
How to Write an Oracle Security Plan

Marlene Theriault and William Heney

The Johns Hopkins University/Applied Physics Laboratory and Oracle Corporation

NOTE: This paper is an edited excerpt (Chapter 7) from the book Oracle Security, by Marlene Theriault and William Heney, copyright O'Reilly and Associates, October, 1998, ISBN# 1-56592-450-9, reprinted by permission.

Scope

There are many steps to securing your system and its data. But one of the first—and one that too few organizations take—is the development of a security policy which outlines and maps out the enforcement of a security plan. We believe that the creation of security policies and the implementation of a security plan must precede the more operational steps of securing your system and database.

What's the difference between a security plan and a security policy? A *security policy* identifies the rules which will be followed to maintain security in a system, while a *security plan* details how those rules will be implemented. A security policy is generally included within a security plan. A security plan might be as simple as a verbal statement from the highest-level management that all accounts on a system must be protected by the use of a password. Or a security plan might be a thick document spelling out in great detail exactly how security will be implemented within the company's systems. Just as there are many individual needs and many different approaches to security, there are many types of database security policies. We'll present many aspects of these policies; some may or may not apply to your specific organization. A checklist at the end of this paper provides a resource which you'll be able to use to evaluate which features of a security plan are important for your own particular environment. Also, bear in mind that, no matter how thorough a plan appears to be, changing environments can lead to holes in a security system. Therefore, you will need to re-examine your security plan on a regular basis to ensure its currency.

About the Security Policy and Security Plan

Why does a company need a security policy and plan? What's the point of having them? A security policy, included within a security plan, helps to ensure that everyone is in sync with the company's needs and requirements. With a firm policy in place, every employee knows what is expected—what the rules are—and how the requirements are to be implemented. The limits are clearly defined and consistent guidance is provided for everyone. Statements within a security plan can help to ensure that each employee knows the boundaries and what the penalties of overstepping those boundaries will be. For example, the following are clear, concise rules which employees can easily understand and follow.

- Always log off the system before going to lunch.
- Never share a password with anyone else.
- Never bring software from home to put on your machine at work.

In order to have a truly solid and meaningful policy defined, the highest level of management needs to be committed to ensuring that the security policy will be enforceable. The security policy might state:

- Any employee leaving a computer unsecured will be formally reprimanded.

- Any employee found sharing a password will be suspended for one day.
- Unlicensed software found on a personal computer will be removed and the personal computer user will be shot at sunrise. Survivors will be prosecuted.

Under certain circumstances, the requirement to have and enforce a security policy may come from an agency outside the company. In the case of banks, external agencies control and define what constitutes security for the databases within the bank. The company, however, might decide that even more rigid standards are necessary or that further definitions are required to ensure that no financial transactions become compromised and that confidentiality is maintained. The bank may implement a security plan that further defines exactly how the standards will be implemented, maintained, and audited.

Management Considerations

Once you've determined that a security plan is required for a company, a person or team of people will be needed to begin to define the policy. This paper assumes a team of people.

Who's on the team?

Members of the team might include the person or people who will be administering the system, one or more application owners, and at least one management person who is high enough in the corporate structure to ensure that the policy—as written—will be enforceable. The system administrator may have a different perspective from the database administrator on what comprises security on a system, but the goals of both should be the same—to ensure that the system cannot be compromised. If the policy is to be in effect across divisions within a company, representatives from each division might be included to ensure that there will be “buy-in” across the entire corporate structure. The goal is to include enough people working together to ensure that all areas of corporate need are met, but to keep the group small enough that the team will be able to create an effective, workable policy.

Establishing overall requirements

After the team is assembled, you'll first need to identify the overall requirements of the organization in regard to system and database security. The requirements might include (but are not limited to) the following:

- A uniform approach to security across computer systems and across databases
- The form and style of authorization required to initiate the creation of an account
- Who handles user account creation for operating system, applications, and databases
- How those accounts will be created
- If standard conventions for usernames and passwords should be imposed and enforcement
- Whether password aging will be enabled and in what time frame
- A determination of access requirements on an application-by-application basis
- Identification of users tracking to ensure that as an employee's job description or location changes, the access to applications remains correct
- Identification of sensitive information and an outline of steps to take for data protection

- Penalties to be enforced as a result of different levels of security breaches

Operating System Security Mechanisms

In implementing a requirement for a uniform security approach across platforms, you'll need to take into consideration the native security mechanisms available on each platform. For example, most operating systems require each user who will be interacting with that system to have a unique username and password. Access by a user on a UNIX or OpenVMS system can be further restricted by placing that user in a specific group. Group membership might enable a user to interact with specific directories on the system. The security plan could detail each of these requirements.

