# Global Information Assurance Certification Paper

**Oracle Security, don't forget to lock the front door!**
**SANS Security Essentials**
**GSEC Practical Assignment Version: 1.3**
**by Ralph E. Wegner**
**08 April 2002**

**Abstract**
Accounts and passwords, from the viewpoint of many end-users, are perceived as the only way our systems are protected from unauthorized access. Security professionals know that this is only the skin of the onion, the front door. However, because of this perception these may sometimes be overlooked or downplayed by those responsible. Don't let that happen to you. Make sure you place a deadbolt on the front door!

The Oracle family of relational databases uses password authentication. This paper will discuss some of the many ways Oracle and Oracle related accounts and passwords are at risk, and some suggested ways of mitigating those risks.

The intended audience for this paper includes anyone involved in Oracle administration, development or security in an Oracle environment.

**Introduction**
Oracle Corporation has been touted for having secure databases. They have gone so far as to toot their own horn claiming to have an "unbreakable database" in 9i **[10].**. Oracle's reputation for security is to be applauded, but their claim, like any other such claim of invincibility, should be taken with a grain of salt **[5**]. With any "secure" system, there are always more defenses that can be built, because the enemy is always working to break down the walls. This paper's goal is to identify some account issues in Oracle and suggest some methods of mitigation.

**Who**
Who will be responsible for Oracle security in an organization? The end users will use the system most often, so they will need to be sufficiently trained to understand policies, implement some procedures and identify security issues. Administrators will likely be implementing most of the procedures and defining the policies. The Database Administrator (DBA) has traditionally been responsible for everything Oracle, but in the security role today, the System Administrator (SA), Security Administrator, Application Administrator and/or Developer could perform or assist in this role.

Who will manage passwords? A number of factors can affect how accounts and passwords related to Oracle are managed. In organizations with well established IT organizations, DBA and SA roles are well defined and may be more or less segregated depending on size of staff, politics and personalities. In organizations that have bought in to Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM)

systems, the tool may help to drive the business processes and thus the related job functions to include password and account management. Other factors such as the type of business, the size and geographic breadth of the organization can affect who in the organization manages passwords.

*Database Administrators*
Traditionally, the DBA has administered all Oracle passwords and roles. If there is an individual or group with the title DBA, they will likely be somehow involved in the account management, if only for the Oracle specific accounts and the primary data owner account.

*System Administrators*
The System Administrators (SAs) typically work closely with the DBAs and will often have an agreement of sharing roles in the absence of the other. In some organizations, a DBA position may not exist and the SA may take on that role.

*Human Resources (HR)/Security Departments*
A non-IT department such as HR or Security might (understandably) have some concern that only one person or a group might have access to particular data (personnel, salary, address, business, etc.). Such concerns can be quelled by setting up auditing to allow the monitoring of access of specific database objects by a third party.

*Application Administrators*
An application or set of roles may be created or procured that either allows a separate administrator to work hand in hand with the DBA group to create users and roles, or the user account management role can be taken away from the DBA group by having a completely separate account/password system such as SAP/R3.

*Developers*
Developers could be granted account management permissions usually for the purpose of developing an interface for password management.

**What/Why**
Accounts and their associated passwords are two of the barriers that protect Oracle's data from unauthorized access. Grants and roles additionally define user access to database objects.

Accounts in Oracle consist of a user id, password, grants and roles. They are created by someone with administrator privilege (the DBA role). User ids may be one or more alphanumeric characters long and must begin with an alphabetic character. Passwords have the same requirements but may alternately be "IDENTIFIED EXTERNALLY", where the operating system controls access and users do not need to enter additional Oracle account information. Oracle may also be accessed remotely via SQL*Net (pre Oracle 8), Net8 or via ODBC connection.

There are a few different general types of Oracle accounts based on the role they perform:

The SYS account in Oracle is the most powerful. It has access to every object and write access to all the data-dictionary views. If the wrong object is changed from the SYS user, the whole instance could be corrupted.

