submission we will get a message your transaction is malicious type transaction not proceed.

Query:-update account set balance =balance +1000 where account_id=49027;

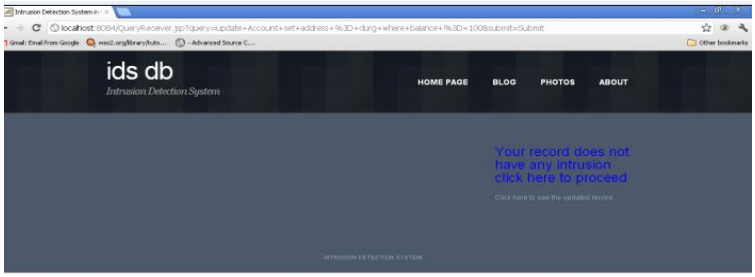

Fig 7. Snapshot of non malicious transaction

Above transaction is a non malicious type transaction because it follow read and write rules .we can see that after read account id and current balance we can update the balance.when we put query on text box and press submit button we will get one message that your query does nort have any intrusion click here to proceed when we click a button we will get update database record.

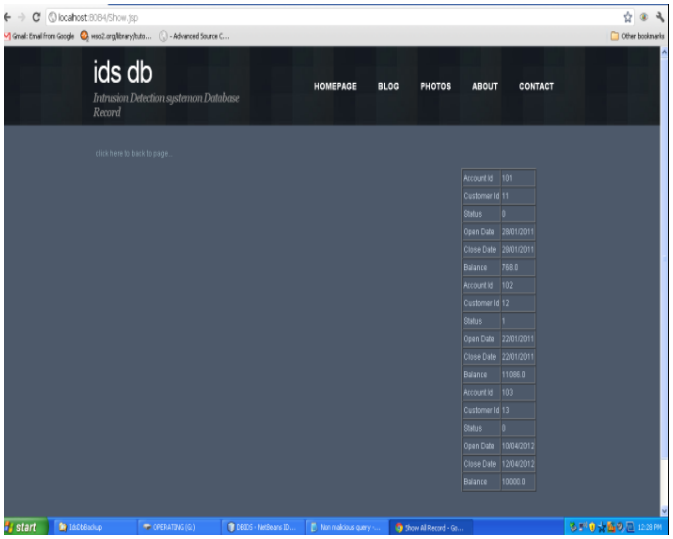

Fig8. Updated database

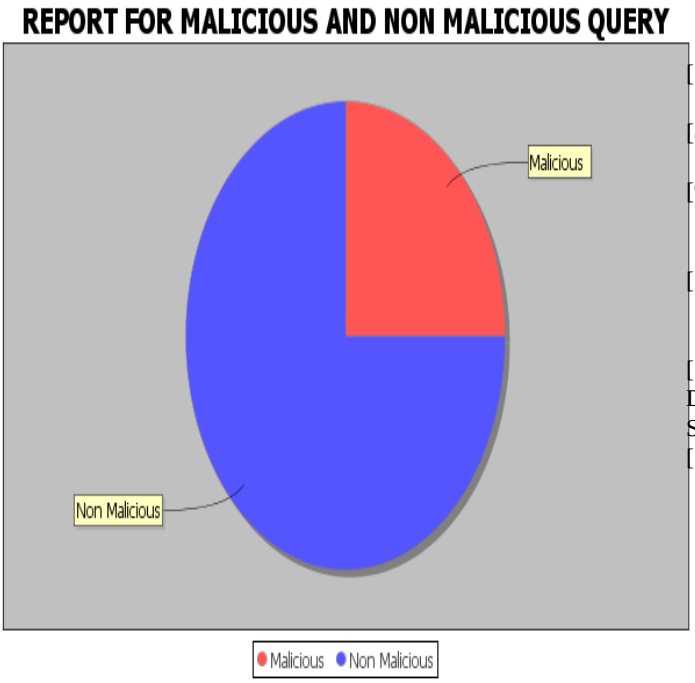

Fig.9 Report of audit log file

In above figure generate a report for how much Transaction in Audit log file is malicious or non malicious. above report will be updated with every transaction.