Identifying Key Components

We recommend that you use a spreadsheet approach to identify the components within your company that the security plan must encompass—for example:

- Each division within the corporation to be included in the policy
- Each platform within the division
- Each database housed on each platform along with its function (Dev., test, prod.)
- Each application supported within each database
- The “owner” or person responsible for authorization of users within the application
- Required security controls for each application, such as roles or grants required
- Username and password composition
- Type(s) of accessibility (Telnet, client server, external identification)
- What form of authorization will be accepted for that application (electronic authorization, verbal, email, hard-copy form, World Wide Web)
- Person authorized to create the accounts for each application
- Forms of backup which will be implemented
- Recovery procedures which should be used
- Database availability
- What type of auditing will be required
- Who will perform the auditing
- How auditing will be performed

A possible spreadsheet might look like the one shown as follows:

Component	Database A	Database B	Database C
Platform/Division	Windows NT (Div. X)	Digital UNIX (Div. Y)	OpenVMS (Div. Z)
Database/SID Name	mary/abc75!d	ralph/xyz_4u	ed/urok+2me
Database Function	Development/test	Production	Pre-production
Application(s)	Accounts Payable	Human Resources	Office Equipment Locations
Application Owner	H. Brown	Personnel Manager	Facilities Manager
Username	User-defined	First initial/last name	Three letters, two numbers
Password	User-defined	2 letters, 1 number, 1 punctuation mark, 3 letters (e.g., XX#!XXX)	Any combination of letters, numbers, and punctuation, up to 8 total
Access Type	Client Server	Log on to application to connect to database	Telnet to platform and connect to database
Authorization Mode	Email	Paper form signed by head of HR	Phone call
Person to Create Account	Application DBA	HR Security Clerk	Database DBA
Auditing Type	Connections to database	Connections to database SELECT FROM salary table	None
Form(s) of Backup	Exports nightly No archivelog mode	File-level backups weekly Archivelog mode on Exports nightly	Exports nightly No archivelog mode
Recovery Procedure	Rebuild database and import	Recover per procedures in the System Recovery Document	Rebuild database and import
Database Availability	Mon–Fri 7:30–18:00	7 days a week, 24 hours a day	Mon–Fri 7:30–18:00
Auditor	Accounts Payable Manager	HR Security Clerk	N/A
Roles Required	ap_clerk ap_manager	hr_clerk hr_developer hr_manager	None
Grants Required	CREATE SESSION, SELECT FROM ap tables, INSERT/UPDATE ON ap tables (clerk, manager) DELETE FROM ap tables (mgr only)	CREATE SESSION, SELECT on specific tables (clerk) INSERT/UPDATE specific tables (clerk) SELECT, INSERT, UPDATE, DELETE on all tables (manager) CREATE TABLE, TRIGGER, PROCEDURE, etc. (developer)	CREATE SESSION, SELECT, INSERT, UPDATE, DELETE ON ANY TABLE

As you can see from this table, many different security requirements may be present in one environment.

Types of Accounts

From the earliest releases of the Oracle database, a mechanism has been provided to let users connect to the database in order to perform tasks via user accounts. There are several different types of Oracle user accounts—both operating system and database—which a company might implement:

- Although they are created with the CREATE USER command, some accounts are used to house application schemas. These accounts own objects like tables, views, indexes, triggers, procedures, etc.

- Another type of account is used by Oracle itself to enable the database engine work to be performed; these accounts are *sys* and *system*.
- In later versions of the RDBMS, an account to enable the intelligent agent to connect to each database is automatically created during database creation. This account is *dbstmp* and carries full DBA privileges.
- Each application might need one or more accounts to enable work to be performed.
- Each user in your system may require an individual Oracle account with specific privileges to enable the user to work with an application.
- One or more accounts may be needed to enable one or more DBAs to perform database maintenance and duties.

Each account type must be considered and a decision reached on whether that account type will be used and how it will be set up and administered. In smaller organizations, there may be little need for some types of accounts discussed in this section. In very large organizations, there may be a need for more extensive divisions of database account types.

Administrator Accounts

The most obvious account type is the one used for database administration. A small company might have one person acting as system administrator, database administrator, and network administrator, while a larger company might have several people acting as administrators for specific areas. At your site, who will have access to the code area for installation and maintenance of the Oracle software? There may be one or more accounts which will need to be established for various administrative tasks as well as privilege sets—both operating system and database privileges—to perform the required tasks.

