Database security

From Wikipedia, the free encyclopedia

Database security concerns the use of a broad range of information security controls to protect databases (potentially including the data, the database applications or stored functions, the database systems, the database servers and the associated network links) against compromises of their confidentiality, integrity and availability. It involves various types or categories of controls, such as technical, procedural/administrative and physical. *Database security* is a specialist topic within the broader realms of computer security, information security and risk management.

Security risks to database systems include, for example:

- Unauthorized or unintended activity or misuse by authorized database users, database administrators, or network/systems managers, or by unauthorized users or hackers (e.g. inappropriate access to sensitive data, metadata or functions within databases, or inappropriate changes to the database programs, structures or security configurations);
- Malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services;
- Overloads, performance constraints and capacity issues resulting in the inability of authorized users to use databases as intended;
- Physical damage to database servers caused by computer room fires or floods, overheating, lightning, accidental liquid spills, static discharge, electronic breakdowns/equipment failures and obsolescence;
- Design flaws and programming bugs in databases and the associated programs and systems, creating various security vulnerabilities (e.g. unauthorized privilege escalation), data loss/corruption, performance degradation etc.;
- Data corruption and/or loss caused by the entry of invalid data or commands, mistakes in database or system administration processes, sabotage/criminal damage etc.

Ross J. Anderson has often said that by their nature large databases will never be free of abuse by breaches of security; if a large system is designed for ease of access it becomes insecure; if made watertight it becomes impossible to use. This is sometimes known as Anderson's Rule.[1]

Many layers and types of information security control are appropriate to databases, including:

- Access control
- Auditing
- Authentication
- Encryption
- Integrity controls
- Backups
- Application security
- Database Security applying Statistical Method

Databases have been largely secured against hackers through network security measures such as firewalls, and network-based intrusion detection systems. While network security controls remain valuable in this regard, securing the database systems themselves, and the programs/functions and data within them, has arguably become more critical as networks are increasingly opened to wider access, in particular access from the Internet. Furthermore, system, program, function and data access controls, along with the associated user identification, authentication and rights management functions, have always been important to limit and in some cases log the activities of authorized users and administrators. In other words, these are complementary approaches to database security, working from both the outside-in and the inside-out as it were.

Many organizations develop their own "baseline" security standards and designs detailing basic security control measures for their database systems. These may reflect general information security requirements or obligations imposed by corporate information security policies and applicable laws and regulations (e.g. concerning privacy, financial management and reporting systems), along with generally accepted good database security practices (such as appropriate hardening of the underlying systems) and perhaps security recommendations from the relevant database system and software vendors. The security designs for specific database systems typically specify further security administration and management functions (such as administration and reporting of user access rights, log management and analysis, database replication/synchronization and backups) along with various business-driven information security controls within the database programs and functions (e.g. data entry validation and audit trails). Furthermore, various security-related activities (manual controls) are normally incorporated into the procedures, guidelines etc. relating to the design, development, configuration, use, management and maintenance of databases.

Contents

Privileges

Two types of privileges are important relating to database security within the database environment: system privileges and object privileges.

System Privileges

System privileges allow a user to perform administrative actions in a database. These include privileges (as found in SQL Server) such as: create database, create procedure, create view, backup database, create table, create trigger, and execute. [2]

Object Privileges

Object privileges allow for the use of certain operations on database objects as authorized by another user. Examples include: usage, select, insert, update, and references. [3] –

Vulnerability assessments and compliance

One technique for evaluating database security involves performing vulnerability assessments or penetration tests against the database. Testers attempt to find security vulnerabilities that could be used to defeat or bypass security controls, break into the database, compromise the system etc. Database administrators or information security administrators may for example use automated vulnerability scans to search out misconfiguration of controls within the layers mentioned above along with known vulnerabilities within the database software. The results of such scans are used to harden the database (improve security) and close off the specific vulnerabilities identified, but other vulnerabilities often remain unrecognized and unaddressed.

In database environments where security is critical, continual monitoring for compliance with standards improves security. Security compliance requires, amongst other procedures, patch management and the review and management of permissions (especially public) granted to objects within the database. Database objects may include table or other objects listed in the Table link. The permissions granted for SQL language commands on objects are considered in this process. Compliance monitoring is similar to vulnerability assessment, except that the results of vulnerability assessments generally drive the security standards that lead to the continuous monitoring program. Essentially, vulnerability assessment is a preliminary procedure to determine risk where a compliance program is the process of on-going risk assessment.

