A WAY FORWARD TO IMPROVING FUNCTIONALITIES OF OPEN DATABASE SYSTEM

A.T. Akinwale , O.T. Arogundade And H. Obianugha

Department of Computer Science, University of Agriculture, P.M.B 2240, Abeokuta, Nigeria
E-mail: atakinwale@yahoo.com, arogundadeo@acm.org

ABSTRACT

The increasing need for information dissemination and the tremendous population growth in today's organizations calls for the need to migrate most application and their associated data to the network where several millions of people can access concurrently. Several challenges such as security risk and management/maintenance of the database have been posed to such system that operates on the network. In this research paper we have been able to determine some functionality that can be incorporated into a database management system that operates in an open network system environment. These functionalities include: application of open password for open database system, creation of register database, creation and monitoring of activity log file, creation of privacy lock on schema and application of a locking control on file. To demonstrate this, a system was designed using the student arm of the University of Agriculture Abeokuta as a case study. In designing the system, C# as the host programming language was used in designing the front-end while Microsoft SQL Server 2005 was used for the back-end design. With the introduction of these five functionalities, open database is secured to be used by the desired users.

Keywords: *Open Database, Relational Database, Log Files, Biometrics, Privacy Lock, Locking Algorithm*

1. INTRODUCTION

Security of computer system is a critical aspect of any information communication technology strategy; hence the security of open database system is an important component therein. Beyond ensuring the security of the host operating system, an open database needs to ensure that database management software itself is secured with all the constraints of security being put into consideration. Red hat database offers a strong set of features for enabling secure database systems. These features encompass the three key security issues: authentication, transmission and access control security. Authentication security is to ensure that only authorized users can connect to the database system. Transmission security is to ensure that data is not vulnerable to snooping while in transit over the network. Access control security is to ensure that users or group of users have access only to information to which they are entitled. Red Hat database provides other numerous security features that are built on such open standards as Kerberos and socket secure.

Databases have been common in governmental departments and commercial enterprises for many years. Today, databases in any organization are increasingly opened up to a multiplicity of suppliers, customers, partners, employees, etc. This idea has been unheard a few years ago. Numerous applications and their associated data are now accessed by variety of users requiring different level of access via manifold devices and channels. For examples, on-line banks allow customers to perform a variety of banking operation via internet and over the telephone while maintaining the privacy of account data. E-commerce merchants and their service providers must store customer order and payment data on their merchant server and keep them secure. E-department allows employees to update their personal information whilst protecting certain management information from unauthorized access. The medical professional must protect the confidentiality of patient data whilst allowing access to essential data for treatment. On-line

Vol 7. No 1. (pp 001 - 007)

brokerages need to be able to provide large number of simultaneous users with up-to-date and accurate financial information. This complex landscape leads to many new demands for open database security.

Database usually contains a company's most valuable information assets and if compromised could wreak havoc [8]. Database security is a serious issue and if not implemented correctly, the consequence can be costly to organizations if their vital data is hacked into or their customers' data leaks out which can even lead to cases of identity theft [5]. Data accessibility is a major goal and concern is data security. Many organizations cannot work properly if databases are down. To make the data or information available implies to provide the security mechanisms to ensure authentication, authorization and auditing procedures [3]. This means that security data involves:

(i) preventing unauthorized access to classified data by anyone without a business need to know.

(ii) preventing unauthorized users from committing mischief through malicious deletion or tampering of data

(iii) monitoring user access of data through auditing techniques [13]

Databases are increasingly subject to attack by internal and external attackers who no longer simply seek notoriety but now want financial rewards. By compromising the security of databases and obtaining customers' personal data, committing fraud or blackmailing the target company, both internal and external attackers can jeopardise the reputation, financial standing and customer trust of a business. As a result, database security has become a serious concern for an increasing number of corporations. Security vendors are responding by designing technologies that complements and extend tradition database solutions.

Open databases are becoming increasingly important in a wide range of operations. As computers have become faster and more powerful and their use more widespread, open database would become critical. For example, open databases can be used in stock trading, telephone switching system, world wide web and network management. Open databases depend not only on the security constraint but also on the number of access of information by the users. The goal of open database system is to meet free access time of transactions by the users. Rather than being open, more important properties of open database should be ability to retrieve expected documents at the right time.

An Open System is a system that does not comprise central administration components like the World Wide Web (WWW) or the intranet/internet in general. It is a system that regularly exchanges feedback with its external environment (that is, users communicating with the system). For a system to be termed open, the boundaries through which feedback can readily be exchanged and understood must be porous. In this kind of system, authenticated users must be able to access the system at any time and be able to retrieve information from the database without restriction. According to [3], to put the data available implies to provide the security mechanisms to ensure authentication, authorization and auditing procedures.