Security Manager

Who will serve as the security manager at your site? This person will be responsible for creating user accounts and monitoring user access and database security. In smaller organizations, the DBA will probably handle these tasks.

Application Manager

For each application, there will be developer accounts, application user accounts, and, possibly, application coordinator accounts. An application developer will need the ability to create objects and to code triggers, procedures, etc., while an application user might need only `CREATE SESSION` access. In larger companies, people are sometimes designated as “application coordinators.” Their job is often to handle the administrative tasks for the specific database in which their application is housed. They may need all of the rights and privileges of a DBA—both at the operating system level and at the database level.

Network Manager

There may be a need for a network manager whose job is to administer the network products and oversee Names Servers and network configurations for Oracle databases. Will a separate person or people be needed for this task at your site? Again, in a small company, the individual DBA may be responsible for all Oracle database networking tasks.

Application Schema (User) Accounts

Oracle's mechanism for creating schemas in which application objects are stored is to actually create a form of "user" account, generally referred to as a "schema," which will probably require more privileges than a general user account. The security policy might state what specific system privileges an application account may or may not be permitted to have. In an Oracle version 8.0.3 database, there are 89 distinct system privileges available; these privileges are listed in Chapter 5, *Oracle Default Roles and User Accounts*, for the book **Oracle Security**. Few, if any, applications have a need to hold all 89 privileges. A security plan might list the privileges that are *never* to be given to any application. Almost all of the privileges that grant ANY should be carefully examined and evaluated for listing in the security plan. Here are a few very good examples of privileges that should never be granted to an application (without a really strong business case):

```
ALTER/CREATE/DROP ANY INDEX/PROCEDURE /TABLE /TRIGGER, etc.  
BECOME USER  
GRANT ANY PRIVILEGE /ROLE  
EXECUTE ANY PROCEDURE  
UNLIMITED TABLESPACE
```

General User Accounts

More and more, applications are being provided with user validation present within the application itself. In these cases, a user account is created in the application. The user connects to the application and the application connects to the database on the user's behalf. The user's privileges are determined within the application. In cases where the application provides the user access control, the security plan might simply list what those accesses are and what user type should be granted which privileges.

In applications that rely on database authentication, a careful examination of what privileges a general user should receive may need to be identified and detailed in the security plan. In both Oracle's Financials Applications and the PeopleSoft HR packages, control of user access to the database is performed through the application's own internal utilities. Regardless of whether an application's security relies on the Oracle RDBMS or the application's own code, the security plan should clearly state what type of user accounts will receive each level of privilege in the system.

Oracle's HR/Payroll product, in releases 9 and 10 of the application, allows "secure users." Oracle recommends that the "secure users" account be one which is created with CONNECT and RESOURCE privileges. However, a tablespace quota is made available to these accounts. With the CONNECT and RESOURCE roles both assigned to an account, the explicit UNLIMITED TABLESPACE privilege is given so a user with these roles can create objects in *any* tablespace—with *no* overt quota assigned. With the CONNECT role assigned to an account and without a tablespace quota being assigned, the account is unable to actually create objects.

Standards for Accounts

You will need to determine the mechanism for the creation of new accounts. There are many possible mechanisms. One form of account creation that is gaining popularity is for a company to enable "restricted" access to their sites on the World Wide Web. A person who wants to access a more privileged area of a web site might be required to register with that site through electronic registration. The person is presented with a form requesting his name, company name, address, email address, and other information. He might be prompted to select a username and password. He submits the completed form and, within some space of time, receives in his

email account an acknowledgment that he has registered, along with notification or verification of a username and a password for his use in accessing the site. At no time has the person seen or talked to a human being, but he has now been authorized as a user on a system.

In this example of web site access, we are not judging the procedure or security. (If we were, we might question sending a password via email.) We are merely outlining a general method of a request for an account in which the requester and the administrator have never seen or spoken with each other. Other forms of more anonymous account requests would be via telephone voice mail or electronic mail. On a more personal level, a meeting might be held between the administrator and the requester.

