IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4992

Database Security: An Overview

Ali Salehnia
Computer Science Department, South Dakota State University, Brookings, SD 57007, ali_salehnia@sdstate.edu

ABSTRCT

Database security is an important issue in database design. Availability, confidentiality, and integrity are properties used to evaluate databases' security level. Methods such as authentication, auditing, and access control could be utilized. These methods are distinguished by the time when they are used. Background knowledge such as relational databases' structure and SQL operations is of great help in understanding database security.

1. INTRODUCTION

Databases and database technology have a major impact on the growing use of computers. Databases play a critical role in almost all areas where computers are used, including business, engineering, medicine, law, education, and library science. Because of the importance of data and information in databases of any field, they have to be secure and protected. Data should be protected from corruption or unauthorized access, and information should be controlled when users retrieve it. Maintaining database security is the responsibility of the database management system rather than the operating system or application programs. Security is a major issue in any database management system, particularly those which use sensitive information. Security is achieved by granting access rights to authorized users.

Database security can be divided into the three following separate but interrelated objectives [2]:

- *Secrecy:* Secrecy is concerned with improper disclosure of information. The terms "confidentiality" or "non-disclosure" are synonyms for secrecy.
- *Integrity:* Integrity is concerned with improper modification of information or processes.
- *Availability:* Availability is concerned with improper denial of access to information. The term "denial of service" is also used as a synonym for availability.

These three objectives arise in practically every information system. For example, in a payroll system secrecy is concerned with preventing an employee from finding out the boss's salary; integrity is concerned with preventing an employee from changing his or her salary, and availability is concerned with ensuring that the paychecks are printed on time. Similarly, in a military command and control system secrecy is concerned with preventing the enemy from determining the target coordinates of a missile, integrity is concerned with preventing the enemy from altering the target coordinates, and availability is concerned with ensuring that the missile does get launched when the order is given.

They also differ with respect to the extent of the objectives themselves and the technology to achieve them is understood. It is easiest to understand the objective of secrecy. Integrity is a less tangible objective, one on which experts in the field have diverse opinions. Availability is technically the least understood aspect. In terms of technology, the dominance of the commercial sector in the marketplace has led vendors to emphasize mechanisms for integrity rather than ones for secrecy needs.

2. PASSWORD MANAGEMNET

In computer security, "Accountability" is a primary component. To satisfy the requirements of accountability, it is necessary to have individual user's identification and authentication. The functionality of individual user's identification provides a means to distinguish different users definitely; at the same time, authentication ensures that the specific user is indeed who he or she claims to be. To provide these functionalities, several authentication mechanisms are invented, based on the method authentication [9][2][4]. Authentication mechanism utilizes password-based mechanism, token-based authentication, and biometric authentication. Easy implementation makes the password-based authentication mechanism a good technology. However, even a strong password-based authentication mechanism has a few shortcomings [32] [20] [28].

How to prevent passwords from being compromised is an important issue. Three methods deal with implementation of this issue [32]: identifying weak password by running a crack program before their breakage; making penetration difficult by enlarging the overhead of cracking computation; and improving users' security consciousness. The first method is closely related with the second method. The aim of eliminating the passwords weakness is to make penetration more difficult. R. Morris and K. Thompson wrote the first paper in the computer security area in earliy 1970's[22]. This paper serves as the fundation of many following papers concerning this topic. As R. Morris and K. Thompson indicated, originally passwords stored in UNIX systems were raw passwords. This means anyone who obtains these passwords whether accidentally or intentionally can read them without any barrier. Obviously, this is unacceptable in the high probability of occurrence of software or hardware failures. In these systems, once the "bad guys" obtain the password file they can access the system freely. In order to prevent thes eproblems, R. Morris and K. Thompson suggested the encrypted password be stored instead of the raw password. With encrypted passwords even when the password file is disclosed, the unauthorized people still need to spend time & energy to decrypt those encrypted passwords[30].