Since open system allows free access to databases (that is, most files in the database can be accessed or downloaded by authenticated users) and yet the integrity of the data in the database must be preserved, the functionality of the Relational Database Management System must incorporate the capability of managing the database to prevent it from unauthorized modification (that is, ensuring database/system security). The security measure that can be incorporated into such databases and its functionality is the centre of this research. Techniques such as: authentications, authorization, auditing procedures and so on have been in use to ensure proper management of a database management system. In this work, we have determined some functionalities that can be incorporated into an open database management system.

2. METHODOLOGY

For any database management system to function properly in an open system, without any disruption, the following functionalities are proposed for implementation.

- Open password for open database management system
- Creation of a register database
- Creation and monitoring of activity log file

- Creation of privacy lock on schemas
- Application of a locking concurrency control on file

2.1 Application of Open Password for Open Database System

Since anybody can have access to an open database management system, another means of ensuring that the system is not misused is to apply an open password system where all alphanumeric keys of the keyboard can be seen on the screen by everybody as they come in to register in order to be granted access to the system. This is to ensure that users do not have access to the system in a cheap way. It would therefore be assumed that anybody that can go through the rigour of registering possesses a level of trust to use the system.

2.1.0 Algorithm for Open Password

Definitions

Let the user input code be denoted by $I_n = c_1c_2c_3\ldots c_n$, and its length be defined as n = L ($c_1c_2\ c_3\ldots c_n$. Let **X** be the list of all the password characters i.e. A,…,Z, a,…,z, 0-9, symbols} and **Y** the set of the corresponding randomly generated numbers such that each value of **X** is assigned a random value from **Y**.

Constraint

The set of randomly generated numbers $S = \{r : 0 \le r \le 9, r \in \mathbf{Z}\}$ is such that $n(X)$ is evenly divisible by $n(S)$, i.e. $\frac{n(X)}{n(S)}$ = **d** must be an integer.

Also, the frequency of each y_i = **d** $\forall i$

Algorithm for generating all characters corresponding to each of the user input code

Computation of $G(c_i)$, the set of characters corresponding to each c_i of user input where $c_i \in I_n$ and i =1, 2,…,n.

Input: User input code, $c_1, c_2, c_3, \ldots, c_n$

Method: Let **n** = L ($c_1c_2c_3\ldots c_n$)

Compute

$$G(c_i) = \{List(y_i)\}, i = 1,2,3,\ldots,n$$

where $List(y_i) = \begin{cases} x_i, & \text{if } c_i = y_i, \\ \text{null}, & \text{otherwise.} \end{cases}$

and $List = \bigcup x_i \rightarrow y_i$

Output: The set of password characters corresponding to each c_i.

2.2 Creation Of Register Database System

In this era of internet which allows anybody with a web browser to access the net, open database can function properly without any form of detriment to the information in the database if a register database is created. The register database will contain the information of any user that is interested in using the open database management system. The information may include national identity card number or passport number, surname, first name, date of birth, sex, colour and fingerprint. As a result of this, for the first time the user enters the site, he or she is requested to fill in the necessary information in the registration form. After this, the user is then given a time frame of 24 hours for activation of his or her user account. After the 24 hours, if the user comes in to use the system, he must be able to provide the same initial information; hence he or she is requested to fill the same form as at first. The system then compares the information in the register database with the ones just provided. If it matches, then the user can be granted access to use the system, else access to the system will be denied.

2.3 Creation and monitoring of activity log file

Activity log file logs all operations that have been performed by the users. Monitoring log file helps to ensure that the system is not misused. For example, changes that are detrimental to the system are not made, and actions that could lead to database breakdown are not performed. If a user is found misusing the system, such a user is barred from using the system. This can be done by granting a Denial of Service (DOS) to such a user.

As soon as a user starts using the system, he is requested to provide his biometrics such as fingerprint, face or hand which is then recorded in the activity log file. Other information includes time-in, time-out, operation performed and remark. As long as

he keeps using the system, all operations performed are logged against his biometrics in the log file. This is to ensure proper tracing and monitoring of any form of attack detection on the database.

As soon as an attack is detected through the log file, the user with that biometrics is denied of further services. As a result of this, before existing users can further access the system, they are required to provide the same biometrics as before which is then compared to the biometrics of the barred user. If it matches, the user will not be allowed to use the system.

Consequently, such user can regain access to the system by going through the first stage of registration and must ensure that all information provided is entirely different from those provided before the denial of service else he cannot be granted access because as new users register, their personal information is compared with the information of the eliminated user to ensure that the same user does not gain access to the system any longer.