Possible Account Requests

The policy team or higher-level management must decide whether access to the database can be granted through an electronic request, or whether some level of management must physically sign a form acknowledging that the employee seeking entry into the system is a valid employee with a proven need to interact with a specific application area. The authorization required might even be as casual as a user picking up a telephone or walking into a designated person's office and saying, "I need access to xyz system" and receiving an account on that system. Thus, we see that account requests might be generated through a number of different venues:

- Electronic requests via a web site or email
- Telephone
- Hardcopy form—with or without a signature of authorization
- Personal interaction with a verbal request or hardcopy form

or a combination of several of the approaches listed here.

Contents of the Form

If a physical record must be made available for future auditing of the system, you'll need to create a form for that purpose. The security plan would include a copy of the form. A form might include the following information:

- The requester's full name
- Physical location
- Telephone number
- Employee number
- Username and initial password
- Access required
- Platform to access
- Database to access
- Type of work to be performed
- Signature of person authorized to approve the request

- Date by which the account is needed

A sample form might be as simple as the one shown here:

Your request will be processed in accordance with the standards that have been established for the Computing System Services. When your access has been established, you will be notified by the Security Manager and/or the Application Administrator.

In applying for access to the XYZ Company applications, I agree to exercise due care to protect the security and integrity of the company data.

_____ Signature of Applicant	_____ Date		
Please get the appropriate signature(s) for the application(s) below and forward the completed form to the Computing System Services Administrative office, Room 3124.			

ORACLE Development			
Application	Option	ADD/DELETE	Authorization
_____	Developer	_____	_____
_____	Test User	_____	_____

For Office Use Only			
Date Received	_____	Date Entered	_____
Remove User	_____	Remove Project	_____
		Userid	_____
		Date Closed	_____

Ways to Create an Account

There are several possible approaches to creating accounts; for example, you could develop a utility to be used by one or more people to create the accounts on one or more systems. An alternative would be for each system administrator and each database administrator to interactively create the requested accounts for their system. Some applications support registration of users within the application and then rely on a single logon to the database. Once a user has logged on to the application, the application connects to the database using a single “super user” account. In this scenario, each application administrator might be in charge of creating the user registrations within his application. The routine that could be activated through a command procedure might be quite simple. The SQL code that would be run might look like this:

```
PROMPT
PROMPT You will be prompted for a username and password.
PROMPT
CREATE USER &&username
IDENTIFIED BY &password
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp;
PROMPT
PROMPT You will be prompted to enter the list of privileges associated with this user
PROMPT
GRANT &privileges TO &&username;
EXIT;
```

This is obviously a very rudimentary routine, which is shown here only to convey the idea of a possible script. You will probably want a much more robust utility at your site.

Standards for Usernames

There are several different types of standards for usernames. What username standards will you enforce at your site?

Advantages and Disadvantages

Let's consider a uniform approach for usernames across systems within a company. Such an approach has several benefits:

- It is easier to administer than randomly generated usernames or usernames selected by the user.
- It ensures that the username will be the same for each operating system, each database, each application, and for email interaction.
- It can require the inclusion of specific characters or numbers within a username, so a standard makes it easier to ensure that those requirements are always met.

A possible disadvantage to having a username standard is that anyone who has been associated with the company may have enough information to be able to determine any employee's username easily.

Suggested Username Standards

In the case where a username is constructed using part or all of a person's actual name, the username is easy to remember; you only need to know what the standard is to determine what the composition of the username is or will be. An example of a standard using parts of a person's name as a username would be the following:

- The first three letters of the person's first name.
- Plus the first letter of the middle name
- Plus the first four letters of the last name
- Plus a designating number at the end to fulfill the requirement of some operating systems to include both alphabetic and numeric or special characters in a username.

The name Mary Lou Janes would be translated to the username *marljane1*.

Following this standard would make all usernames a maximum of nine characters long. You could go one step further and give special significance to the number chosen. The number "1" could be used to indicate that this is the first employee to have a particular username. If more than one employee has the same first and last name, the first employee would receive the number "1," while the second employee would receive the number "2."

If more than one employee has the same first, middle, and last names, the standard would need to include exception handling. The final username for an employee named Ralph Kenmore Scott might be *ralkscot1*. A second employee named Ralph Kenneth Scott would end up with the username *ralkscot2*. A third employee whose name is Raldania Karen Scott would end up *ralkscot3*. Since this is a potential problem with username standards, we recommend including a further identification mechanism like a comment line designating which username has been assigned to which user. The publication of usernames within a company private phone book helps to distinguish who has which username.

In the case where a user's actual name is shorter than three letters for a first name or last name or where no middle name is present, the username would be shorter than the nine alphanumeric characters normally used. The

standard would outline the possible exceptions that are to be allowed. Thus, Ed Bin (no middle initial) would receive the username *edbin1*.

