Dilip kumar Choubey, Prof. Joy Bhattacharjee, Prof. Roopali Soni / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 1, January -February 2013, pp.1437-1444

"Loss Minimization of Web Databases by Fine Grain Approach"

Dilip kumar Choubey, Prof. Joy Bhattacharjee, Prof. Roopali Soni
M.Tech IVSem O.C.T Bhopal India
Department Of C.S.E O.C.T Bhopal India
Head, Department of C.S.E O.C.T Bhopal India

Abstract

Information is the most valuable asset for organizations. One of the goals of organizations is to share their data and at the same time to enforce their policies. Web database is a combined production with database technology and Web technology. Web database is placed on the Internet, there are many security problems. The secrecy and the integrity are two important demands of security system. When database access control and the network security are addressed separately, the security systems are not optimized sufficiently as a whole. Fine-grained access control (FGAC) must be supported by web relational databases to satisfy the requirements of privacy preserving and Internet-based applications. We propose a model of integrating network security with criterion based access control to handle network security and the fine grained Web database access control simultaneously. We have implemented our model in college database and performance is evaluated. Whenever any unauthorized user altered our data a system called Web-Secure report to the authorized user via E-mail or Short Message Service (SMS). The implementation results show that how our model is suitable for web database security.

Keywords: Fine grained access control, Web database security, multiple policies, Privacy preservation, Network security, access control

1. Introduction

Internet users interact with and use web applications every day for a wide spectrum of tasks, ranging from online banking to social networking, and everything in between. Security in database has become an important problem because of the large amount of personal data, which is tracked by many business web applications. Web database is combination of database and web technology. Web database is placed on the Internet, there are many security problems. Web and distributed databases play[02] the key role in most of these Web applications and thus it is critical to protect them from unauthorized access and malicious attacks. One of the key components of every web application and arguably the most important in terms of security [19] is the web application's database. The web database is the heart of any data-driven web application, and must be guarded from numerous types of malicious attacks. Security is a major concern in the application of web database techniques to datasets containing personal sensitive or confidential information. To address this issue, a more efficient and flexible security mechanism is required to systematically authenticate users, control network traffic [14], and provide efficient fine-grained access control. Traditional policies treat tables or columns as the basic access control unit [03].

As security has gained significant importance, organizations have been forced to protect individual preferences and comply with many enacted privacy laws. This has been a strong driving force for access control in relational databases [07]. Traditional relation level access control is insufficient to address the increasingly complex requirements of access control policies where each cell in the relation might be governed by a separate policy. In order to address this demand, we are in need of a more fine grained access control scheme, at the row-level or even the cell-level.

Security is an integrative concept that includes the following properties [1]: **confidentiality** (absence of Unauthorized disclosure of a service or piece of information), **authenticity** (guarantees that a service or piece of information is authentic), **integrity** [01](protection of a service or piece of information against illicit and/or undetected modification), and **availability** (protection of a service or piece of information against possible denials of service caused maliciously) [04].

Current day database applications, with large numbers of users, require fine-grained access control mechanisms, at the level of individual tuples[17], not just entire relations/views, to control which parts of the data can be accessed by each user. The authorization rules (security policy) may be different for different Web and distributed databases. It is convenient to implement flexible fine-grained access control for each database based on different authorization rules [13].

Currently, network security and database security are often addressed separately and therefore the security system is not optimized properly as a whole.

We propose a model of integrating network security with criterion based access control to

handle network security and the fine grained Web database access control simultaneously.

We consider the issue of security of the web database at the database layer and network layer. Our main emphasis is at database layer[05] where we have applied fine grained access control to achieve security at row level or even cell level.

We have implemented our model in college database and performance is evaluated. Whenever any unauthorized user altered our data a system called WebSecure report to the authorized user via email or SMS. The implementation results show that how our model is suitable for web database security.

The rest of the paper is divided as follows. Section 2 present the security related work. Section 3.1 covers the proposed algorithm. Section 3.2 present implementation. Section 4 Present Experimental result and performance evaluation. Section 5 concludes the paper and also present future work.