The SYSTEM account is similar to SYS but only has read access to the data-dictionary views. It is the account typically used to perform most database administration functions. The system account has the DBA role.

An operator account is often created in larger organizations to give non-DBA users sufficient permissions to perform critical functions such as performing backups, exports, etc. while at the same time reducing the chance that a typo or erroneous command would corrupt data.

Data owner accounts own the user data. These types of accounts typically have greater privileges than standard user accounts but only limited access to SYS owned objects. This account is typically used by the IT staff to access the user data, using business rules to perform routine jobs such as periodical (month-end, year-end) processing.

Standard user accounts can have a variety of configurations, and are typically have the least permissions.

Accounts that are IDENTIFIED EXTERNALLY are linked to a specific operating system account. The user logging into that operating system (UNIX, VMS, Windows, etc.) account has access to Oracle without providing additional userid and login information. The logic here is that the account is secure based on the security of the operating system account.
These accounts usually require an account with the naming convention:
**ops$accountname.**

The preceding types of accounts can be accessed through SQL*Net, Net8 and ODBC network interfaces, as well as through the Oracle SQL*Plus interface via the operating system.

Which instance's accounts should be secured? In short, all of them.

There are many different types of Oracle SIDs, The most obvious being the production instance, whose security should be obvious. Some other instances that should be protected might not be so obvious. Development and test instances, although seemingly safe, could hold a backdoor account that was once migrated to a production instance and forgotten. A repository instance for discoverer or another tool could likewise provide a door to that tool for a potential attacker.
Rman, which is Oracle's recommended backup instance, holds critical backup data for your database instances.

Any of the preceding instances might have a db link connection to one of your other instances. With that access the door is open for an unwanted visitor. The wonder of the network has made the IT landscape much less secure than it once was.

If you are using a third party backup tool such as EDM or Veritas, make sure you are using secure communication with it so that passwords don't flow across your network in clear text. The enemy is not always outside the gate!

The combination of Userids and Passwords which gives users access to the database is the first part of an Oracle account. Grants, and since Oracle 7, roles, allow administrators to grant access and specific object privileges to user accounts, as follows:

*Grants* are how Oracle gives a user account access to an Oracle object (table, index, view, etc.). Grants also may be used to give users access (connection) to the database.

*Roles* are a method Oracle uses to simplify account access to Oracle objects. In older versions of Oracle, each user was granted access to Oracle objects separately. This meant that if two users needed the exact same access to the exact same objects (tables, indexes, etc.), an administrator would have to perform identical "grant" commands for each user for as many objects as they each needed. The development of roles, in Oracle 7, allowed the administrator to create a named role, such as "accountant" or "director" which included all the grants for a given function or job function. A user was then granted one (or more) of the roles so that fewer grants needed to be given to individual users and administration was simplified. An administrator could now create roles in advance and then simply grant one role to a user instead of many individual object grants.

**How**
***policies***
The Oracle database should be an integral part of the organization's security policy. If there is not an organization policy in place, the Database group should make their own. In the absence of defense in depth, at least defend your own position!

A good policy on strong passwords coupled with associated education is critical to reducing the security risk to your database.

It has been well documented and clearly proven by many tools that, given the standard character set, which Oracle adheres to, a password that uses many characters and a combination of upper, lower, numeric and specials, is significantly stronger than the opposite.

The downside of a strong password, to a certain extent, is the stronger the password, the more likely the password will be forgotten. This is especially true in an environment where a user needs to remember a multitude of passwords. Unfortunately there is no cut-and-dried solution to coming to a compromise of password strength and rememberability. Today's tools in most operating systems and RDBMSs, including Oracle, offer the ability, to force the use of strong passwords on users. The earlier you get the users accustomed to making strong passwords, the less pain you will have.