You might also want to consider having different standards for usernames for different types of accounts. Perhaps you could have one or more of the following types of standard formats:

- User accounts which do not own objects
- Accounts which do own objects (schema accounts)
- Accounts which belong to DBAs and/or security managers

Standards for Passwords

Many children have clubs in which a secret word is used to let us gain entry to the clubhouse. In my club, the password was *hobgoblin*. Since no one but our group knew the secret word, we could feel pretty confident that someone saying *hobgoblin* at our clubhouse door was a member to be allowed in. Operating systems and the Oracle database use passwords in much the same way.

Password Decisions

When you are developing your database security plan, you'll need to make a number of decisions about password use at your site:

- Whether a user will be permitted to create and change his own password
- How frequently the password will expire and how long a grace period will be allowed before the account is locked
- Whether a set standard for password composition is to be used, and what that composition will be
- Whether account lockout will be enabled, whether the account can be automatically unlocked, or whether a security manager will have to intervene to unlock a locked account
- Whether a password will be permitted to be reused, and what length of time must pass before a password can be reused
- How the user or designated account manager will actually change the password—through a created form, through a SQL script, etc.
- If users will not be permitted to change their own passwords, the mechanism by which users will be notified of password changes

The decision to enforce a specific pattern for passwords raises the question of just how secure the password will really be since anyone who knows the imposed template will know the form passwords for the system must take. A previous employee could potentially pose a security threat because the username and password structures are known to him or her. If a template for passwords is to be used, we recommend that you make that template complex enough to ensure greater difficulty in breaking the password security.

Oracle8 supports a number of new password features. You'll need to consider all of the following when you set password standards:

Password aging and expiration

To help ensure that a password will not be compromised, we suggest that passwords be changed on a system at least every three months. The decision of whether to enforce password aging and expiration should be identified in the security plan. The longer a password remains in effect, the greater the possibility of password compromised.

Password reuse

If a password will be permitted to be reused, we recommend that you consider restricting its use to no more frequently than every seventh password cycle. A better approach would be to completely exclude a password that has been used from being used by that person again.

Failed login attempts

The number of failed login attempts that will be tolerated on a system should be defined in the security plan. You must also determine the actions that should be taken when the number of failed login attempts has been exceeded.

Account locking and unlocking

The decision might be made to never enable account locking or never enable automatic account unlocking. Either decision should be documented within the security plan. If account locking is going to be enabled, you will need to define the personnel who will be in charge of performing the account unlocking.

Changing Passwords

Since passwords in Oracle8 can now be aged and expired, the decision about whether users should be allowed to change their own passwords becomes a bit more complex. If the user is not going to be permitted to change his or her own password, then the use of password expiration might not make sense in the company's plan. Determining who will be able to change passwords, and how often the passwords will be required to be changed, becomes vital if the user is not permitted to change passwords. You must also identify, in your plan, the mechanism by which users will be notified of their password changes. If a large number of users' passwords are going to be changed at the same time, the only feasible way to notify them all quickly might be email. In some sites, the requirement has been to hardcopy and distribute new passwords via regular "snail" mail. Phone calls might be another less secure way to relay password changes when notification must occur very quickly. We do not recommend using telephones to relay password changes since the person on the other end of the phone may not be the person to whom you think you are talking.

If a user will be permitted to change passwords, you must define the mechanism for changing those passwords. You might want to create a utility that enables the user to change passwords—especially if direct SQL access is not going to be permitted in the system. Something as elaborate as an Oracle Form or as simple as a SQL statement might be required for a password change. The ability to force the choosing of a password other than one which has already been used is now available (password history), so you will need to specify the length of time that must elapse before a password can be reused.

Standards for Roles

In the spreadsheet shown earlier in this paper, references to "ap_clerk," "hr_manager," and "hr_developer," among others, were used in the "Roles" area of the chart. The convention displayed there was the application name coupled with a task nomenclature—for example, the Human Resources application (hr) coupled with the "clerk" tasks of entering or updating data within areas of the hr application. In this application, the ability to

delete information was not a duty which was deemed appropriate for a clerk to perform. Only a manager can delete information.

The security plan team must decide on the naming conventions which will be used for role creations on a database-by-database and application-by-application basis. The composition of each role (who will be allowed to perform what actions) will also need to be identified, as well as the designation of who will create and assign roles for each application in each database.

