Security in Data Warehouses

By
Edgar R. Weippl, Secure Business Austria, Vienna, Austria
Favoritenstrasse 16
1040 Wien
Tel: +43-1-503 12 80
Fax: +43-1-505 88 88
E-mail: eweippl@securityresearch.at

Keywords: Security; data warehouse; data mining; statistical database security; privacy

Security in Data Warehouses

ABSTRACT

The last several years have been characterized by global companies building up massive databases containing computer users' search queries and sites visited; government agencies accruing sensitive data and extrapolating knowledge from uncertain data with little incentive to provide citizens ways of correcting false data; and individuals who can easily combine publicly available data to derive information that – in former times – was not so readily accessible. Security in data warehouses becomes more important as reliable and appropriate security mechanisms are required to achieve the desired level of privacy protection.

INTRODUCTION

Landwehr (2001) defines how the etymological roots of the term "secure" are found in "se" which means "without," or "apart from," and "cure," i.e. "to care for," or "to be concerned about".

While there are many definitions of the primary requirements of security, the classical requirements are summarized by the acronym CIA. CIA is the acronym for confidentiality, integrity, and availability. All other security requirements such as non-repudiation can be traced back to these three basic properties.

Avizienis (2004) defines *confidentiality* as the absence of unauthorized disclosure of information, *integrity* as the absence of improper system alterations and *availability* as readiness for correct service.

- *Dependability* is a broader concept that encompasses all primary aspects of security save confidentiality, and, in addition.
- *Reliability*, which refers to the continuity of correct service.
- *Safety*, defined as the absence of catastrophic consequences for user(s) and environment.
- *Maintainability*, which is the ability to undergo modifications and repairs.

BACKGROUND

While security obviously encompasses the requirements of the CIA triad this article will focus on the mechanism of access control (AC) as this addresses both confidentiality and—to some extent—integrity. Database security was addressed in the 1960s by introducing *mandatory access control* (MAC), driven mainly by military requirements. Today, *role-based access control* (RBAC) is the commonly used access control model in commercial databases.

There is a difference between trusting a person and trusting a program. For instance, Alice gives Bob a program that Alice trusts. Since Bob trusts Alice he trusts the program. However neither of them is aware that the program contains a Trojan. This security threat leads to the introduction of MAC. In MAC, the system itself imposes an access control policy and object owners cannot change that policy. MAC is often implemented in systems with mulitlevel security (MLS). In MLS information objects are classified in different levels and subjects are cleared for levels.

The *need-to-know* principle, also known from the military, stipulates that every subject receives only the information required to perform its task. To comply with this principle, it is not sufficient to use sensitivity labels to classify objects. Every object is associated with a set of compartments. Subjects are classified according to their security clearance for each given area/compartment.

Classification labels are of the form (*Ss*,*Sc*) where *Sr* is a sensitivity and *Sc* a set of compartments. (*Os*,*Oc*) dominates (*Ss*,*Sc*) if (*Ss*,*Sc*)<=(*Os*,*Oc*).

This <= relation is true when

- *Ss*<=*Os* where the <= relationship here is with respect to the classified < sensitive < secret < top secret sensitivity classification, and
- *Sc*<=*Oc* where the <= relationship is a subset relation of sets.

The Bell LaPadula (BLP) model (1975) forms the fundamental architectural idea behind guarantee of secrecy in MLS. The Biba model by the Mitre Corporation (1997) is used to protect integrity: BLP's no-read-up and no-write-down properties are inverted to the no-write-up and no-read-down rules. Today, Oracle's Label Security and DB2's Label Access Control are contemporary examples of this security model.

The most widely used access control model is the role-based access control (RBAC) model. This section will briefly summarize various properties of NIST's RBAC model as pointed out by Sandhu et al. (2000). The notion of *scalability* is multi-dimensional. RBAC does not define the degree of scalability implemented in a system with respect to the number of roles, number of permissions, size of role hierarchy, or limits on user-role assignments, etc.

As RBAC is based on permissions that confer the ability to do something on holders of the permission, it does not contain *negative authorizations* (prohibitions). The *nature of permissions* is not specified in the RBAC model itself. Permissions can be either fine-grained or coarse-grained and may also be customized. The exact nature of permissions is determined by the application.

