Volume 2, No. 3, May-June 2011
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info
ISSN No. 0976-5697

Network security with Fine Grained Databas Access Control model for Web Database

Amit Kuraria*
Shri Ram institute of technology
Jabalpur,India
kuraria.amit@gmail.com

Prof.Vikram Jain
Shri Ram Institute of Technology
Jabalpur,India
Vikram.srit@gmail.com

Prof. Sweta Modi
Shri Ram Institute of Technology
Jabalpur,India
Sweta_modi@gmail.com

Abstract: Web database is combination of database and web technology. Web database is placed on the Internet, there are many security problems. Widespread availability of internet access coupled with the increase in number of web applications has led to a surge in the amount of personal data stored online. The secrecy and the integrity are two important demands of security system. A wide variety of attacks that can be grouped into the category of "injection attacks" can be used to maliciously attack the web database server and database of a web application. These attacks consist of injecting specially formatted data into the application with the hope of corrupting the internal functioning of the database server. When network security and the database access control are addressed separately, the security systems are not optimized sufficiently as a whole. We propose a method of integrating network security with criterion based access control to handle network security. Fine grained access control must be supported by web databases to satisfy the requirements of privacy preserving and Internet-based applications. This 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.

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

I. 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 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 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, and provide efficient fine-grained access control.

The rest of the paper is divided as follows. Section 2 present the security related work. Section 3 covers the proposed algorithm. Section 4 present implementation and result. Section 5 concludes the paper and also present future work.

A. 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 technology and web technology. Data security 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 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 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.

II. 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).

Views are used to specify and store access permission for users. When a user submits a query, DBMS first finds all views 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) 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. Less work has been done with the FGAC model. The work of Agrawal *et al.* (2005) 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 Datalog. The security analysis was also provided. Moreover, they enforced policies by compiling policies in Transaction Datalog 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) 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 throughfunctions 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 rowlevel access control, column-level VPD 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.

III.OUR APPROACH

A. Proposed Algorithm

Algorithm

Input: user *U*, relation *R*, action *A*, database *D*.

Output: the combined Fine Grained Access Control policy *P*out.

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

12 *PR*=Union of (PSrole);
13 *P*out=*PD*∧*PR*.

IV. IMPLEMETATION

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 WebSecure, 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 got 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 WebSecure site and join it with the local parameter table and detect the inconsistencies. In order to optimize the activity performed by the WebSecure, 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 WebSecure 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 WebSecure maintains a copy of the frozen query results and periodically, it transfers the whole relation of the main database to the WebSecure 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 WebSecure 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 WebSecure 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 WebSecure, 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 WebSecure site.

V. CONCLUSION AND FUTURE WORK

The system was considered extremely useful by the administrators. In fact, several configuration problems have been disclosed and the administrators promptly identified improvements that could be made to their installations. The system also allowed us to put into evidence several security characteristics which were common to all installations. Such recurrence seems to be due to lack of security knowledge by the administrators and the use of operating systems that do not provide easy to use mechanisms necessary to comply with general security best practices. 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.

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.

VI. 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 Dependabland secure computing, 2005.