Policies on account and password length, composition, expiration and aging are based on system limitations and will help to keep unauthorized users from breaking into your organization's information systems. Password lockout policies can help stop brute force attacks on your systems. Password sharing and compromised password policies should be clearly defined.

Password distribution policy should definitely be accounted for in security policy. In addition to the physical process of how the password makes it from the creator to the user, a password request policy and related forms should exist. Don't forget to make sure Oracle accounts are included in HR's inprocessing and outprocessing checklists!

Change default passwords! Default passwords for many Oracle accounts including system and sys users (dba users) are well known and have default values that must be changed immediately.

Finally, but not insignificantly, the repercussions of not following policies need to be explicitly spelled out, backed up by management and the legal departments and enforced. A policy that isn't enforced is rather useless and can come back to bite you when you really need it.

All of these policies need to be additionally supported by user education and spot-check reinforcements.

***procedures***

How will passwords be managed?

Passwords can be managed from the command line in SQL*Plus as well as Oracle Enterprise Manager (OEM). OEM is Oracle's standard GUI tool for performing simple management of many parts of Oracle. The OEM tool "Security Manager" allows an administrator to add and delete accounts (and roles) as well as modify passwords.

Command line methods of creating an account with a password and altering a password follow:

**create user scott2 identified by abc123YZ%;**
**alter user scott2 identified by rEd&r0ver;**

In Oracle 8i and later, you may alternately type "password" from SQL*Plus and you will be prompted for and old and new password

An application can use an ODBC or an SQL*Net connection to pass account password information from the application to Oracle. It can also be used to change that account and password information.

Giving account and object permissions is accomplished in Oracle with the GRANT command. Conversely, the REVOKE command removes such permissions. Examples follow:

**grant connect, resource to scott2;** -- gives scott2 acct access to system and resources
**grant select on paytable to scott2;** -- gives user access to read from paytable;
**grant insert, update, delete on atable to scott2;** -- gives user access to change atable
**revoke connect from scott2;** -- remove access from scott2 account

Controlling password information for users and groups:

The product_profile and user_profile tables are tools that can assist in password management. They may limit such things as connect time, CPU resources, failed login attempts, password aging and number of sessions a user may have concurrently. These features allow much more granular control of Oracle's default password mechanism. A few significant parameters that should be considered to be placed in the product and user profiles are:

**password_verify_function** Allows you to name a script to control the complexity of a password.
**password_reuse_max** How many times a password changes before it can be reused.

*Auditing accounts and passwords*

Auditing is a method of monitoring access to Oracle and it's objects.

Auditing accounts is just as important as setting them up in the first place! Auditing should be done to monitor access to accounts as well as ensure the correct accounts exist (using the view ALL_USERS) and strong passwords are being used (using a tool such as COPS [**11**], Cybercop scanner [**12**] or Vigilent Password Manager [**13**]).

Auditing can be activated in Oracle by setting the parameter in init.ora AUDIT_TRAIL equal to TRUE and cycling the instance.

Once auditing is accessible, two different types of audits can be performed, login audits and object audits. Auditing in Oracle causes system tables (specifically SYS.AUD$) to grow in size, so, make sure to occasionally remove old records from SYS.AUD$. It is also recommended to move SYS.AUD$ from it's default location in the SYSTEM tablespace.

*Login audits*
Login audits can audit attempts to connect to the database. Some commands for activating login audits are:

**audit connect;**
**audit dba, not exists, resource whenever successful;**
**audit all whenever not successful;**

Once activated, login audits can be tracked by using the views
**dba_audit_connect, dba_audit_dba, dba_audit_exists, dba_audit_resource and dba_audit_trail**

You may show which login audits are active (by access or session) by using the view **dba_sys_audit_opts**.

Login auditing can be deactivated by using a **noaudit** command corresponding to the original audit command, such as:
**noaudit connect;** Make sure to exclude the "by access" and "by session" clauses.

