

Oracle Database Security

Nathan Aaron
ICTN 4040
Spring 2006

Introduction

It is important to understand the concepts of a database before one can grasp database security. A generic database definition is “a usually large collection of data organized especially for rapid search and retrieval (as by a computer)” (Database). This is not much different than Oracle's database definition, “An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information.” (Oracle Corporation) Databases can range from simplistic to complex. An example of a simple database is an address book. An address book provides great functionality but limits itself to specific information. For example, what if you need to include information about the model car the contact drives, or what their favorite food is? Chances are you would need another database. In a business environment it does not make sense to maintain multiple hard copy databases. Businesses must maintain large amounts of data. Examples of data are inventory, finances, payroll, employee information, and sales history.

Like you, businesses cannot afford the risk of an unauthorized user observing or changing the data in their databases. There are several types of concerns that are realized about database security. They are: “unauthorized data observation, incorrect data modification, and data unavailability” (Bertino, E). An example of unauthorized data observation would be a database user accessing information that they are not authorized to view. Incorrect data modification can be intentional or unintentional. Intentional data modification could be a student changing their grade or a data entry clerk accidentally entering the price of a line item incorrectly for an order. Data unavailability exists when “information crucial for the proper functioning of the organization is not readily available when needed.”(Bertino, E).

A person could fill a library on books related to how an Oracle database operates. An Oracle

database is a complex piece of software comprised of networking, memory and file system components. The understanding and application of Oracle database security must be handled in small chunks. It is possible to compromise security by applying security techniques without an understanding of them.

Need For Database Security

Any security solution must meet the following criteria: secrecy or confidentiality, integrity, and availability (Bertino, E). Secrecy and confidentiality may be the most recognized of the criteria. They are both items that end users are well aware of. For instance, a customer stores their credit card information within their account at an online retailer. If the database is compromised, either because of poor database design or poor application design, the customer information is made available. This example compromises the secrecy and confidentiality criterion. The next criterion is integrity. Keishi Tajima of Kyoto University states that "User access to a database is either an action to get some information from the database, or an action to give some information to the database in order to make it reflected by the database state."(Tajima, Keishi) With this in mind we see that database users share a great deal of responsibility. Data integrity ensures that data is being modified by an authorized user and that it is being modified properly. It is possible for a user to be given improper rights to database objects. This means that a user could modify a table that they should not have access to. It is very important for Database Administrators (DBA's) to constantly monitor user security. Imagine an employee changing their salary information.

The last criterion that a database security solution should meet is availability. Availability is ensuring that the database is available through hardware and software problems, and malicious attacks. Although Oracle databases can handle 1000's of concurrent connections simultaneously, it is possible to cause a denial of service attack against the database. For

example, an Oracle database uses a listener that runs on port 1521 by default. The listener listens for connections to the database and hands the connections off to the database. By flooding the listener with requests, it is possible to fill the listener's log file and cause the listener to stop accepting connections. As you can see from these examples it is important to have database security solutions that meet the three criteria.

Oracle Database Overview

There are many database systems on the market. The world's largest commercial database runs on Oracle and is 100 Terra bytes (Oracle Corporation). Although some of the techniques in this document may apply to other database vendors, this paper is written for Oracle. It is important to understand the architecture of an Oracle database. Without a general knowledge of the files and processes that must exist, it is impossible to grasp Oracle database security.

We know that a database is of no value if users cannot access the data contained in it. There are many tools and clients on the market that allow a user to access data within an Oracle database. The Oracle client software includes a tool called SQL*Plus. SQL*Plus is a command line tool that allows users to execute Structured Query Language (SQL) against an Oracle database. SQL*Plus connects to a database listener running on the database server. By default the port is 1521. Once the client connects, the listener starts a dedicated process on the server and passes the connection to the new process. The Oracle Database is capable of operating in either Dedicated Server Mode or Shared Server Mode. The client connection examples in this document assume the database is running in Dedicated Server Mode. Figure 1 shows an example of the connection process.