2.4 Creation Of Privacy Lock On Schemas

Privacy lock is associated with the database schema in such way that a user who wishes to access any file in the database, is allowed if the privacy lock condition is met. In creating the database schema, privacy lock constraint is imposed on each of the schema that forms the database. The constraint prevents the use to alter, display or copy part of the schema into subschema or creates a new schema into database. It specifies exclusive or protected retrieval and exclusive or protected update access. For example, if a user wishes to perform any operation such as download, print or read any file from open database, as soon as the database file is selected, the registration form and the open password system is displayed. At this point, the user can only be allowed to perform any operation if the information on the form is correctly provided. That is the information provided matches the user's information in the register database.

2.5 Application of a locking concurrent control on file

There are many algorithms on concurrent control, however, a locking system is chosen in this work for demonstration the importance of open database security. This method is used to allow free access to a file when it is necessary. This method maintains a system wide lock table for recording concurrently access the tables in open database system. The work implemented only read lock whenever a transaction accesses tables in its read phase and read blocks can be shared. If a lock requested is denied, the requesting transaction is blocked until the lock is released.

3. IMPLEMENTATION

To demonstrate these functionalities, a system was designed using the student arm of the University of Agriculture Abeokuta, Nigeria as a case study. The front end was designed in ASP.NET with C# as the host programming language while MSSQL 2005 was used for the back end design. The programming language was used for the implementation of the algorithm in section 2.1.0. From the algorithm, a form interface was generated as shown in figure 1. The form shows the alphanumeric in black with their corresponding numeric code in red.

Each time the system is launched, a different random number is generated for each of the alphanumeric keys. At the point of registration, before the user completes the registration form, a screen is displayed. The user then fill in the required information using corresponding random number generated for each alphanumeric keys of the keyboard. At this point the system already knows all the information it needs to know concerning that user. After the 24 hours from the date of the initial registration, the user must be able to provide information that matches all information provided at first, as a new set of random numbers would have been generated. The user then makes use of the new set of random number generated to complete the form. Failure to do this will lead to a denial of access. This means that the user is not ready to use the system. Anybody that is interested in using the open database management system should be able to go through the processes of registration. If otherwise, such user would not be able to use the system.