However, there are still methods to decrypt those encrypted passwords. Among them, the most frequently used is the exhaustive searching method. The exhaustive searching method is not fit to find a specific password. What it does is to try each trial raw password, let those raw passwords be encrypted by the encrypt algorithm, and see if the resulting password is the same as the stored encrypted password. Feldmeier and Karn [15] investigated and sugessted elements necessary to make the exhastive searching method useful as follows: High performance/price ratio computers; Large on-line word lists; A known password-encrypted algorithm; A constraint on the acceeptable running time for the login program; A publicly readable password file; and passwords with a significant probability of being in the list.

In reality, most database system managements protect their end users' passwords through implementing these steps by maintaining a password database and the access to this database will be controlled [21]. To control access, most database management systems use their internal password-produced algorithm to produce the end user password randomly. Apparently, passwords produced by this method are difficult for people to remember. In order to alleviate this conflict, it is usually allowed to change their password slightly by end users so that recitation will be easy. But there are still basic requirements that these passwords must confirm to. For example, the password's length should be no shorter than 8 characters. In addition, most systems also require end users to change their password periodically. This is accomplished by attaching a time stamp to each new produced password. Periodically passwords will be checked. Expired passwords will be disabled[30].

Hitchings [24] and Davis and Price [16] argue that this narrow perspective has produced security mechanisms that are, in practice, less

effective than they are generally assumed to be. Now let us have a brief look at what happens in real database security management.

2.1 Low Security Consciousness

Although many security suggestions or rules are set up, they are rarely used due to end users' low security consciousness. For example, in database management, there are limitations on password length, password composition, password lifetime, and attempting logging rates, etc. But investigation shows that end users rarely use these rules unless the internal security mechanism enforces them to do so.

2.2 Lack of Security Knowledge

There is a doctrine in database security management – the "need to know" principle [27]. Under the direction of this principle, most people believe the less end users know of the database, the more secure the database will be. So end users are rarely told about which secure method is applied in the database, its weakness, etc. Without this necessary security knowledge, end users don't know what they should protect, and how.

2.3 Bad Communciation Between Database Experts and End Users

Agood database security management needs the effort from end users. But information indicates that, in the real world, there is a bad commnunication between computer experts and end users. Computer experts do their job to add more security policies to increase security. On the other hand, end users adopt their own methods to increase the security of their personal information as well. They don't understand the concepts and methods introduced by computer experts well[27][11]. So how can we anticipate the database system' security will be increased greatly?

3. AUDITING IN RELATIONAL DATABASE MANAGEMENT

According to the subjects who perform the audit (for example internal/external auditor), the object being audited (for example annual balance sheet) and the rules & principles that are checked for compliance with the auditing objects (for example rules for the rendering of accounts, law or data protection) [7], there are many kinds of auditing. For example there are two kinds of auditing – external auditing and internal auditing depending on the subject performing the audit. External auditing, conducted by objective outside persons, examines not only financial statements but also accounting records and other relevant information [14]. However, internal auditing is carried out within an enterprise, usually by an internal audit group that reports to a high level in the enterprise or to an audit committee of the board of directors [5]. Thus, after we have these concepts, it will be easy to deduce what is auditing in database.

Database auditing is the monitoring and recording of activities occurring within a database [33]. It provides a functionality to collect a set of records to show that the system is intact and works properly. There are two distinguish auditing methods. One method is to divide auditing into internal auditing and external auditing. Internal auditing is a system-level internal auditing mechanism used to audit a database system almost continuously. On the other hand, external auditing is an auditing invoked manually by functions exported to the outside world. The second method is to divide auditing into operational auditing and system auditing. Operational auditing is a review of computer operations that covers system security policy, data integrity controls, system development procedures, and backup recovery procedures [26]. Compared to operational auditing, which audits many computer operations; system-auditing try to prove the system is working well by tracing the details of a specific transaction.