1.1 Motivation

In the past decade alone, the widespread availability of broadband internet connections coupled with the relatively low cost of internet-capable computers has led to a boom in the number of internet users. As entire populations log on to the internet, the amount of physical data stored by web-based applications continues to soar. Internet users interact with and use web applications every day for a wide spectrum of tasks, ranging from online banking to social networking, and everything in between. Web database is a combined production with database technology and web technology. Data security [18] is a major issue in any web-based application. Web database was placed on the Internet, there are many security problems. Real world web databases have information that needs to be securely stored and accessed. Web applications are becoming increasingly commonplace and the database can be easily accessible. In the old web database system, some database rights were granted to legal users. Many applications are developed with loosely-typed scripting languages [06] and make use of a single database user with full permissions, a so-called administrator user. Information is the most valuable asset for organizations. The information disclosure from such databases may have very serious impact on organization business. It is important to properly handle network and web database security issues [08] including authentication, denial of service, and fine-grained access control. So new security mechanism and access control [3] approaches for databases and especially for web databases have become a dire necessity. But with the development of web systems, the number of attacks on databases increased and it has become clear that their security mechanism and access control mechanism is inadequate for web-based database systems.

2. Related work

Fine-grained access control was first introduced as a part of the access control system in INGRES by Stonebraker and Wong (1974), which was implemented by query modification technology. The basic idea of query modification is that before being processed, user queries are transparently modified to ensure that users can access only what they are authorized to access (Bertino *et al.*, 2005; Wang *et al.*, 2007) [20][15].

Views are used to specify and store access permission for users. When a user submits a query, DBMS first finds all views [08] whose attributes include the attributes of the issued query, and then add the predicates of these views to the predicates of the original query to form a new modified query, which will be carried out.

Recently, work on the policy for preserving privacy has boosted the research of FGAC (Agrawal *et al.*, 2002; Bertino *et al.*, 2005). Bertino *et al.* (2005) [20] presented a privacy preserving access control model for relational databases, which needs a basis of FGAC in relational databases. Nevertheless, they did not describe how to implement the model.

LeFevre *et al.* (2004) proposed a practical approach to incorporating privacy policy enforcement into an existing application and database environment where the implementation of FGAC at cell level was provided.

All works described above focused mainly on the enforcement of FGAC, and did not provide a FGAC Model which supports many access control policies [11]. Less work has been done with the FGAC model. The work of Agrawal *et al.* (2005) [20]and Barker (2008) suffered from specific aforementioned limitations.

Chaudhuri *et al.* (2007) also extended SQL language to support fine-grained authorization by predicated grants. Not only the column- and cell-level authorizations, but also the authorizations for function/procedure execution were supported. Moreover, they designed query defined user groups and authorization groups to simplify the administration of authorizations.

Olson *et al.* (2008) presented a formal framework for reflective database access control policies where a formal specification of FGAC policies was supported by Transaction Data-log. The security analysis [16] was also provided. Moreover, they enforced policies by compiling policies in Transaction Data-log into standard SQL views (Olson *et al.*, 2009). The shortcomings are that the negative authorization and multiple policies at fine granularity (which are the major contributions of our model) were not taken into account.

Kabra *et al.* (2006) [09] considered two different aspects of FGAC: efficiency and information leakage of enforcement of FGAC. Using query modification to enforce FGAC, there

may exist redundancies in the final executed queries because of the same predicates between the FGAC policies and the queries issued by users. These redundancies include not only cheap comparisons, but also expensive semi-joins, which would increase the execution time.

Kabra *et al.* (2006) also considered the potential of information leakage through channels caused by exceptions, error messages, and user defined functions. For remedying the two problems, they proposed methods for redundancy removal, the definition of safety query plan, and the techniques to generate safe query plans.

Wang *et al.* (2007) proposed a correctness criterion of FGAC for databases, which contains three properties: secure, sound, and maximum. They argued that any algorithm used to implement FGAC must be sound and secure, and should strive to be maximum. They also pointed out that no algorithm exists that is both sound and secure. Then, they proposed an algorithm that is sound and secure. In this paper, we do not consider these aspects. There is another important related work.

