Thirteen Ways to Make Your Oracle Database More Secure

Mike Dean
Sr Staff Consultant
Database Specialists, Inc
http://www.dbspecialists.com

Introduction

I make no claims that I came up with all of these ideas by myself. There are many Oracle professionals out there that have done years of research into this subject and this paper is a compilation of my experience along with the ideas of many others. I have spent more than 15 years working with Oracle databases, the majority of which have been spent as a Production DBA on mission critical, highly classified databases for the United States Department of Defense.

In my experience, database security is often overlooked, misunderstood and generally ignored. Corporations will spend tons of money and time to address database design, performance, scalability and availability but security always seems to be an after-thought with not enough resources devoted to it.

The reason for this apparent lack of concern seems to fall into four schools of thought:

a) *"I just paid a huge amount of money for Oracle and it should already be secure"*
b) *"My database is behind a company firewall so I am not worried"*
c) *"Nobody knows or cares enough about my data to bother stealing it"*
d) *"I pay my DBAs a lot of money. I assume they are taking care of it"*

In reality, none of these could be further from the truth. Database security is not automatic and a firewall will only provide a thin layer of defense. A misconfigured firewall will provide no layer of defense. From common thieves looking for credit card numbers to industrial spies looking to steal corporate secrets to international spies looking to steal government secrets, there are people out there actively searching for information to steal. Don't be so naïve as to think no one will find your data interesting or valuable. Or, maybe they don't want to steal your data but wouldn't mind knocking your website off the Internet for a while. If a hacker manages to crash the database that supports your e-commerce website, then that will certainly cost you time, money and customers. On the final point, I think most DBAs do not spend very much time actively trying to secure their databases. This is usually through no fault of their own as it is management that sets priorities, tasks and funding which does not allow adequate time for this type of work. It is, however, the DBAs responsibility to educate themselves about security issues and how they may impact the databases for which they are responsible. It is also the DBAs responsibility to make management aware of security issues and their potential impact.

This paper specifically addresses Oracle database security, but many of these ideas are applicable to any type of database and even to IT security in general. This is not intended to be a complete guide to securing Oracle databases, but just some high-level ideas and examples. For a complete Oracle security checklist, you can download the latest from the Center for Internet Security at www.cisecurity.org. These suggestions are in no particular order of importance.

Disclaimer: This paper contains my opinions that I believe to be valid, however I make no guarantee that implementing them will prevent your database from being stolen by bad guys.

1) Insist on strong passwords that are changed on a regular basis

Individuals should have their own accounts protected by strong passwords and the sharing of accounts forbidden. The password strength and expiration policy needs to be enforced by the database in the form of password profiles and the password verify function (available as of 10g). As of 11g, Oracle can enforce case-sensitivity in passwords. Passwords should expire on a regular basis and be locked after a certain number of failed login attempts. The one exception to this rule would be for accounts used by applications to connect to the database. You can leave yourself vulnerable to a Denial of Service attack if someone repeatedly tries to connect with a valid username but invalid password and manages to lock that account.

I recommend you go the extra step to enforce this policy by using a password cracker on a regular basis to identify weak passwords. One such free tool that I have used with success is called woraauthbf, written by Laszlo Toth and is available at http://www.soonerorlater.hu.

I realize that changing database passwords on a regular basis can be a nightmare. With multiple accounts in multiple databases, it can be such a challenge that I think most organizations fail to accomplish this. Implementing a centralized User Management system seems to be the ultimate solution. Oracle does this in the form of Enterprise Users that are managed via Oracle Internet Directory, which is Oracle's LDAP directory. For more details about Enterprise User Management, refer to the Oracle Database Enterprise User Administrator's Guide
at http://docs.oracle.com/cd/B19306_01/network.102/b14269/toc.htm

2) Beware of plain-text passwords

You can have a super strong password but if it is sitting in an unprotected text file, it isn't very secure. Make sure you understand and document all the configuration files that contain passwords so they can be checked for proper permissions. Make sure they are not world-readable and exist only where needed. You will also need to know where these files are so you can change the passwords on a regular basis. In addition, be aware that passwords can sometimes be visible to certain OS commands.