To perform auditing, audit-ability is a basic precondition for auditing, permitting the objectives of an audit to be carried out speedily and effectively [7]. Thus, in computer database systems, a proper auditing mechanism should have the following five goals [5]: Allow for reviewing patterns of access; Allow for discovery of attempts to bypass system controls; Allow for discovery of use of privilege; Act as a deterrent; and Provide additional assurance.

With these functionalities, the auditor is able to observe, check, and test the database system to see whether it functions under control or not. However, the above functionalities are not enough to perform an efficient auditing. To provide an efficient auditing, an auditing plan is necessary. Usually developing an auditing plan involves steps such as mastering the system internal mechanism, writing a rough draft concerning what will be audited, in which order, etc, designing action-oriented plan, and conducting walk through.

In practice, when an auditing mechanism is implemented in database systems, a lot of practical problems need to be settled too, such as which events should be audited, and in which form is the auditing result stored [5].

3.1 Which Events Should Be Audited?

Since there are many things that can be audited, to audit everything is impractical and impossible. In database auditing mechanisms, typically auditable things are login events, access events, querying events, modifying information stored in database, statements, privileges, roles, etc. Generally they can be divided into three categories: statement level auditing, system level auditing, and object–level auditing [33]. Thus an auditor must decide what will be audited when developing an auditing plan.

In trusting OS, only access to file and segment, which are the object types in OS, needs to be recorded. However, in DBMS, there exist numerous object types, compared to those existing in trusting OS. Objects can fall into two categories – named object and stored object. Stored objects are things such as a table, a row, or a tuple. Named objects are things such as a relation, a review, or a metadata etc. Obviously to record all accesses to database objects will result in a large, unreadable audit log, which actually has no practical use. In DBMS, accessing data stored in database can be accomplished by accessing them directly or accessing them through views which will greatly enlarge the volume of recordable events.

3.2 In Which Form Is the Auditing Result Stored?

Normally in database systems, an auditing trail is used to store the auditing results. An auditing trail can be defined as "a set of records that collectively provides documentary evidence of processing used to assist in tracing original transactions forward to related records and reports, and/or original backwards from records and reports to their component source transaction" [DoD 83]. Because an auditing trail records what happens in the database system and is used to reconstruct those actions completed in database systems, preventing the audit trail from fraud becomes very important. To achieve this aim, usually the audit trail is stored in a highly protected file or a small internal database system.

Now, after discussing many things about auditing from the theoretical view to the implementation view, let us make a summary about issues that a designer should be aware of, concerning auditing [26].

- Audit Policy: what events should be audited? How should an appropriate audit policy, for the perceived risks, be determined and adapted when the threat changes?
- Auditable Event: which event should the DBMS be capable of recording? Which data about each event should be recorded?
- Audit Storage: will service be denied if the DBMS shuts down when the audit trail overflows or the audit mechanism malfunctions?
- Audit Credibility & Protection: is the audit mechanism credible? How is the DBMS audit mechanism guarded against disclosure, alteration, purging or disabling?
- Audit Analysis: should the audit trail be stored in the database itself to facilitate analysis?

4. ACCESS CONTROL

What is access control is a mechanism that controls the access of subjects to objects. Subjects refer to normal database users, programmers and administrators. And objects normally refer to data stored in database or some other things [2] [8] [1] [3].

As described before, access control maintains the access of subjects to objects. To maintain control, most research work is emphasized in the following three aspects: discretionary access control, mandatory access control, and role-based access control. Each has its advantages and disadvantages, comparing to others.

4.1 Discretionary Access Control (DAC)

DAC controls the access by applying a discretionary protection policy, which governs the user's access to the information on the basis of the user's identity and the rules that specify, for any user and any object in the system, the types of accesses (e.g., read, write, or execute) the user is allowed for the object [6] [10] [29] implementing DAC, the principle "least privileges" is maintained; i.e., only those privileges necessary will be allocated. No additional privileges will be provided. Using this method, the management of users' access authority will be difficult sometimes in cases such as a group of end users who need the same access authority. With DAC, we need to authorize each end user explicitly. Furthermore, what can we do if the group's authority needs to change or be canceled in a future time. Since it will be a good idea to change each member's privilege separately, we need some kind of mechanism to make the access management easy.