Bertino *et al.* (1997) proposed an extended authorization model for relational databases, which supports negative authorization. This work inspired us to extend the FGAC model to support negative authorization. The main difference between their work and ours is the granularity of negative authorization: the model they proposed can support only negative authorization at coarse granularity (tables, views), but our model can express negative authorization at finer granularity (rows, columns, or cells).

Oracle's Virtual Private Database (VPD) model [9] supports finegrained access control through functions that return strings containing predicates. Oracle virtual private database (VPD) also uses query modification to implement FGAC (Oracle Corporation, 2005). VPD supports FGAC[10] through functions written as stored procedures which are associated with a relation. When a user accesses the relation, the function is triggered to return predicates, and the database rewrites the SQL statement submitted by the user to include these predicates. For providing enhanced access control, in addition to row level access control, column-level VPD [12] has been added to Oracle to provide column-level access control, which in turn associates functions with columns. A function is associated with each relation, and when invoked returns a string containing predicates that enforce fine-grained access control; the function takes as input the mode of access and an application context which includes information such as user-id of the end user.

3 Our approach

The detection of an inconsistency by the detection engine can be considered as a system event. The attributes of the anomaly, such as user, role, SQL command, then correspond to the environment surrounding such an event. Intuitively, a policy can be specified taking into account the anomaly attributes to guide the response engine in taking a suitable action.

Fine-grained access control (FGAC) must be supported by web relational databases to satisfy the requirements of privacy preserving and Internet-based applications. We propose a model of integrating network security with criterion based access control to handle network security and the fine grained Web database access control simultaneously.

Inconsistency Attributes
User The user associated with the request
Role The role associated with the request
Source IP the IP Address associated with the request
Date and Time Date/Time of the anomalous request.
Client App The client application associated with the request.

3.1 Proposed Algorithm

Algorithm
Input: user ***U***, relation R, action A, database D.
Output: the combined Fine Grained Access Control policy *P*out.

ON ANOMALY DETECTION
IF ROLE != USERROLE and Source IP IN NETWORK and OBJECTTYPE=table
And SQLCOMMAND IN{INSERT,UPDATE,DELETE}
THEN SUSPEND
CONFIRM REAUTHENTICATE
ON SUCCESS
Access to web database according to Policy
ON FAILURE
Abort, Disconnect and response report to authorized user

RS←∅;
PStemp←∅;
PSrole←∅;
PStemp=Get all FGAC policies (U, R, A, D);
PD=Inter Section of (PStemp);
RS=Get the Role Set (U, D);
for all r in RS **do**
PStemp←∅;
PStemp=Get RFGAC Policies Set (r, R, A, D);
PSrole←Inter Section of (PStemp);
end for
PR=Union of (PSrole);
Pout=PD∧PR.

3.2. Implementation

The machine on which the experiments were performed has a system memory of 2GB, clock

speed of 2.50 GHz, cache size of 512 KB, bus speed of 100 MHz, and a block size of 1 KB. All the experiments were run on the ORACLE 11g database management system. The tables used in our experiments are of Educational Institute. The tables used are students (eno, name, tos), faculty (id, name, dept, courseid), courses (courseid, coursename), grades (eno, courseid, grade), dept (deptno, dname), emp(empno, ename, job, mgr, hiredate, sal, comm, deptno), bonus (ename, job, sal, comm), registration (eno,courseid), salgrade (grade, losal, hisal), users (username, domainname, ipaddress, privatekey), tmp(eno, grade, newgrade).

The WebSecure is a system, which we are developing to detect and report inconsistencies and security breaches in a web database. It runs on a separate site (typically different from the one on which the application runs and the one on which the main database resides) monitoring the access to the database indirectly. It runs periodically at pre-decided intervals and checks for the consistency of the database relations and reports any discrepancies. It also has a local copy of original data, which can be used to restore the database to the consistent state.

We built a package in Java. Any application that uses this package should call the appropriate function and send the query along with the parameters. We create a table containing the parameters of the queries passed, one table per query type. And similarly we have one result table per query type. When a new query is received at the Web-Secure, we detect if it is a duplicate and insert it into the table accordingly. We then run the query on the main database and store the result we get in the corresponding result table. Also each set of parameter and result tables are identified by the parameters in the query.