Moreover, RBAC does not specify the ability of a user to select which roles are activated in a particular session. The only requirement is that it should be possible to allow a user to activate multiple roles simultaneously. It does not matter if the user is able to explicitly activate roles or if all roles are automatically activated by the system.

RBAC Constraints

Since permissions are organized into tasks by using roles, conflicts of interests are more evident than if dealing with permissions on a per-user basis. In fact, a conflict of interest among permissions on an individual basis is hard if not impossible to determine. Separation of duties among roles (i.e., defining mutually exclusive roles) provides the administrator with enhanced capabilities to specify and enforce enterprise policies. Since RBAC has static (user-role membership) and dynamic (role activation) aspects, the following two possibilities can be distinguished accordingly.

First, *Static Separation of Duties* (SSD) is based on user-role membership. It enforces constraints on the assignment of users to roles. This means that if a user is authorized as a member of one role, the user is prohibited from being a member of a second role. Constraints are inherited within a role hierarchy.

Second, *Dynamic Separation of Duties* (DSD) is based on role activation. It is employed when a user is authorized for more roles that must not be activated simultaneously. DSD is necessary to prohibit a user from circumventing a policy requirement by activating another role.

Administrating RBAC

Definition of roles and constraints, assigning permissions to roles, and granting membership to roles are the most common administrative tasks in RBAC. When a new employee enters the company, the administrator simply adds this person to one or more existing roles according to the users tasks and needs. Similarly, users can be removed from a role when they leave the company or added to new roles when their functions change.

It is commonly agreed that one of RBAC's biggest advantages is its easy administration. Nonetheless, managing a large number of roles can still be a difficult task. However, Sandhu and Coyne (1996) present an intriguing concept that shows how RBAC might be used to manage itself. An administrative role hierarchy is introduced, which is mapped to a subset of the role hierarchy it manages.

Coexistence with MAC / DAC

Mandatory access control is based on distinct levels of security to which subjects and objects are assigned. Discretionary access control (DAC) controls access to an object on the basis of an individual user's permissions and/or prohibitions. RBAC, however, is an independent component of these access controls and can coexist with MAC and DAC. RBAC can be used to enforce MAC and DAC policies as shown in (2000). The authors point out the possibilities and configurations necessary to use RBAC in the sense of MAC or DAC. For a detailed discussion on defining and organizing roles please refer to Nyanchama and Osborn (1994), who introduce a formal role graph to facilitate role administration. Ferraiolo and Kuhn (1992), for example, published fundamental concepts on granting and revoking membership to the set of specified named roles.

SCIENTIFIC CONCEPTS

Classic access control is still the mechanism of choice to protect not only databases but also data warehouses. The difference between a database and a data warehouse is that database is designed and optimized to process individual tuples and the data warehouse is optimized to respond to queries that analyze aggregated data. OLTP (On-Line Transaction Processing) systems are secured by controlling access to individual tuples but for data warehouses the issue of data protection is more complex. For typical access control there are several shortcomings. First and foremost, users can do anything with the data once they have access to it; Second, even if access to fine grained detail data is not permitted, querying different similar datasets can reveal fine details; this is also known as inference attacks. The first issue can be addressed—in theory—by usage control as described by Park and Sandhu (2004), the second by several methods of statistical database security. Both topics are very active fields of research.

Usage Control

The main problem with data collection is that people might allow companies to use data for specific reasons (such as recommending related products) but do not consent to other uses of the same data. Usage control byPark and Sandhu (2004) is a concept that makes it possible to enforce pre- and postconditions when using data. It is similar to a traditional reference monitor, only that the restrictions are enforced during the entire access, as proposed by Thuraisingham (2005): The privacy control would "limit and watch access to the DBMS (that access the data in the database)."