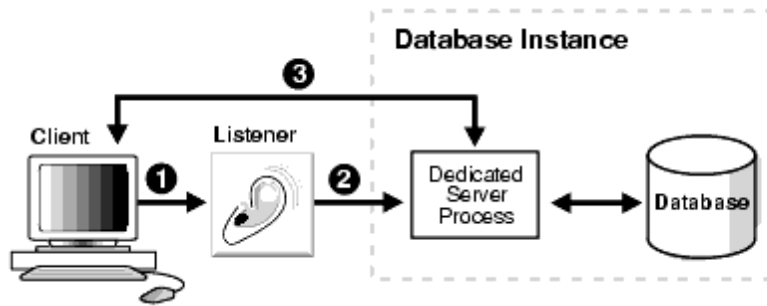


Figure 1 (Oracle Corporation)

The dedicated server process is the link between the client and the instance. An Oracle instance is the combination of memory structures and processes that exist when an Oracle database is started. An Oracle database is a combination of logical and physical structures. An example of a logical structure is a tablespace. A tablespace groups “related logical structures together.” (Oracle Corporation). Tablespaces contain objects like tables or indexes. Oracle is made up of many physical database structures. Some examples are datafiles and control files. To list and describe all components of an Oracle instance or a logical/physical database structure is beyond the scope of this document. It is important to understand that for an Oracle database to be effective, client, network, server processes, and files must exist.

Oracle classifies database security as system security and data security. System security asks the questions, “Has the user provided a valid username and password?” and “What actions can the user perform on the database?” Data security asks the questions: “What objects does the user have access to?” and “Are the user’s actions being audited?” Oracle uses discretionary access control, meaning access to information is based on privileges. For example, you have full access to the objects that you create in your schema. You also have the ability to grant permissions to objects in your schema to other users. For example, can the user SCOTT select or update my salary table.

Oracle Database Security

Now that we have briefly reviewed the client connection to the database, we will review some basic things that can provide a good starting point for database security. First, we will look at security from a network perspective. Next, we will look at security that must exist on the server itself. Then, we will end with ensuring that data and objects within the database are secure. While reviewing these Oracle database security issues we must recognize how they affect secrecy or confidentiality, integrity, and availability.

We know that the client must connect to the database via the listener. The client uses SQL*Net to “transfer requests from the client to the server via the underlying protocol and Operating System.”(Oracle Corporation) Some examples of underlying Operating System protocols that SQL*Net supports are TCP/IP, SPX, and Digital DEC Pathworks. There are security issues inherent in any of the underlying Operating System protocols. These must be addressed outside of this document. The main concern is how we secure the communication between the client and the listener with Oracle technology. The listener executable on Linux and Unix is `lsnrctl`. This file is usually owned by the oracle user on Linux/Unix or administrator on Windows. If a user is capable of logging on to the server as the owner of `lsnrctl` it is possible to shut the listener down. Stopping the listener would mean that new client connections would be denied to the database. Remember that the listener “hands-off” connections to the database, so existing connections should be unaffected. The listener can be managed remotely. This means it can be started and stopped from another machine. It is important to set a password on the listener to prevent remote and unauthorized users from controlling it. The listener is capable of logging requests. It is very important to enable logging and to also review the logs frequently. It is important to archive the listener log file periodically, because if it gets too large, the listener may not function properly on some

platforms. The listener has configuration files that need to be secured. If the database is running standalone on a Linux/Unix server, it should be verified that the permissions on the listener.ora and the sqlnet.ora files are 600. This would prevent any other users on the server from connecting to the database. There are numerous exploits associated with the listener. They range from buffer overflows to denial of service attacks. It is very important to check for new patches periodically. Even though traffic can be encrypted using Oracle Advanced Security never open the listener port at the firewall. Allowing SQL*Net access from the Internet will allow anyone the opportunity to connect to your database. It is also advisable to run the listener on a port other than 1521.

On Linux/Unix the Oracle Instance and Database is comprised of numerous Operating System processes and files. Each Operating System has its own vulnerabilities which must be addressed. It is important to shutdown services like FTP and TELNET that may not be used on the server. Apply all Operating System patches that Oracle has certified for the Operating System that the database is running on. It is important to review Operating System users periodically to ensure that there are no accounts open that may have access to Oracle files. For instance, Oracle has the ability to write trace files for database sessions to the Operating System. Trace files can contain SQL that is being executed against the database. It may be possible for users on the server to view the trace files and see confidential information within the queries. There are other files on the server that should be secured so that Operating System users don't view information about the database or even remove files such as datafiles. Securing datafiles is particularly important. If the datafiles reside on a Linux/Unix system, the permissions on the datafiles should be set to 600. 600 allows only the Oracle database processes to read and write the datafiles. Unless you are encrypting the data Oracle stores the data unencrypted. It may be possible for users on the system to grep

the datafiles to extract information like credit card or social security numbers. On Linux/Unix it is important to monitor what users are in the DBA group. Users in this group can connect to any database running on the server. Once connected, they will have the ability to administer the database. When installing the Oracle database binaries it is important to only install the minimum files necessary. Keeping track of and maintaining patches for software that isn't being used is time consuming. Specific components can always be installed at a future time.