For example, if you run a SQL script at the command line like this:
sqlplus system/oracle@orcl @scriptname.sql
The entire command will be visible to anyone that happens to be logged onto the server when the script is run. In Windows, "tasklist –v" will display the username/password and

on Unix, the "ps" command will do the same. On the other hand, if you run the SQL script like this:
sqlplus /nolog @scriptname.sql
and have *connect system/oracle@orcl* in the SQL script itself, then the username and password will not be visible. (Make sure you have proper permissions on the script). An even more secure solution would be to use OS Authentication or Secure External Password Store and then you can run the script without specifying a password at all.

3) Secure the Perimeter and everything that leads to the Database.

In order to have a secure database, you need a secure database server. For that, you need a secure application server and network and firewall, etc, etc. You get the point. Your database itself can be rock-solid but if someone hacks the server and logs in as "oracle", then they pretty much own your database. A vulnerable web server can allow a hacker to gain access to the network. Once inside the network, they may be able to poke around long enough to find a way into your database server. Make sure that every path that leads to the database is just as secure as the database itself.

A properly configured firewall is the first line of defense to keep hackers out of your network. You can go a step further by implementing Validnode Checking, which is a feature of Oracle that acts as an Access Control List for IP addresses that are allowed (or denied) to access your database. While this is NOT a substitute for a good firewall, it will add an extra layer of protection.

Never expose your database directly to the Internet. I know that it is really convenient to be able to SQLPlus directly into the database from your home computer, but it is very insecure and just asking for trouble. You can (and should) use a Port Scanning tool such as Nmap (www.nmap.org) to detect open SQLNet network ports (1521, 1522, 1526, among others). If possible, avoid even using the standard SQLNet ports.

4) Practice the Principle of Least Privilege

Users should be granted the privileges to access the data that they need in order to do their jobs and nothing more. This will invariably mean more work for the DBA to determine access requirements and do individual grants but it is truly necessary. As a DBA, you should know which users require which access and it should be documented. I have seen many, many times new applications come out of development destined for production where one of the "requirements" is that the application user has DBA privileges. This is horrible security practice and should never be allowed in production.

5) Hire trustworthy people.

This may seem like an obvious point, but its importance cannot be overstated. DBAs, System Administrators, Network Administrators and various other people inside and outside of the IT department all have access to sensitive information. Thorough background checks should be a standard part of the hiring process and this goes for both employees and contractors. When I worked at a large Defense Contractor, all employees were subject to rigorous criminal and financial background investigations and drug testing. Contractors, on the other hand, could basically come in off the street and start working almost immediately. The assumption was that the contracting companies were doing the background investigations. This turned out not to be true in the case of a contractor that was hired as a Software Engineer and then later fired when it was discovered he had a previous conviction for theft and fraud.

6) Use Database Auditing

The Auditing functionality that is available in Oracle is vast and very powerful. Auditing is an absolute must for every Oracle database in which security is a concern. Simply put, if you are not even auditing your database activity, then you are really not serious about database security. Auditing won't always prevent an intrusion or data theft, but can certainly be used to detect one and provide valuable forensic evidence after the fact. In my experience, there is minimal overhead imposed by auditing although you will need to manage the size of the audit trail as it can grow quite large.

For the most part, you want to audit every command that fails and only a handful of commands that succeed. For example, you would want to audit all DDL statements, logon/logoff activity and "alter" commands. You certainly don't want to audit every successful select statement! In some cases, however, you may want to monitor all activity against certain tables that contain sensitive information.

Once you start auditing your database, you should monitor it on a regular basis to look for anomalies. Look for things like failed login attempts, "Insufficient Privileges" errors and "Table or View Does Not Exist" errors. These can all indicate that someone is poking around in your database looking for trouble.