Another problem associated with DAC is the "Trojan Horse" attacks. The execution of a "Trojan Horse" program may provide access to the database by unprivileged persons [6] [11] [19].

Here is an example of a "Trojan Horse" attack works [17] [1]. Let's suppose there is a computer programmer, a manager and a table called "Secret". Also suppose that the manager has the access rights to the table access but not the programmer. In order to be able to access the contents of the "Secret" table the programmer could use the following method. First, he/she would set up a table called "Stolen" and gives the manager and himself/herself the right to access and write the table. Then the programmer would build up a normal business application with a hidden program. This program is used to copy the contents of the table "Secret" to the table "Stolen" secretly when the application is run under the privilege of the manager. Using this method, the programmer can access and obtain unauthorized data. The "Trojan Horse" problem illustrates that although each access is controlled and allowed only if authorized, it is possible to bypass the access restrictions and to read the data without the authorization of the data owner [22][31][18][17]. We must realize that this weakness is invoked by the DAC itself. Once DAC is implemented in security management, the database has the possibility that it will suffer attacks.

4.2 Mandatory Access Control

Mandatory Access Control (MAC) is an access policy supported for systems that process especially sensitive data (e.g., government classified information or sensitive corporate data). Systems providing mandatory access controls must assign sensitivity labels to all subjects (e.g., users, and programs) and all objects (e.g., files, directories, devices, windows, and sockets) in the system. A file's sensitivity label specifies the level of trust that a user must have to be able to access that file. Mandatory access controls use sensitivity labels to determine who can access the user's system [4].

The mandatory access controls implement a multi-level security policy–a policy for handling multiple information classifications at a number of different security levels within a single computer system. They are based on security labels associated with each data item and each user. A label on a data item is called a security classification, and a label on a user is called a security clearance. In a computer system, every program run by a user inherits the user's security clearance. It is important to understand that when a particular program, such as a text editor, is executed by a secret user it is run as a secret process, whereas when executed by an unclassified user, it is run as an unclassified process [23][31][12].

Every access control mechanism, implementing MAC, must obey two rules

- Rule 1 — Simple Security Property: subject S can read object O only if the security level of S is equal to or larger than the security level of O.
- Rule 2 — *-Property: subject S can write object O only if the security level of S is equal to or less than the security level of O.

Under the restriction of these rules, attacks such as "Trojan Horse" can be avoided. Let's revisit the example stated in DAC [17]. If we let the security level of the table "Stolen" be less than the security level of the manager, then even the "Trojan Horse" still exists in the application and tries to write to the table "Stolen" when the manager is running the application. But, according to Rule 2 - *-Property, such activity is not allowed and the "Trojan Horse" attack is averted.

5. SUMMARY

This paper has addressed some of the concerned dealing with database system security. Because nowadays database systems become more and more complex, it will be increasingly difficult to prevent illegal penetration. Much research work has been done in this field and many methods, such as access control, authentication, and auditing have been suggested to provide required security. Still there are many security holes existing in modern database systems. The closure of these holes needs to be addressed through user education and implementations of security methods.

6. REFERCNES

[1] Abram, M. and Olson, I. (1990). "Computer Access Policy Choices", Computer and Security. Pp. 699-714.