As mentioned earlier, Oracle permissions are based on privileges. Users must access the database using their username and password. Once connected, users are able to create objects in their own schema only if they have a quota on their default tablespace. It is important to realize that users can only create objects in tablespaces that they have quotas on. An example of a user misusing quotas would be if he created a table in the tablespace userdata and then filled the tablespace by adding too many rows to the table. This may cause other users who had objects in the tablespace userdata to not be able to insert or update their data. It is important to examine user quotas periodically. There are many types of privileges associated with an Oracle database. This document will only examine very basic privileges like select, insert, update, and delete. Let's look at an example of assigning privileges. The user SCOTT owns a table called EMP. SALLY and BILL work in HR and need to select and update employee information regularly. SCOTT needs to grant SALLY and BILL select and update on the EMP table. To make things even easier, Oracle supports roles. It would be possible to create a role called HR_ROLE and assign select and update to the role. Once the role exists we assign the role to SALLY and BILL or any other user that needs these privileges. It is important to note that if a user has been granted privileges on an object using the "with grant option", the user can grant that privilege to another user. If we use the previous example, SALLY could grant select on the EMP table to JOE if she had

been granted select using the “with grant option”. This is a security issue because JOE's job description may not require him to have select access on EMP. It is very important to evaluate user privileges frequently. Oracle has auditing functionality that can be configured. Auditing allows DBA's to monitor misuse and abuse of privileges. There are several types of auditing. The first is statement Auditing. Statement Auditing audits the statements that specified users are executing against any schema. Fine-grained auditing audits access to objects based on their content. Schema object auditing allows DBA's to audit statements that are being executed against a specific object. Schema object auditing applies to all database users. Privilege auditing audits system privileges. Any number of users can be specified to be monitored with Privilege auditing. Auditing can be configured to store the audit information within the database itself or at the Operating System level in files. It is important to know that with auditing enabled, tremendous amounts of information will be logged. This information will need to be cleaned up periodically. As mentioned previously, Oracle stores data unencrypted by default. Depending on the version of the Oracle database it may be possible to use the DBMS_CRYPT, DBMS_OBFUSCATION_TOOLKIT, or Transparent Data Encryption to encrypt data. Regardless of the encryption method, the data cannot be unencrypted without a key.

Conclusion

Databases provide users convenient access to data. Database security means that the data is kept secret or confidential, the integrity of the data cannot be compromised, and the data is available when users need it. This paper examined basic security measures that should be taken into consideration when dealing with an Oracle database. Database security is a complex topic that needs to be studied in depth before being put into practice. For an Oracle database to function properly many technologies have to interact. It is important to constantly review network, server, file system, and database security.

Works Cited

“Database.” Merriam-Webster Online Dictionary. <<http://www.m-w.com/dictionary/database>>.

Oracle Corporation. Oracle Database Concepts 10g Release 2 (10.2). Oracle Corporation. October 2005

Bertino, E., Sandhu, R.. “Database security - concepts, approaches, and challenges.” *Dependable and Secure Computing, IEEE Transactions on* Volume 2, Issue 1, Jan.-March 2005 Page(s):2-19.

Tajima, Keishi. “Static detection of security flaws in object-oriented databases.” *Proceedings of the 1996 ACM SIGMOD international conference on Management of data SIGMOD '96*, Volume 25 Issue 2

Oracle Corporation. “The World's Largest Commercial Database Runs Oracle.” 05 April 2006. <http://www.oracle.com/solutions/performance_scalability/winter2005.html>.

Oracle Corporation. Oracle® Database Net Services Administrator's Guide 10g Release 2 (10.2). Oracle Corporation. October 2005.

Oracle Corporation. “Client - Server Architecture - Metalink note 62142.1.” 29 November 2001. 05 April 2006. <<https://metalink.oracle.com>>.

References

Integrigy Corporation. “Oracle Database Listener Security.” January 2004. 05 April 2006. <http://www.integrigy.com/info/Integrigy_OracleDB_Listener_Security.pdf>.

Sinha, Rajiv. “A Security Checklist for Oracle9i.” Oracle Corporation. March 2001. 05 April 2006. <www.oracle.com/technology/deploy/security/oracle9i/pdf/9i_checklist.pdf>.

Loney, Kevin, and Bryla, Bob. “Oracle Database 10g DBA Handbook” Emeryville, California: McGraw-Hill/Osborne 2005.