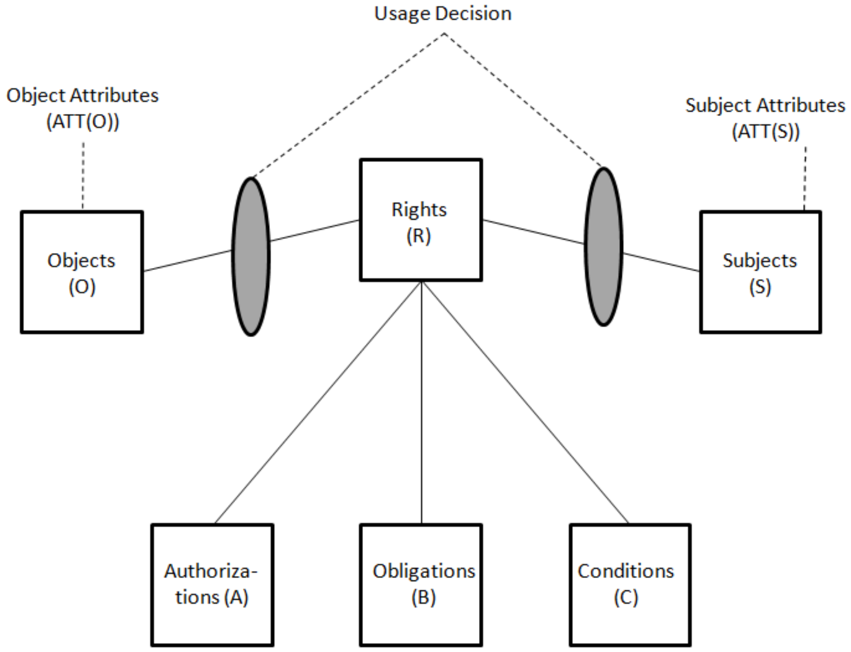

Figure 1: The UCONABC usage control model by Park and Sandhu (2004).

Statistical Database Security

A statistical database contains information about individuals, but allows only aggregate queries (such as asking for the average age and not an individual's age). Nonetheless, inference can be used to infer some secret information. Data warehouses are built to support data mining. If a data mining tool can be used to derive sensitive information from unclassified information legitimately obtained, there is an inference problem, as discussed by Bertino et al. (2006).

Well-established protection concepts for statistical database security, such as: restriction-based techniques, query set size control, expanded query set size control-audit based (assumed information base), perturbation-based techniques, data swapping (distribution unchanged), random-sample queries, fixed perturbation (modify data), and query-based perturbation. For an in-depth description Castano et al. (1994) and Willenborg and De Waal (1996) are excellent sources.

Query set size control Enforcing a minimum set size for returned information does not offer adequate protection. Denning (1982) described trackers that are sequences of queries all within the size limits allowed by the database; when combined with AND statements and negations, information on individuals can be inferred. While simple trackers require some background information, Denning (1979) as well as Denning and Schlorer (1983) show how general trackers can be used without in-depth background knowledge.

A very simple example illustrates how inference causes information leakage. If it is known that Alice is the oldest person but her age is unknown, repeatedly asking "How many people are older than *X* years? " with different values of *X* until the database returns the value 1, allows inference of Alice's age. By enforcing that each query returns aggregated data of more than one rows will not solve the problem. Repeatedly querying "How many employees are older than *X*? " until the system rejects the query because the query would return less than *N* rows, identifies a minimum set. This set includes *N*+1 employees, including Alice, *w* are older than *X*; let *X*=66 at this point. Subsequently, a query "Retrieve the sum of ages of all employees who are older than *X*? " will return a result *R*1. The last query "Retrieve the sum of ages of all employees who are not called Alice and are older than *X*? " will return *R*2. Finally, subtract *R*2 from *R*1 to obtain Alice's age. The example includes a query "not called Alice" that excludes a single item. If the "not equal" operation would not be allowed, a binary search could still be used to exclude a single item with a comparison operator. Simple control of result sizes as described here are not designed to prevent such an exclusion.

In *audit-based expanded query set size control* aka. Nabil and Worthmann's (1989) 'query set overlap control' the system decided whether to grant access to an "assumed information base," which is the history of all the requests issued by the user. The assumed information base contains all possible inferences that can be generated with the results of all previously issued queries; before answering a new query the system has to decide whether the query could be combined with the assumed information base to infer confidential information.

Perturbation-based techniques (cf. Table 1) are characterized by modifying the data so that the privacy of individuals can still be guaranteed even if more detailed data is returned than in restriction-based techniques. Data can be modified in the original data or in the results returned.