We get the entire main database table to the Web-Secure site and join it with the local parameter table and detect the inconsistencies. In order to optimize the activity performed by the Web-Secure, we run as less number of queries as possible during the periodical activity of verifying the consistency of the database. For every type of query, i.e., for set of queries of the same type (meaning same number and type of parameters), the Web-Secure runs a single query in order to check the results with the local copies. As the parameters differ only in values, we perform a join of the parameter table with the database table to get the results and compare them with the locally stored result tables containing original contents.

Our earlier model of Web-Secure maintains a copy of the frozen query results and periodically, it transfers the whole relation of the main database to the Web-Secure site and does a join of the parameter table with this relation as part of the query to detect differences between these copies. Now, instead of shipping the whole relation we can compute a checksum on the tuples of the relation in the main database, ship the checksum to the Web-Secure site, compute a similar checksum of tuples in the local copies and compare the two checksums. If they do not match, it can be reported as a security breach. In fact, if detecting inconsistency is all that is required, we could do away with creating the result tables. Instead of storing the results as a table, all we need to do is store the checksum of the query result. During the periodic check, we compare this checksum with the calculated checksum of the tuples in the main database relation.

We have incorporated this facility into our Web-Secure code. We use MD5 checksums; we compute checksum of every tuple that is a 16-byte value and compute the exclusive-OR of all the checksums. This is the value that is transferred to the Web-Secure, where a similar procedure is followed to compute the checksum, and compared with the computed checksum of the local copy. This reduces the network traffic significantly, which would otherwise have been very high, if the whole relation was transferred periodically to the Web-Secure site.

4. Experimental Results and Performance Evaluation

We have Performed the row level or Cell Level Security on the table as mentioned below and The Data Dictionary is also given below of each table:

DEPT

DEPTNO	NUMBER(2)
DNAME	VARCHAR2(14)
LOC	VARCHAR2(3)

Department table consists of Department Number(DEPTNO), Department Name(DNAME), Location(LOC).

COURSES

COURSEID	VARCHAR2(20)
COURSENAME	VARCHAR2(20)

Courses table consists of Courses Identity (COURSEID),Courses Name(COURSENAME).

EMP

EMPNO	NUMBER(4)
ENAME	VARCHAR2(10)
JOB	VARCHAR2(9)
MGR	NUMBER(4)
HIREDATE	DATE
SAL	NUMBER(7,2)
COMM	NUMBER(7,2)
DEPTNO	NUMBER

Employee table consists of Employee Number (EMPNO), Employee Name (ENAME),Job(JOB), Manager(MGR),Hiredate(HIREDATE),Salary(SAL),Commission (COMM),Department Number (DEPTNO).

GRADES

ENO	VARCHAR2(20)
COURSEID	VARCHAR2(20)
GRADES	VARCHAR2(20)

Grades table consists of Employee Number (ENO), Course Identity(COURSEID),Grades(GRADES).

BONUS

ENAME	VARCHAR2(10)
JOB	VARCHAR2(9)
SAL	NUMBER
COMM	NUMBER

Bonus table consists of Employee Name (ENAME), Job(JOB),Salary(SAL),Commission(COMM).

STUDENTS

ENO	VARCHAR2(20)
NAME	VARCHAR2(20
TOS	VARCHAR2(20

Students table consists of Employee Number(ENO), Name(NAME),Type of Course Name(TOS).

REGISTRATION

ENO	VARCHAR2(20)
COURSEID	VARCHAR2(20)

Registration table consists of Employee Number (ENO), Course Identity (COURSEID).

FACULTY

ID	VARCHAR2(20)
NAME	VARCHAR2(20)
DEPT	VARCHAR2(20)
COURSEID	VARCHAR2(20)

Faculty table Consists of Identity(ID),Name (NAME),Department(DEPT),Couse Identity (COURSEID).

SALGRADE

GRADE	NUMBER
LOSAL	NUMBER
HISAL	NUMBER

Salary Grade table consists of Grade (GRADE), Low Salary (LOSAL) ,High Salary(HISAL).