[2] Abrams, M, Jajodia, S, and Podell, H, eds, Information security: An integrated collection of Essays, IEEE Computer Society Press, 1995

[3] Adams, Anne and Sasse, Martina Angela, "Users are not the enemy: Why users compromise computer security mechanisms and how to take remedial measures," *Communications of the ACM*, 42(12):40-46, December 1999http://www.acm.org/pubs/contents/journals/cacm/1999-42/#12

[4] Atzeni,P. and Antnellis, D. (1993). Relational Database Theory. Benjamin/Cummings

[5] Audit, NCSC Technical Report – 005, Volume 4/5, 1996http://www.radium.ncsc.mil/tpep/library/rainbow/NCSC-TR-005-4.pdf

[6] Bertino, Elisa, Jajodia, Sushil and Samarati, Pierangela, Database Security: Research and practice, Journal of Information System, May 1995

[7] Bhaskar, K, Computer Security: Threats and Countermeasures, published by NCC Blackwell, 1993

[8] Castano, S, Fugini, M.G, Martella, G,and Samari, P, Database Security, Addison-Wesley, 1994

[9] Controlling Database Access, Oracle8i Concepts, Chapter 29http://technet.oracle.com/doc/server.815/a67781/c25acces.htm

[10] Csilla, F. (2000). Discrete Access Control. O'ReillyAssociates, Inc

[11] Date, C.J. (1986). An Introduction to Database Systems. Fourth Edition. Addison-Wesley.

[12]. Desai, B. (1997). An Introduction to Database Systems. Galgotia Publications

[13] Fortier, Paul. (1997). Database Systems. McGraw-Hill.

[14] Feldmeier, D. C and Karn, P. R, UNIX password security—ten years later (invited) 1989. Lecture Notes in Computer Science Volume 435 http://www.ja.net/CERT/JANET-CERT/./Feldmeier_and_Karn/crypto_89.ps

[15] Handbook of Information Security Management, section 5-3 system securityhttp://www.cccure.org/Documents/HISM/464-469.html

[16] Hitchings, J, Deficinecies of the traditional approach to information security and the requirements for a new methodology, Computer and Security, 14, 1995, 377-383

[17]. Harrison, M., Ruzzo, W., and Ullman, J. (1976). "Protection in Operating Systems," Communication of ACM. Pp. 461-471.

[18] HongHai, S. "Bell-Lapadul" http://www.cs.unc.edu/~dewan/242/f97/notes/prot/node13.html

[19] http://luna.pepperdine.edu/~ckettemb/class/Codd12R.html

[20] http://paris.cs.berkeley.edu/~perrig/projects/usenix2000/node2.html

[21] Lunt, T. "Database Security" http://www.ecs.csun.edu/~btimmer

[22] Morris, R and Thompson, K, Password security: A case history, *Communications of the ACM*, 22(11), Nov 1979http://www.acm.org/pubs/contents/journals/cacm/1979-22/#11

[23] Osborn, Sylvia, Mandatory Access Control and Role-Based Access Control Revisited,http://www.acm.org/pubs/citations/proceedings/commsec/266741/p31-osborn/

[24] Parker, D.B, IT Security: The Need for International Cooperation, Restating the foundation of the information security. In G.C.Gable and W.J.Caelli,Eds., Elsevier Science Publishing, Holland, 192

[25] Parker, D. "safeguards Selection Principles", Proc. Of 2nd IFIP International Conference on Computer Security. Pp 83-96.

[26] Polyinstantiation, NCSC Technical Report – 005, Volume 3/5, 1996http://www.radium.ncsc.mil/tpep/library/rainbow/NCSC-TR-005-3.pdf

[27] Procedures and Packages, Oracle 8i Concepts, Chapter 18http://technet.oracle.com/doc/server.815/a67781/c17pckgs.htm#5253

[28] Rennhackkamp, M. (1997). Database Security Issues. . McGraw-Hill.

[29] Rob, p. and Coronel, C. (1999). Database Systems. 3rd Edition.

[30] Sandhu, Ravi and Samarati, Pierangela, Authentication, Access Control, and Audit, http://www.acm.org/pubs/citations/journals/surveys/1996-28-1/p241-sandhu/

[31] Silvano, C. et al. (1995). Database Security. Second Edition. Addison-Wesley.

[32] Shortcomings of Password-based Authentication http://paris.cs.berkeley.edu/~perrig/projects/usenix2000/node2.html

[33] Theiault, Marlene & Heney, William, Oracle Security, published by O'Reilly, 1998