Data swapping	Data is exchanged between

	different records; individual information is thus protected while calculated statistics are not impacted.
Random sample queries	A set of answers to a specific query are created dynamically by selecting a random subset instead of all data item. This approach works well only for large datasets.
Fixed perturbation	Data is modified—not swapped—as soon as it is loaded into the data warehouse.
Query-based perturbation	Data is modified for each query dynamically. The advantage is that the accuracy can be varied individually depending on the user's trustworthiness.

Table 1: Different perturbation-based techniques

According to Samarati and Sweeney (1998) k-anonymity refers to a concept that guarantees that data of an individual will remain indistinguishable from that of at least *k*-1 others. The basic idea to protect privacy is centered on quasi-identifiers. Quasi-identifiers are usually a combination of data items that probably allow an identification of a person such as birth date, ZIP code, and gender. The idea, explained by Sweeney (2002), is that the data provider knows which data is externally available, for example a list of people with their names, birth date, ZIP code and gender. High dimensionality, however, may cause problems. Charu (2005) points out that once the number of dimensions increases to about 20, even 2-anonymity cannot be preserved in most cases without losing too much original information.

K-anonymity can be attacked using the homogeneity attack or the background knowledge attack. The homogeneity attack is very simple. If all *k* datasets have the same values in the field with sensitive data, privacy is not protected. For instance, if 5-anonymity is guaranteed and all patients suffer the same illness, there is no longer privacy. The background knowledge attack uses background knowledge to exclude impossible or unlikely datasets,as shown by Loukides and Shao (2007). Machanavajjhala et al. (2006) state that l-diversity is a measure that even the attacker needs *l*-1 relevant pieces of background knowledge to infer a (positive) disclosure.

APPLICATIONS

The previous sections focused on security concepts that are relevant for databases in general and—as data warehouses are databases optimized for a certain type of queries—for data warehouses. When implementing a data warehouse it is, however, essential that security is considered in an end-to-end way. The real goal is to protect the data and not only the data in the data warehouse. Before data is loaded into the DWH, it needs to be extracted from the source systems and is subsequently transformed, cleansed and prepared for loading. During this process the data has to be secured to the same standard as in the data warehouse. When clients query data, client security also becomes an issue. The data may be well protected in the DWH but a compromised client with full access to the DWH will certainly compromise all of the data. Security considerations need to consider all layers of the system involved. A DWH is not secure unless the underlying operating system is well secured and network security is adequately addressed.

DWH security and privacy is an active research area and is also relevant for industrial projects. Oracle, for instance, provides a detailed white paper on this topic. In industry, the challenge is mainly securing an entire and complex system, whereas academia strives to establish methods to preserve privacy of individuals and yet allow for the computation of meaningful statistics and detection of patterns.

FUTURE TRENDS AND OUTLOOK

Future research in data warehouse security will address several issues. First, with the increasing size of DWHs containing very personal information, privacy-preserving techniques will become more important. This area of research has also received more attention because nation-wide data gathering programs for national security are established. Second, while this theoretical research is certainly important, there are many more aspects to security that need to be considered. A nationwide DWH needs to be secured as an entire system including the mechanisms of data delivery, data querying, and usage. Security in DWH rests on three tiers: (1) technical infrastructure such as firewalls, encryption, (2) security in data gathering, privacy preserving techniques, and (3) secure applications including authentication, access control, authorization and auditing[20].

With the increasing number of DWH applications, incorporating security into training and education are important. Guimaraes (2006) describes a curriculum that addresses both database security and data warehouse security. Fernández-Medina et al. (2006) propose a model for access control and audit in DWHs. This approach is promising because it supports the specification of security requirements in early stages of establishing a DWH.

REFERENCES

[1] Nabil, R. A., & John, C. Worthmann (1989). Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys*, 21(4), 515–556.

[2] Charu, C. (2005). Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st VLDB Conference.*

[3] Algirdas Avizienis, J.-C., Laprie, Randell, B., & Landwehr, C. (2004). Basic

concepts and taxonomy of dependable and secure computing. *IEEE Transactions of Dependable and Secure Computing*, 1(1), 11–33.