USERS

USERNAME	VARCHAR2(20)
DOMAINNAME	VARCHAR2(20)
IPADDRESS	VARCHAR2(20)
PRIVATEKEY	VARCHAR2(20)

Users table consists of User Name (USERNAME), Domain Name (DOMAIN NAME), IP Address (IPADDRESS), Private Key(PRIVATEKEY).

TMP

ENO	VARCHAR2(20)
GRADE	VARCHAR2(2)
NEWGRADE	VARCHAR2(2)

Temprorary table consists of Employee Number (ENO),Grade(GRADE),New Grade(NEWGRADE).

There are eleven tables in our concept of fine grain approach.Now we will performed our approach on the grade table.The fine grain approach is applied where the database tables are used not frequently.The example of our approach can be applied on grade system and if tables commercially, it can be applied on Land Registry,fixed deposits of banks etc.

Now We have Performed the Operation on the table grade. Before doing the Operation The Ticket authorization Serever will check for the authorized user.

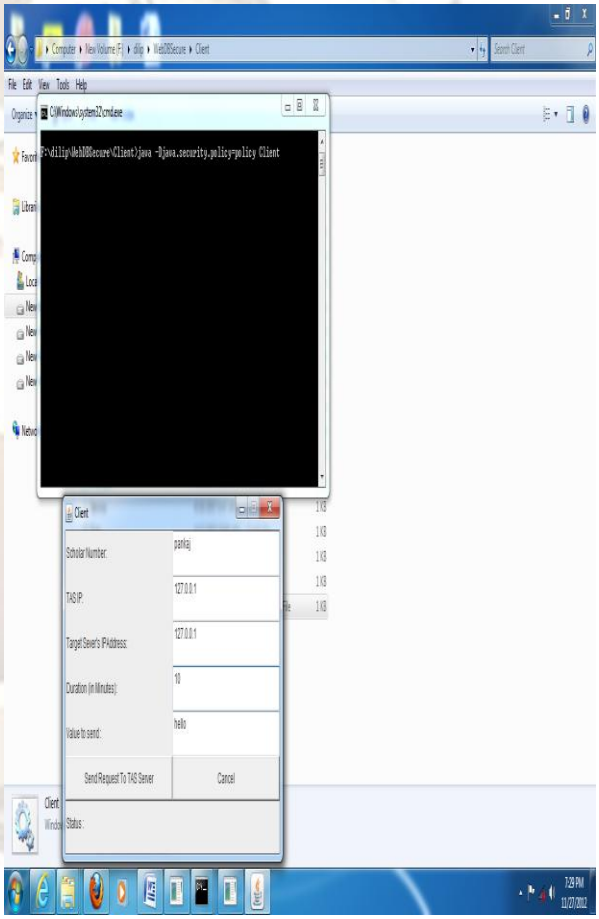

Fig. 1

Here, authorized user mean he should be registered in database and should be aware to Server IP Address.If this true,Then Ticket Authorization Server will allow to perform the operation to authorized usres.So that We assume the application Software are secure because as we may see in

Fig.1,the user should be registered as well as should be aware to server IP address.

We assume the sense of following parameter:
Scholar Number : Registerd User in database.
TAS IP :Client IP
Target Servers's IP Address:Server IP Address (Where we have to send the value).
Duration(in minutes):for how much time.
Value toSend:What we have to send
and then send request to TAS server.
So, To Perform a task User should be aware for the above parameter .

If unauthorized User will bypass directly/logged to this database system Then he may alter the useful data ,So it is clear that Database system are inseure so We are providing the security at database System. Here We are providing the security on grade table.

