Securing Databases
Paul Carmichael
April 9, 2001

The security of an organizations database is essential today. Databases are the foundations of business systems such as web servers and ERP applications. The data stored within the databases frequency includes sensitive information such as employee and/or client details, financial and confidential data. Databases have not been subject to the same security of networks and operating systems and are extremely complex systems.

Database Risk Management

Databases are vulnerabilities to security breaches because of the complex nature of the database system, insecure password storage, misconfigured systems or unrecognised system backdoors. To reduce the risk of these vulnerabilities an organization can apply security principles. The following general principles of security should be enforced

1. Less privilege- a user should only be granted those privileges that they need to perform there job function;
2. Defence in depth-multiple layers of protection;
3. The prevention of security breaches is good but the detection of breaches is a must;
4. Encryption should be used whenever possible; and
5. Clear, defined security policy and procedures.

As there are many aspects to security the following are relevant to securing databases

1. **Secrecy and confidentially**- data should not be disclosed to anyone not authorised to access it;
2. **Accuracy, integrity and authenticity**- data should not be able to be maliciously or accidentally corrupted or modified and the origins of the data should be verifiable; and
3. **Availability and recoverability**- database systems should kept working and recoverable in the event of data lost.(Heney, 1998, pp3-4)

In addition to securing the database it important to ensure that the principles of defence in depth are also enforced. Without defence in depth principles applied such as network security, anyone can potentially connect to the database server on the well-known port numbers by passing the OS security. Network security can be enforced through firewalls, router access control list (ACL)'s, and intrusion detection system (IDS)'s. The underlying operating system (OS) should also be hardened or secured to ensure that access to the database cannot be achieved through a poorly configured OS.

Apply security principle to databases

The aspects of security relevant to databases can be implemented through appropriate

1. Authentication of users;
2. Access control to objects and authentication of authorised applications;
3. Administration policies and procedures;
4. Secure initial configuration;
5. Auditing; and
6. Backup and recovery strategies.

1. Authentication of users

It is important to ensure that all users connecting to the database authenticate. Static passwords should

be used as a minimum for all connections. These passwords should be stored securely within the database in a strong encrypted format. For databases that require higher levels of security one-time passwords or public key encryption (PKI) X.509 digital certificate smart cards can be used.

For databases that use passwords, a password policy should define that passwords

- must have a minimum length;
- must contain symbols or numbers and alphabetic character; and
- that are easy to guess should be disallowed.

Databases such as Oracle 8i allow administrations to enforce these password policies.(Oracle, 1999, p2)

2. Access control to objects and authentication of authorised applications

Databases permissions can be controlled through access granted to database objects, such as tables, synonyms, views, indexes, stored procedures or triggers. Controlling access granted to objects should be defined during the design phase. During this phase the database administrators (or designers) should work to define the operations to ensure that the principle of less privilege is implemented.

Synonyms allow access to objects with knowing the owner of the object. A table, *employees* own by *joe* in Oracle would have to be referenced by joe.*employees*. Through the use of synonyms the table can be referenced by *employees*. Synonyms can simply object tracking but assist in hiding the underlying database structures, which helps to protect the database malicious actions, thus ensuring the principle of defence in depth is implemented.

Views can be secured by controlled access through row or column level security or by pre joining several tables.

A new form of architecture called the three-tier requires an application server that holds and executes the application using language such as Java to communicate with the workstation. The applications are executed on the server and communicate with the database. Typically the workstation would authenticate with the application server, and the application server would authenticate with the database server. For the application to authenticate with the database the application would require a username and password (for a secure configured system). The username/password should not be hardcoded into the application, therefore some databases such as Oracle have mechanisms to get around this problem, these should be reviewed before implementation, and such options include trust relationships.

3. Administration policies and procedures

Securing in organizations systems and data cannot be effective without the development of a written security policy. The policy defines the framework that will be used to enforce security and manage risk. Once the policy has been defined a plan can be developed and such a plan would define how to secure the organizations databases. The policy and plan is important because organizations have varying requirements for data protection. Some organizations require tight and controlled external and internal access to data, while others may have quite open internal access to data. The security policy will assist the database and system administrators to ensure there efforts in securing the database are appropriate to the organization. The policy can define standards to be used across database systems such as accounts, usernames, passwords, roles, and objects. Policies and Procedures should also define auditing and logging requirements of the system and managing change control.

Databases are increasingly becoming an integrated component for Web servers, Java applications and other emerging technologies. Unmanaged security vulnerabilities within the database can have a direct connection with downtime, system integrity and consumer confidence. Administration policies must

clearly define how system and database patches will be managed to ensure that all relevant patches are applied.

Stored Procedures in general are executed on behalf of the user but use the privileges of the owner. Good policy would see the schema owners also own all the stored procedures.(Heney, 1998, pp53-4)

Access control to objects and management of users can be simplified through the use of roles. Roles are a collection of privileges that can be assigned to users. An employee management systems role could include database administrator, human resources, payroll, developers and staff. These roles can be defined to include only the operations required to complete the job function, again the principle of less privilege. In addition to roles, profile can be used to control allocation and use of resources to users within the database. Profiles can be used to prevent one user running an operation that is resource intensive, that is in effect making the system unavailable to all other users. Policy should ensure that roles, users assigned to roles and profiles are reviewed regularly to ensure they are still relevant.

Oracle profiles can be used to disable some privileges such as the ability to run SQL queries or commands, this can provide another layer of protection particular for users who do not require the need of SQL queries.