[4] Bell, D., & La Padula, L. (1975). *Secure computer system: Unified exposition and multics interpretation.* Esd-tr-75-306, Technical Report mtr-2997, Bedford, MA: The MITRE Corporation.

[5] Bertino, E., Khan, L. R., Sandhu, R., & Thuraisingham, B. (2006, May). Secure knowledge management: Confidentiality, trust, and privacy. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 36(3), 429–438.

[6] Castano, S., Martella, G., Samarati, P., & Fugini, M. (1994). *Database Security*. Addison-Wesely, ACM Press.

[7] Mitre Corporation (1997, April). *Integrity considerations for secure computer systems.* Technical Report esdtr-76-372, esd,/afsc, mtr 3153, Bedford, MA: Mitre Corporation.

[8] Denning, (1982). *Cryptography and Data Security*. Addison Wesley.

[9] Denning, D. E., & Denning, P. J. (1979). Data security. *ACM Comput. Surv.*, 11(3), 227–249.

[10] Denning, D. E., & Schlorer, J. (1983, July). Inference controls for statistical databases. *IEEE Computer*.

[11] Fernández-Medina, E., Trujillo, J., Villarroel, R., & Piattini, M. (2006). Access control and audit model for the multidimensional modeling of data warehouses. *Decis. Support Syst.*, 42(3), 1270–1289.

[12] Ferraiolo, D.F., & Kuhn, R. (1992, October). Role-based access control (rbac). In *Proc. 15th NIST-NSA National Computer Security Conference*, Baltimore, MD.

[13] Guimaraes, M. (2006). New challenges in teaching database security. In *InfoSecCD ’06: Proceedings of the 3rd annual conference on Information security curriculum development*, 64–67, New York, NY, USA, ACM.

[14] Landwehr, C.E. (2001). Computer security. *Int. Journal of Information Security*, 1(1), 3–13.

[15] Loukides, G., & Shao, J. (2007). Capturing data usefulness and privacy protection in k-anonymisation. In *SAC ’07: Proceedings of the 2007 ACM symposium on Applied computing*, 370–374, New York, NY, USA: ACM.

[16] Machanavajjhala, A., Gehrke, J., Kifer, D., & Venkitasubramaniam, M. (2006). L-diversity: Privacy beyond k -anonymity. In *ICDE ’06: Proceedings of the 22nd International Conference on Data Engineering (ICDE’06)*, 24, Washington, DC, USA: IEEE Computer Society.

[17] Nyanchama, M., & Osborn, S. (1994). Ifip wg 11.3 working conf. on database security. database security viii: Status and prospects. In *Proc. 15th Annual Computer Security Applications Conference*, North-Holland.

[18] Osborn, S., Sandhu, R.S. & Munawer, Q. (2000). Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transaction on Information and System Security*, 3(2), 85–206.

[19] Park, J., & Sandhu, R. (2004). The uconabc usage control model. *ACM Transactions on Information Security*, 7(1), 128–174.

[20] Priebe, T., & Pernul, G. (2004). Sicherheit in Data-Warehouse- und OLAP-Systemen. *Rundbrief der Fachgruppe Modellierung betrieblicher Informationssysteme (MobIS) der Gesellschaft für Informatik e.V. (GI)*.

[21] Samarati, P., & Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its through generalization and suppression. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*.

[22] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996, February). Role-based access control models. *IEEE Computer*, 29(2), 38–47. doi: http://csdl.computer.org/comp/mags/co/1996/02/r2toc.htm.

[23] Sandhu, R.S., Ferraiolo, D., & Kuhn, R. (2000, July). The nist model for role-based access control: Towards a unified standard. In *Proc. of 5th ACM Workshop on Role-Based Access Control*, Berlin, Germany: ACM, ACM Press.

[24] Sweeney, L. (2002). k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5), 557–570.

[25] Thuraisingham, B. (2005). Privacy constraint processing in a privacy-enhanced database management system. *Data & Knowledge Engineering*, 55, 159–188.

[26] Willenborg, L., & De Waal, T. (1996). *Statistical Disclosure Control in Practice*. Berlin: Springer-Verlag.