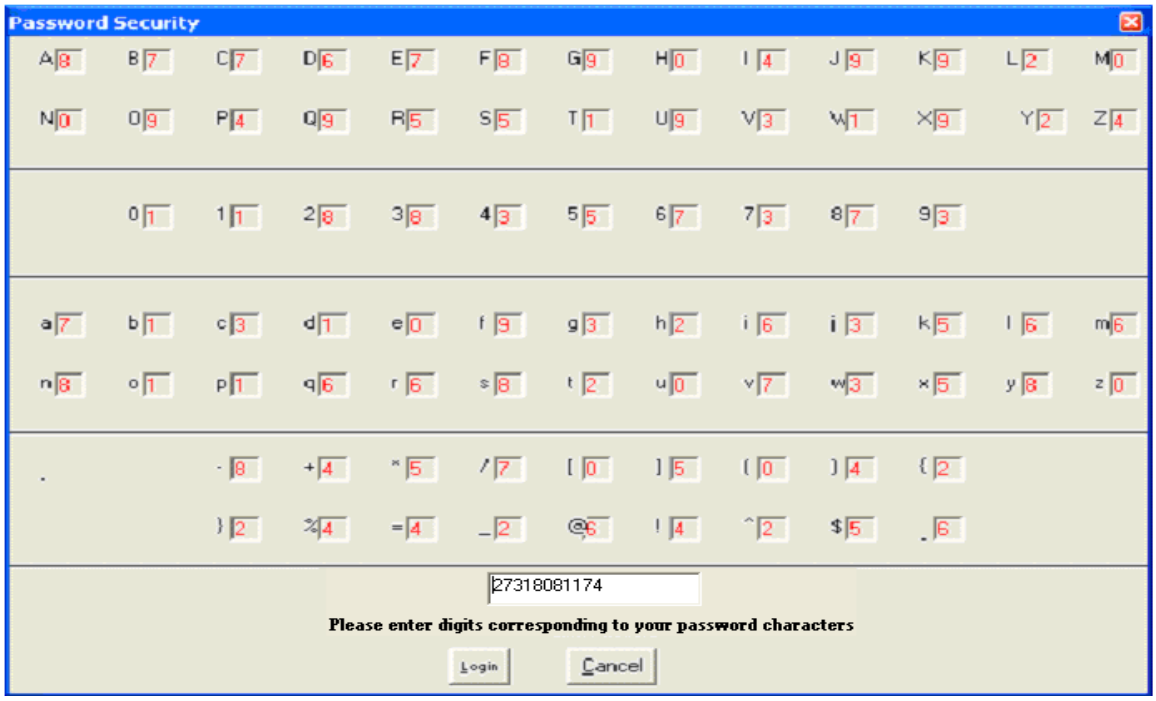


Fig. 1: Form interface

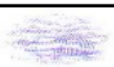
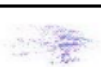
Upon Accessing of the university of Agriculture database, a registration form is displayed as shown in figure 2, which the user must fill using the interface in figure 1. After twenty four hours the user will activate the account by supplying the same information on the registration form in figure 3. If both information matches, a user biometric is added into the log file as shown in table 1. The second registration form which must be completed after the twenty four hours time frame is shown in figure 3. Since the information users provided at first as depicted in figure 2, matches the information provided after twenty four hours as depicted in figure 3, access was granted to them and they can from henceforth use the system. As long as they use the system, all activities performed are logged and monitored to ensure proper management of the system.

Fig. 2: registration form which is been completed using the open password algorithm.

The simulation of locking system is based on University database files that are resident in single hard disk operating on shared memory multiprocessors. These files are modelled as being uniformly distributed across all the hard disk. The execution of a transaction consists of multiple instance of alternating file access request and read file steps until the entire the read file in it complete or it is aborted for some reasons. When a transaction makes a file access request, for instance, lock request on a file in database, the request must go through locking algorithm to obtain a lock on the file. If the request is generated, open access form as indicated in figure 2 appears which a user must use open interface in figure 1 to fill before the

Fig. 3: Second registration form which is been completed using the open password algorithm.

Table 1. Log file to record user' activities

Log File					
S/N	Biometric	Time In	Time Out	Operation	Remark
1		08/04/2009 12:00:00	08/04/2009 02:00:00	download	okay
2		08/04/2009 09:00:00	08/04/2009 01:00:00	reading	okay
3		08/04/2009 02:00:00	08/04/2009 08:00:00	alteration	denied
4		08/04/2009 16:00:00	08/04/2009 02:00:00	print	okay

transaction proceeds to perform read operation which consists of a disk access and CPU computation. The transaction passes through hard disk and CPU queues. If the request for the lock is denied, the transaction will be placed into a file queue. The waiting transaction will be awakened when the requested lock is released. Transactions arrive in a poisson stream, for instance, their inter-arrival time are exponentially distributed.

4. CONCLUSION

In this paper an open security control framework for open database system is proposed. The proposed framework considers existing security constraints for protecting the integrity of open database information. Each constraint was demonstrated to authenticate users' particulars whenever to access open database information. Effort is still needed to select the best constraint for properly regulation information access. More explicit specification of the deadlock prevention mechanism is required in the system together with other database operations. This system detects conflicts in reading the files as soon as they occur and resolves them using blocking. It means that it conserves resources by blocking and many filled forms can be tied up in resource queue which may invariable generate thrashing.

REFERENCES:

[1]. Allen, G. Taylor ' SQL for Dummies' , 5th Edition, Willey Publishing Inc.2003.

[2]. Fernando, S. Lozano 'Introduction to Relational Database Design'', http://www.os2ss.com/users/lozano 1998.

[3]. Joaquin, A. Trinanses "Database security in High Risk Environment", International Business Machine Corporation 2001.

[4]. Kroenke, David M. "Database Processing, Fundamentals, Design and Implementation", Prentice – Hall Inc.1997 Pg 130-144.

[5]. Ashaya Bhatia ' Database Security, Information Technology Toolbox' Inc 2008.

[6]. Auto Chuvakin ' Network Database a System Log Management: The what, why and how', Computer Technology Review 2008.

[7]. Dan Rahmal 'Database Security, System and Methodology for Identifying and Protecting Weak Spots in Your Enabled Database', Miller Freeman Inc., 1997.

[8]. Duane Winner 'Making your Network Safe for Database', International Business Machine Corporation 2001..

[9]. Liker Kose Distributed Database Security, Data and Network Security, Spring GYTE Computer Engineering 2002.

[10]. Kroenke David and David J. Auer Database Concept, 3rd edition, Prentice Hall Inc., New York 2007.

[11]. Lighstone S. et al. Physical Database Design: The Database Professional Guide to Exploiting Indexes Views, Morgan Kaufman Press.2007

[12]. Red Hat Red Hat Database: Open Source Database and Security, Red Hat Inc., USA 2001.

[13]. Zikopoulous Paul The Database Security Blanket, International Business Machine Cor poration. 2001.

[14]. O' Neil P. and O' Neil E. Database: Principle, Programming and Performance, Morgan Kaufman Publishers, San Francisco 2001

[15]. Connoly T.M. and Begg T.E. Database Systems, A Practical Approach to Design, Implementation and Management, 3rd Edition, Addison Wesley 2002.