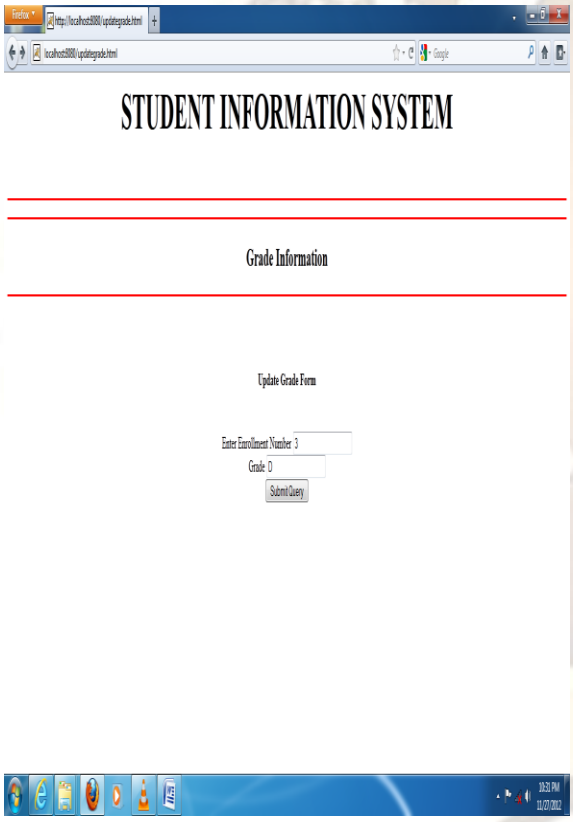

Fig. 2

In the Student Information System, meaningfully unauthorized user Will alter data on grade table, so We have Provided the Security on grade table. Only one of the enroll grade will be alter at a time by unauthorized user. Then instantly The Web Secure will report to authorized user by email/sms with old value and updated value, and then authorized user may act. So here we are minimizing loss of databases by Fine Grain Approach. The above snapshot describes the Fig. 2 mentioned concept here only one cell of one table is being modified. Our approach gives the logoff data that is when modified two of modification old value and also the updated value.

In the Fig. 2 named as Grade Information We are changing the grade table through cell level approach. Here only authorized user is able to change the grade but If there is an impersonification, the impersonified person can only change a single data therefore the whole table is secure from the attack of an intruder. This is the concept behind of our approach.

If Unauthorized User gets an entry into the database Server then he would not be able to manipulate the Whole table, only One Cell of the data may be modified that too will be received by authorized user and he would get the information by mail & SMS Which Would describe the old and updated value.

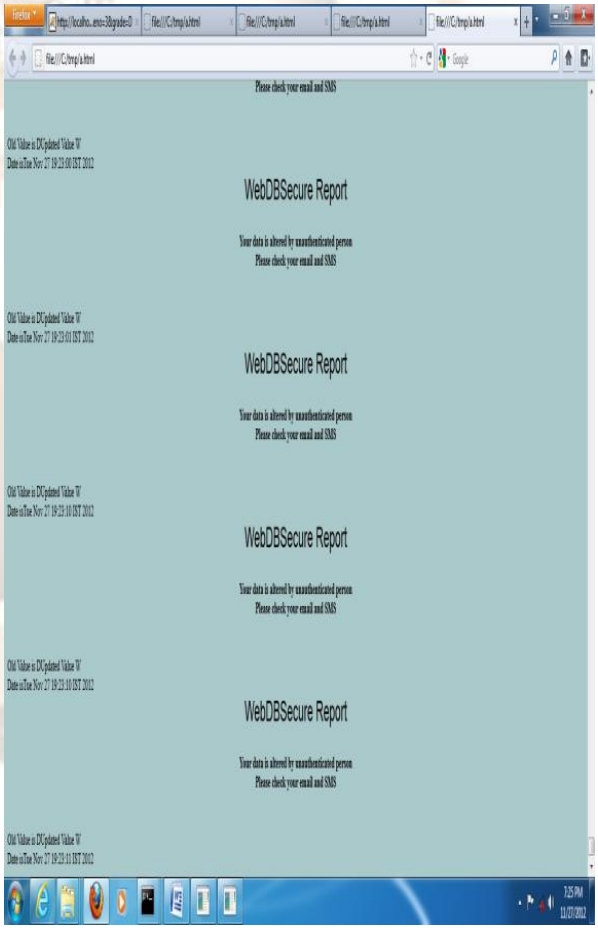

Fig. 3

So, As we may See in Fig. 3 that only one cell will be alter at a time and instantly The Web Secure will report to authorized user by email/sms with old

value and updated value and then authorized user may act. So,It may be noted that that we are minimizing the Loss of database using Fine Grain Approach. The performance can be evaluated hypothetically.