Many databases include default roles, these roles if used can provide users with privileges that they do not require and role names can often be misleading. In Oracle the *CONNECT* role allows a user the ability to do more that just connect to the database. In addition to default privileges of the default Oracle roles can change between versions creating additional security risks if not managed. Policy should ensure that default database roles are not used unless the capabilities of the role are understood. The *DBA* role gives a user almost unlimited privileges, privileges which should not be granted to developers, therefore it is important that policy ensures that no developer is assigned the *DBA* role or equivalent.

Increasingly, database systems can be accessed through dial-up or Internet connections. Staff member leaving an organization are a potential security risk. Policies and procedures within the organization should ensure that their access privileges are revoked as soon as possible to prevent the duplication or damage of data through remote connections; similar procedures should clearly define account requests and authorisation processors.

Databases and the underlying operating system have security weaknesses and new vulnerabilities are discovered occasionally. Policy should state the importance of managing these vulnerabilities and this should be done through the installation of the latest security patches, after of course, through testing.

4. Secure initial configuration

Default database configurations such as Oracle have well-known default accounts and passwords that provide varying levels of access to the database. During the initial configuration these accounts should be disabled and/or the passwords changed.

A poorly configured database may provide the mechanism to compromise an entire network. By gaining access to powerful built in "extended stored procedures" an attacker could potentially gain OS level administration that could be used to attack the remaining network. The database must be set up so that when a stored procedure is accessed they run with minimal less levels.

For the database system to function correctly the database system files must be installed and available. An attacker who can remove a critical file could potentially cause data loss and/or use the database to crash. Database system files must be set-up with restricted read and write access so that an attacker cannot remove and/or modify these files.

Databases such as Oracle require control and configuration files to maintain operation and state; in addition for the need to control access to these files, the database should own these files, not the system administrator or a user. System administrators do not require access to these, and therefore should not be given access.

In addition to managing the configuration, there may be a need for the database to comply with computer system evaluation standards. Vendors such as Oracle comply with varying standards depending on the version number.

A security policy may require that all critical data files are stored in an encrypted format. Some databases support full database encryption, this again adds another level of protection to your database.

5. Auditing

The database includes many features that allow the auditing of database access and operation. In addition to the database auditing features, changes to critical configuration files (such as Oracle init file) should be logged at the end of the file or some other predefined place to maintain a record of changes to the database.

Auditing of the database should assist the administrators to detect any unauthorised or malicious activities. Auditing does impact the performance of the database therefore logging of activities can only assist the detection process or when the logs are reviewed regularly. The security plan and risk assessment will determine what tables or features of the database are critical, therefore where resources are scarce auditing of these may be the most appropriate.

Auditing of actions could include

- unsuccessful attempts to connect to the database;
- startup and shutdown of the database;
- viewing, modifying or the removal of information from tables;
- creating or removal of objects; and
- executing programs.

Audit results can be fed into a separate database and triggers provide a mechanism for this type of auditing. For example Insert operations on a table can be used as a trigger to update a separate audit log table that users and administrators do not have access to. A summary report can then be generated from the table to reduce the time required to review the logs.

The information recorded in the log should be appropriate and meaningful. As a minimum include

- who created the information;
- who has modified the information; and
- what was changed.

The use of 3-teir architectures provides an additional level of complexity to logging. Often the application server will make changes to the database on behalf of the user. It is important that the application server passes on the identity of the real user to the database, again the database administrator should review the documentation to determine the most appropriate approach.

In addition to database auditing 3rd party tools such as ISS Database Scanner can be used. These tools can be used to provide a baseline of the database and audit against it.

6. Backup and recovery strategies

Database corruption, accidental damage, unauthorised or malicious activity can lead to huge losses without appropriate backup strategies. Backup and recovery procedures should be thoroughly tested at regular intervals and off-site backups storage will allow recovery from a disaster.

Recovery procedures should be tested to ensure

- staff are confident in performing all forms of recovery;
- backup and recovery strategies have been properly analysed; and
- you can successfully read tapes on different tape drives from those used to make them.

The backup strategy should also define how the backups are to be performed. There are many options to perform backups including

1. Cold backups – performed with the database shut down;
2. Hot backups – performed with the database up and available; and
3. Logical backups – a snapshot in time, performed with the database up and available.(Heney, 1998, p287)

References

BrainTree Security Software, Security at the Heart of B2B E-Transactions, 1999, http://www.pentasafe.com/whitepapers/B2bwp.pdf (3rd April, 2001)

BrainTree Security Software, Securing PeopleSoft ?Data, http://www.pentasafe.com/whitepapers/PeopleSoft.PDF (3rd April, 2001)

BrainTree Security Software, Client/Server Database Security, http://www.pentasafe.com/whitepapers/Cswp.pdf (3rd April, 2001)

Internet Security Systems, Securing Database Servers, http://documents.iss.net/whitepapers/securingdbs.pdf

Internet Security System, Database Scanner – Getting Started, Version 4.1, Jan 2001, URL http://documents.iss.net/literature/DatabaseScanner/DBS41_ug.pdf (7th April, 2001)

Heney W, Theriault M, Oracle Security, CA, O'Reilly, 1998

Oracle, Database Security in Oracle8i, An Oracle Technical White Paper, November 1999, URL http://technet.oracle.com/deploy/security/seceval/pdf/seceval_wp.pdf (3rd April, 2001)