*Object audits*
Object audits allow you to track which specific changes users are making to objects (tables, indexes, etc.). The major change from the other auditing method is that you now add the **by session** or **by access** clause to determine if the audit record is written once each session or once each time an object is accessed, respectively. The **by access** option will make for a much larger report of your object audits. Examples of object audits follow:

**audit alter, audit, comment, delete, grant, index, insert, lock, rename,**
**select, update on BIG_TABLE by session whenever successful;**
**audit all on dba_users by access;**
**audit grant on default by session;**

Object audits are turned off similarly to login audits by using the **noaudit** command.

*Reviewing what you are auditing*
You can use the view sys.sm$audit_config to see what you are currently auditing (tested with 7.x and 8.0).

You will see CREATE SESSION under AUDIT_TARGET if you are auditing CONNECT.

You should know how to secure your audit trail! The following command will alert you of any non-dba users trying to change sys.aud$:
**audit all on sys.aud$ by access;**

Some other useful auditing views include **audit_actions** and **user_tab_audit_opts**.

User accounts can be monitored for unexpected changes through security manager or via SQL*Plus with the following query from the system user:

**select * from dba_users;**

*Other password related procedures*
Some other items related to password security should also be considered:

*Minimal Privleges*
Although not typically defined in policy, part of standard procedure for administrators is typically to give a user account minimal privilege to allow the user to do their job. A UNIX or NT administrator would not give a regular user root or administrator privilege, so why would a database administrator give the DBA role to the same user? This can easily be related to the standard security policy of "deny all, allow some".

The procedure of granting minimal privilege will provide you heartache from users, application vendors (who will insist their product needs many privileges) and yes, maybe even your management, but toe the line and they will eventually see your way. The DBA role allows too much power to a regular user. A thorough definition of the access that a user really needs will usually find that they only need a small subset of what they have asked for.

*db setup files have passwords*
In some versions of Oracle for windows, instance startup files are created that contain Oracle system userids and passwords. These need to be especially protected because they could allow an unwanted user easy access to your system with minimal effort.

*Unprotected operating system files contain passwords*
An application, developer or contractor may at some point place a file containing passwords on the system. These are difficult to identify and control. More common are unsecured scripts that have embedded userid and passwords. These could be detected with some vigilance by using UNIX find or the equivalent tool in your favorite operating system to locate account password strings in files.

*Network security*
Network risks are primarily concerned with passwords can be sniffed on the network.

One specific example of an issue occurs in UNIX.

Using TELNET to execute SQL*Plus in the format:
**sqlplus system/password**
Will result in a valid SQL prompt and an oracle userid and password that is viewable by most Unix users using the **ps -ef | grep sqlplus** command.

There are three primary methods for resolving this issue:

1. Don't let users (typically administrators and operators) use the command in this fashion. Exceptable substitutes are **sqlplus** or **sqlplus system**, both of which will have Oracle prompt you for the remaining information.

2. Ask the system administrators modify the df command to not show the command executed along with the process number or truncate the command to 6 or 8 characters. The SAs usually enjoy a good challenge!

3. Replace TELNET with Secure Shell (SSH), a public domain terminal emulator with greatly improved security. This will help with many security issues that the DBA and SA have to face.

***Additional tools***
Beyond OEM, command line and script based tools that come with Oracle, there are a number of freeware and commercial tools available that can assist in account and password management and security. Descriptions of a few select tools follow:

*Computer Oracle Password System (COPS)*. An auditing system comprised of shell scripts, perl scripts and compiled programs that can be used to verify strong passwords and perform other security checks. **[11]**

*Oracle Advanced Security (ASO)* - since Oracle 8i a separate package that can be purchased from Oracle to add another level of security, including Secure Socket Layer (SSL) on top of Oracle. **[7][8]**

*VigilEnt Password Manager for Oracle* – Oracle password security tools including break-in detection and evasion. **[13]**