V. CONCLUSIONS

Intrusion detection mechanisms play a crucial role in the security landscape of a organization. In this paper we have focused Intrusion detection system for database . In our work , initially calculate threshold value from audit log file instead of constant value we checked the given transaction is a malicious and non malicious types with help of some dependency rule which we have created . we also generate a report of audit log file for how much transaction in audit log file is malicious or non malicious. After every Transaction audit log file will be updated.

REFERENCES

[1] Y. Hu, B. Panda, "A Data Mining Approach for Database Intrusion Detection", Proceedings of the ACM Symposium on Applied Computing, pp. 711-716 (2004)

[2] E. Bertino, R. Sandhu, "Database Security − Concepts, Approaches, and Challenges", *IEEE Transactions on Dependable and Secure Computing*, Vol. 2, No. 1, Pages 2-19, Jan.-March 2005.

[3] J. Han, M. Kamber, Data Mining: "Concepts and Techniques", Morgan Kaufmann Publishers (2001).

[4] U. Fayyad, G. P. Shapiro, P. Smyth, The KDD Process for Extracting Useful Knowledge from Volumes of Data, Communications of the ACM, pp. 27-34 (1996).

[5] L. A. Gordon, M. P. Loeb, W. Lucyshyn and R. Richardson, "2005 CSI/FBI Computer Crime and Security Survey", http://i.cmpnet.com/gocsi/db area/pdfs/fbi/FBI2005.pdf.

[6] R. Richardson, "2007 CSI Computer Crime and Security Survey",http://i.cmpnet.com/v2.gocsi.com/pdf/CSISurvey2007.pd.

[7]Tripti Sharma , Khomlal Sinha "2011 "IJEAT Journal Intrusion Detection System Technology"

[8] R. Bace, P. Mell, Intrusion Detection System, NIST Special Publication on Intrusion Detection System (2001).

[9] Aurobindo Sundaram , An Introduction to Intrusion Detection,Crossroads, Volume 2, Issue 4, Pages: 3 – 7, 1996

Teresa F. Lunt. A survey of intrusion detection techniques. Computers & Security, 12(4): p.p. 405-418, 1993.

[10] Aurobindo Sundaram , An Introduction to Intrusion Detection, Crossroads, Volume 2, Issue 4, Pages: 3 – 7, 1996

[8] Sandeep Kumar. Classification and Detection of Computer Intrusions. Ph.D. Dissertation, August 1995.

[9] E. Biermann, E. Cloete and L. M. Venter, A comparison of Intrusion Detection systems, Computers & Security, Volume 20, Issue 8, Pages 676-683, December 2001,

[10] Lubomir Nistor, Rules definition for anomaly based intrusion detection, 4th National Information Systems Security Conference, 1999.

[11] S.Y. Lee, W.L. Low, P.Y. Wong, Learning Fingerprints for a Database Intrusion Detection System, Proceedings of the European Symposium on Research in Computer Security,

[12] www.datarepository.com

Khomlal Sinha, received B.E. (Information Technology) in year 2005 and in pursuit for M.Tech. (Computer Sc.) from Chhatrapati Shivaji Institute of Technology (CSIT), Durg, Chhattisgarh, India, His interests are Digital Image Processing, Operating Systems and Data Mining. Also he is having Life Membership of Indian Society of Technical Education, India (ISTE) and Institutional Member of Computer Society of India (CSI).

Prof. Tripti Sharma, received B.E. (Computer Sc.) and M.Tech. (Computer Sc.) in the years 2002 and 2010 respectively. Currently working as Associate Professor and Head in the Department of Computer Science & Engineering at Chhatrapati Shivaji Institute of Technology (CSIT), Durg, Chhattisgarh, India, Her interests are Digital Image Processing and Data Mining. Also she is having Life Membership of Indian Society of Technical Education, India (ISTE), Membership No- LM 74671 and Institutional Member of Computer Society of India (CSI). Membership No-N1009279.