The graph i.e., shown below evaluates the performance between users and number of interactions.

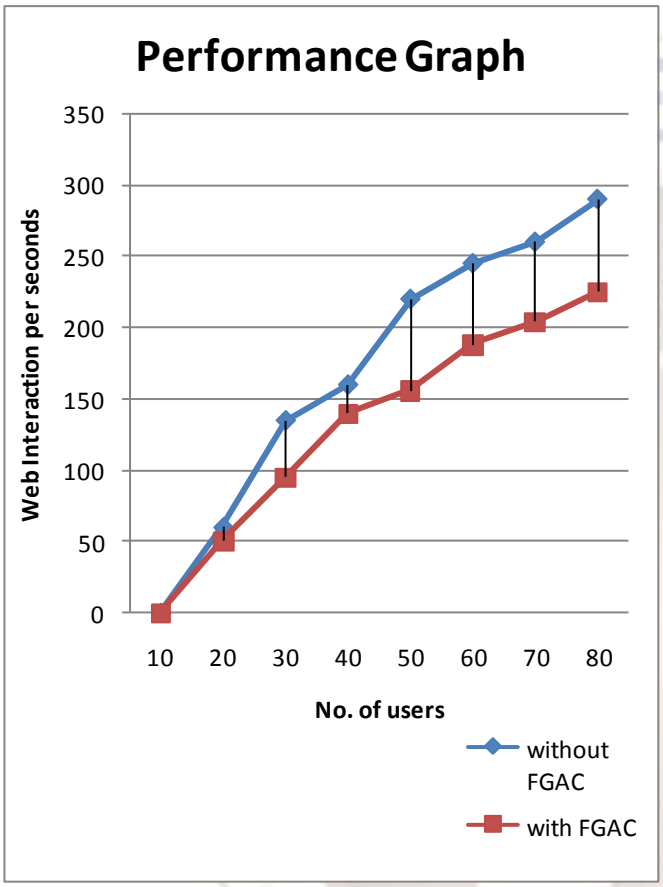

Fig. 4

In the Fig. 4 performance graph we measure the performance of FGAC and without FGAC with Web interaction per second Vs no. of users. As we may see in the Fig. 4 the number of users in any number interacting / hitting more web interaction per second in without FGAC and less in with FGAC so it is clear that we may retrieve less information in case of FGAC because less interaction/hitting with web interaction per second. So it may be noted it is providing security to the database server.

The chart which has been given below is in histogram format. The number of users are same in the X-axis and number of seconds is in the Y-axis.

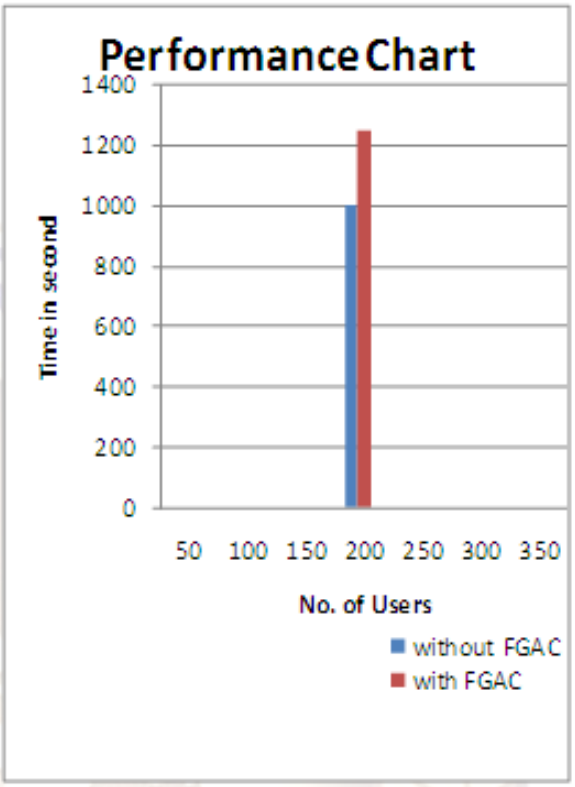

Fig. 5

In the Fig. 5 performance chart as we may see that we are comparing the performance of without FGAC and FGAC with time in second versus number of users.