For complete details on how to implement auditing, refer to the Oracle Database Security Guide available at http://docs.oracle.com/cd/B19306_01/network.102/b14266/toc.htm

7) Protect your backups

Stealing a backup of your database is as good as getting the database itself. Encrypt your backups and make sure they are stored securely.

8) Stay current with Oracle versions and apply Critical Patch Updates on a regular basis

Oracle has been improving the security of its software for many years and you should try to stay relatively current. I realize that many people are running older versions of Oracle and are hesitant to upgrade, but you are truly vulnerable to many different attacks if you have an older version. Oracle releases CPUs every quarter to address known security problems. In order to protect yourself from these vulnerabilities, you should apply these patches on a regular basis. These vulnerabilities are real and can be exploited to gain unauthorized access.

9) Use Bind variables

In addition to the importance of using bind variables for performance and scalability, they are also critical for database security by preventing SQL injection. There is a great discussion about this issue by Tom Kyte at http://asktom.oracle.com/pls/apex/f?p=100:11:0::::P11_QUESTION_ID:23863706595353

10) Only install and configure software that is actually needed.

This will reduce the overall attack surface for your database. For example, unless you actually run external procedures from within the database, you should disable the EXTPROC listener configuration. If you are not using XML Database, then don't install it and you won't have to worry about the network ports that it opens.

11) Monitor your database security on a regular basis

Over time, things can change, sometimes without you being aware, that will leave your database vulnerable. A user that once had a really secure password has changed it to "password" and put it on a sticky note on their desk. A firewall that was once air tight may now have huge holes in it. Privileges that you locked down last year have all been unlocked by the application of a patchset or the mistake of another DBA. It is important to regularly audit your database security posture in order to know that you are still secure.

12) Know your data and use encryption when necessary

Within your database, there is likely some data that is much more sensitive than others and you should consider using encryption to make sure it stays secure. Being able to identify the data that needs this extra level of protection is the first step. Financial information, personnel data and classified information are all good candidates for encryption.

Oracle offers many different ways to encrypt data, both inside and outside the database using the separately licensed Advanced Security Option. Transparent Data Encryption

(TDE) can be used to encrypt data in the database and will automatically decrypt it when queried by anyone with appropriate privileges, making it transparent to the application. This will protect the data as it sits in the datafiles but won't protect it from users. In other words, if someone steals your actual datafiles, they wouldn't be able to read the data because it is encrypted but if someone finds a DBA username/password laying around and logs in, they will be able to read the data.

You can also encrypt SQLNet traffic to and from your database with the Advanced Security Option. This can be setup via the NetManager GUI or by setting various SQLNET.ENCRYPTION* and SQLNET.CRYPTO* parameters in the sqlnet.ora on both client and server. More details on all of this can be found in the Advanced Security Administrators Guide at
http://docs.oracle.com/cd/B19306_01/network.102/b14268/toc.htm

13) Use a Security Checklist

When it comes to actually hardening your database, it is important to have a methodical, repeatable approach by which you can accurately assess your overall database security posture. The Center for Internet Security (www.cisecurity.org) publishes and maintains security checklists for Oracle versions 8, 9, 10 and 11. These documents cover a wide variety of database security issues, from default init.ora parameters that should be changed to PUBLIC grants that should be revoked.

Read these documents and start to think about whether your database is in compliance. I would be willing to bet that it isn't. Some of the items seem impractical and perhaps even impossible to implement in every database, but it is certainly a worthy goal to try to come as close as possible to 100% compliance. This document (or one like it – there are others out there) should serve as your baseline for database security.

About the author

Mike Dean is an Oracle Certified Professional who has been working with Oracle databases since 1996, mostly as a Production DBA on government and commercial projects in the Washington DC and Northern Virginia area. Since 2011, he has worked with Database Specialists as a Senior Staff Consultant helping customers with a wide variety of issues on their mission-critical systems. Mike can be reached by email at mdean@dbspecialists.com