*McAffee Cybercop Scanner* - A commercial security tool that has hooks into Oracle. **[12]**

*Using the database as a password security tool.*
In organizations with many systems and administrators and users, password management can quickly become a nightmare. Oftentimes each administrator will develop their own "stash" of userids, passwords and related information. This can and often does become a major issue when an administrator is away and a system needs access. This could be the perfect opportunity for an administrator to refine their design, development and security skills by creating the perfect repository (in Oracle, of course) for systems, accounts, passwords and any other pertinent data. It would have to be secure (ssl, ssh), easily accessible (web based?) and segmented (you don't necessarily want all the administrators to know all the system, root and administrator passwords) to allow all those administrators to track down their cheatsheets and shred them.

**when**
So, when do you need to ensure that accounts and passwords are secure?

1. During the planning stages for information security of your databases and organization.
This is where you set up policies and start preparing procedures to secure your accounts. Usually done once with some routine follow-ups.
2. During the design and creation of the instance.
When a new system rolls in or a test or development system needs creation. This happens infrequently.
3. During account creation, maintenance and deletion.

User turn around depends on the size of your organization and how often their requirements to access data change.
4. During routine auditing of database accounts.
This should happen on a regular basis. Quarterly or more frequently you should test your system to ensure that everything looks as it should.

**where**
You need to watch your own Oracle databases to ensure that accounts and passwords are secure. However, you should also coordinate to make sure that outside agencies that you are connecting to and that are connecting to your databases have secure connections.

**Conclusion**
Oracle is a relatively secure system. Accounts and passwords are the front door that will only stop intruders if you ensure that you lock it. Identify your weaknesses, build defenses up and check for problems so that you can sleep with only one eye open and know that you've bolted the front door to Oracle security.

## References


1. Devraj, Venkat, **Oracle 24x7 Tips & Techniques**, Berkeley, CA, Oracle Press Osborne/McGraw-Hill, 2000.

2. Koch, George and Kevin Loney, **Oracle 8: The Complete Reference**, Berkeley, CA, Oracle Press Osborne/McGraw-Hill, 1997.

3. Theriault Marlene and William Heney, **Oracle Security**, Sebastopol, CA, O'Reilly & Associates, Inc., 1998.

4. **An Overview of Oracle Database Security Features,** Lorraina Hazel, CNE, May 13, 2001 (SANS GIAC Certification Paper),
URL: http://www.sans.org/infosecFAQ/appsec/oracle.htm

5. **Oracle9i Security Flaws Revealed,** Dennis Fisher, February 6, 2002,
URL: http://www.eweek.com/article/0,3658,s%253D712%2526a%253D22437,00.asp

6. **Oracle9iDB Security**, Oracle Corporation, 2002
http://www.oracle.com/ip/index.html?se_offers.html

7. **Oracle Advanced Security (8i),** Oracle Corporation, 2001
http://otn.oracle.com\deploy\security\aso\htdocs\overview.htm

8. **Oracle Advanced Security (9i),** Oracle Corporation, 2001
URL: http://otn.oracle.com/deploy/security/aso/content.html

9. **Oracle Security FAQ**, Frank Naudé, June 15, 2001
URL: http://www.orafaq.com/faqdbase.htm

10. **Oracle's Secure, but Rest of World Isn't,** John Taschek, February 4, 2002,
URL: http://www.eweek.com/article/0,3658,s%253D1869%2526a%253D22335,00.asp

11. **Security: COPS**, Anthony Lawrence, December 1998
URL: http://pcunix.com/Security/cops.html

12. **Password Guessing/Grinding**, Network Associates, 2001
URL: http://hq.mcafeeasap.com/vulnerabilities/vuln_data/9000.asp

13. **VigilEnt Password Manager for Oracle,** Braintree Security Software,
URL: http://www.sqlsecure.com/Products/Password_Manager/password_manager.html