The compliance program should take into consideration any dependencies at the application software level as changes at the database level may have effects on the application software or the application server.

Abstraction

Application level authentication and authorization mechanisms may be effective means of providing abstraction from the database layer. The primary benefit of abstraction is that of a single sign-on capability across multiple databases and platforms. A single sign-on system stores the database user's credentials and authenticates to the database on behalf of the user.

Database activity monitoring (DAM)

Another security layer of a more sophisticated nature includes real-time database activity monitoring, either by analyzing protocol traffic (SQL) over the network, or by observing local database activity on each server using software agents, or both. Use of agents or native logging is required to capture activities executed on the database server, which typically include the activities of the database administrator. Agents allow this information to be captured in a fashion that can not be disabled by the database administrator, who has the ability to disable or modify native audit logs.

Analysis can be performed to identify known exploits or policy breaches, or baselines can be captured over time to build a normal pattern used for detection of anomalous activity that could be indicative of intrusion. These systems can provide a comprehensive database audit trail in addition to the intrusion detection mechanisms, and some systems can also provide protection by terminating user sessions and/or quarantining users demonstrating suspicious behavior. Some systems are designed to support separation of duties (SOD), which is a typical requirement of auditors. SOD requires that the database administrators who are typically monitored as part of the DAM, not be able to disable or alter the DAM functionality. This requires the DAM audit trail to be securely stored in a separate system not administered by the database administration group.

Native audit

In addition to using external tools for monitoring or auditing, native database audit capabilities are also available for many database platforms. The native audit trails are extracted on a regular basis and transferred to a designated security system where the database administrators do not have access. This ensures a certain level of segregation of duties that may provide evidence the native audit trails were not modified by authenticated administrators. Turning on native impacts the performance of the server. Generally, the native audit trails of databases do not provide sufficient controls to enforce separation of duties; therefore, the network and/or kernel module level host based monitoring capabilities provides a higher degree of confidence for forensics and preservation of evidence.

Process and procedures

A good database security program includes the regular review of privileges granted to user accounts and accounts used by automated processes. For individual accounts a two-factor authentication system improves security but adds complexity and cost. Accounts used by automated processes require appropriate controls around password storage such as sufficient encryption and access controls to reduce the risk of compromise.

In conjunction with a sound database security program, an appropriate disaster recovery program can ensure that service is not interrupted during a security incident, or any incident that results in an outage of the primary database environment. An example is that of replication for the primary databases to sites located in different geographical regions.[4]

After an incident occurs, database forensics can be employed to determine the scope of the breach, and to identify appropriate changes to systems and processes.

Statistical methods

The greatest threat to database security are non-tracked unauthorized changes by internal and external users. Algorithms based on cryptology and other statistical methods are deployed to both identify these events and report threats to administrators. Such shield DB approach maps large dataset into its small digital fingerprint which, is continuously updated with every change in main database by registered applications. Desired fingerprints are then matched with actual at preset intervals for identifying the changed locations (rows and columns) in the main database, date and time of unauthorized changes, even made through privileged authority.

See also

- Negative database
- Database firewall
- FIPS_140-2 US federal standard for authenticating a cryptography module
- Virtual private database

References

1. Guardian newspaper article on a security breach, in which Anderson's Rule is formulated (http://www.guardian.co.uk/commentisfree/henryporter/2009/aug/10/id-card-database-breach)
2. Stephens, Ryan (2011). *Sams teach yourself SQL in 24 hours*. Indianapolis, Ind: Sams. ISBN 9780672335419.
3. Stephens, Ryan (2011). *Sams teach yourself SQL in 24 hours*. Indianapolis, Ind: Sams. ISBN 9780672335419.
4. Seema Kedar (1 January 2009). *Database Management Systems*. Technical Publications. p. 15. ISBN 978-81-8431-584-4.

External links

- http://iase.disa.mil/stigs/checklist/index.html
- http://iase.disa.mil/stigs/stig/index.html
- http://www.databasesecurity.com/dbsec/database-stig-v7r1.pdf
- Dark Reading - Tech Insight: Database Activity Monitoring (http://www.darkreading.com/security/encryption/showArticle.jhtml?articleID=208803797)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Database_security&oldid=642111776"

Categories: Database security