Number of users in even same number we may see that more time is taking in case of with FGAC while less in without FGAC with respect to web interaction per seconds so again it is clear that we are getting more security in case of FGAC because we are working on cell level therefore time required is more and information retrieval is less.

5 Conclusion and future work

Web database is combination of database and web technology. Security of data stored in the web database is of prime importance in today's world with growing E-business. In order to preserve data privacy, we assume that no one except the data owner or authorized users have the right to access the original data. We propose an authentication mechanism when certain anomalous actions are executed against critical system resources such as anomalous access to system tables.

The system was considered extremely useful by the administrators. These results clearly show that our approach is very accessible to most administrators and can be of extreme importance in helping them to become more aware of the security

flaws existent in the configuration of the environments that they manage. The termination of the users requests at the early stage avoids to unnecessarily processing the requests further. The implemented system can be applied to many areas such as education, finance, marketing, health care, government, and military. We implemented our system in Education domain. The propose policy mechanism and access control mechanism is applicable for any existing web databases and is capable to prevent many kinds of attacks, thus significantly decreases the web databases' attack surface. We propose a system which report when any unauthorized user modifies our web database via SMS.

Future work: Web database security and Semantic web is constantly research topic. Our future works is to provide semantic web capability to analyze user access and authentication.

References

[1] Zhu Yangqing, Yu Hui, Li Hua, Zeng Lianming, Design of a new web database security model, IEEE, 2009, 292-297
[2] Leon Pan, A Unified Network Security and Fine-Grained Database Access Control Model, IEEE 2009, pg 265-270
[3] Xueyong Zhu, William Atwood, A web database Security model using the Host identity protocol, IEEE 2007
[4] Lianzhong Liu, Qiang Huang, A framework for database auditing, IEEE, 2009, 982-988
[5] Afonso Neto, Marco Vieira, Henrique Maderia,An appriasal to assess the security of database configurations, IEEE, 2009, 73-80
[6] Qing Zhao, Shihong Qin, Study on security of web based database, IEEE, 2008, 902-910
[7] WU Pufeng, Zhang Yoqing, An overview of Database security, Computer Engineering, Vol 32,2006,85-88
[8] Zhou Wen, A new web accessing database module basing in security of information computer security, 2008, 63-66
[9] S. Sudershan, Govind Kabra, Ravishankar Ramamurthy, Redundancy and Information Leakage in Fine-Grained Access Control, ACM SIGMOD 2006
[10] Jie SHI, Hong ZHU, A fine-grained access control model for relational databases, IEEE 2010, Pg 575-585
[11] Sohial Imran, Irfan Hyder, Security Issues in Databases, IEEE 2009, Pg 541-545
[12] Wang Baohua, Ma Xinqiang, Li Danning, A formal multilevel database security model, IEEE 2008, Pg 252-265
[13] Marty Humphrey, Sang-Min Park, Jun Feng, Norm Beekwilder, Fine-Grained Access Control for GridFTP using SecPAL, IEEE 2007, Pg 1-9
[14] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Pin-Han Ho & Xuemin (Sherman) Shen, "A Novel Anonymous Mutual Authentication Protocol With Provable Link-Layer Location Privacy", IEEE, 2009.
[15] Jie Wang & Jun Zhang, "Addressing Accuracy Issues in Privacy Preserving Data Mining through Matrix Factorization", IEEE, 2007.
[16] Anup Patel, Naveeta Sharma, Magdalini, "Negative Database for Data Security", IEEE 2009
[17] Attribute- Based Encryption for Fine-Grained Access Control of Encrypted Data", IEEE 2008
[18] Qing Zhao, Shihong Qin, " Study on Security- based Database", IEEE 2008
[19] Alex Roichman & Ehud Gudes, "Fine-grained Access control to web databases", ACM SACMAT, 2007.
[20] Elisa Bertino & Ravi Sandhu, " Database Security- Concepts, Approaches, and challenges", IEEE Transactions on Dependable and secure computing, 2005.