Oracle-Supplied Roles

By default, until Oracle version 7.1.6, Oracle supplied three default roles within a database (CONNECT, RESOURCE, and DBA). From version 7.1.6 forward, Oracle supplies two additional roles (SYSDBA and SYSOPER).

Because the composition of these roles has changed from version to version of the RDBMS, we recommend that DBAs define their own roles for user access. For example, in Oracle's version 6, the RESOURCE role was granted to users who were performing development tasks within a database because the RESOURCE role included the ability to create tables. In Oracle7, the ability to create tables appears in the CONNECT role. However, no tables or indexes can actually be created without a tablespace quota being granted to the user.

Granting Access to the Database

Oracle provides the ability to grant privileges directly either to specific users or to roles. The security team will need to decide whether privileges will ever be directly granted to a specific user or will be granted only through roles. If direct grants are allowed, you'll have to decide under what circumstances they will be used. During application development, for example, the developers will have access to the application schema and will have, through that account, many direct privileges.

Oracle provides the ability to grant privileges in the GRANT statement WITH GRANT OPTION, WITH ADMIN OPTION, or without any option. The security plan should designate whether these options will or will not be permitted within the databases being defined.

Standards for Views

Views are wonderful mechanisms for hiding data from different classes of users. For example, I would not want my salary to be visible to the majority of employees in my company. Suppose that my salary is resident in a table called EMPLOYEES. The table might be comprised of columns for the employee's name, location, telephone number, manager's name, department name, and salary. In this case, the only information within this table that might be considered sensitive is salary. Therefore, you might create a view called "emp_view" to display all of the columns except the salary column.

As with roles, the security plan will need to define any conventions for view names, a designation for who will be permitted to create views, a designation of who can grant access to which views, and an identification of who can say that a view is, in fact, necessary or unnecessary.

Standards for the Oracle Security Server

In Oracle8, a new Security Server facility has been supplied to either stand alone or work in conjunction with several third-party vendor products for generating certificates of authentication. These certificates enable users to

connect to various applications and databases without having to provide a username and password each time they want to change connections from one place to another. The Oracle Security Server is discussed in Chapter 15, *Using the Oracle Security Server*, of the book **Oracle Security**.

If a company is planning to implement, or has implemented, a single sign-on utility for user authorization, the security plan should detail who will administer the Oracle Security Server and how users will receive accounts and interact with the Security Server.

Standards for Employees

Within your organization, there will be many different interactions with employees. You will need to determine how to deal with these interactions and add your organization's policies to the security plan. The driving force behind any security policy should be what an employee needs to know to perform his or her job effectively.

Employee Procedures

What privileges do particular employees need? What limitations should you place upon these privileges? A major problem with trying to determine how employees will be able to access a database or application is the need to balance giving enough privilege to enable the employee to get the job done against the risk of allowing too much access to sensitive information. If a security plan becomes too rigid, employees may be made to feel that they are not trusted or may not be able to perform their jobs effectively.

Pre-employment tracking

Before an employee is ever hired, an employment application, resumé, or both, is usually submitted for consideration to a company. Many companies track their candidate submittals using computer programs that interact with a database. The information presented in an job application or resume is private and must be handled with care. Your security plan should include procedures for employment application and resumé handling.

New hires

Once an employee is hired for a position, the security plan should clearly state the steps to be followed for giving a new employee access to platforms and databases needed to perform his or her job effectively. If you've already determined the possible functions for an employee using a specific application, the task of knowing what accesses (roles and grants) are needed by that employee will be made much more easily.

Changing positions

Over a period of time, the employee will learn and (you hope) grow within your organization. He may find that the job he was hired for is no longer challenging enough, or he may receive one or more promotions. With promotions, the employee may require different privileges to perform his new duties. Or an employee may take courses or receive in-house training and become proficient enough to apply for and obtain a position in another department or branch or division of the company. In each of these cases, the security plan should cover what actions should be taken as an employee changes positions within the company.

The disgruntled employee

No one likes to think about the possibility that a coworker may become so upset that he might take action to intentionally damage part or all of a database or system. Unfortunately, this possibility does exist and you must address it. Many years ago at a company in which Marlene Theriault worked, there was a Christmas office party where an employee enjoyed a little too much wine, went back to his office to do a little work before the holiday break, was unable to log on to his account because he mistyped his password repeatedly, decided that his boss had modified his account to deny him access, and relocated some of his boss's data files to another directory location as "payback" for the assumed insult. The good news is that the action was easily corrected and the boss was understanding enough to let the employee off with a mild reprimand. Under less amicable circumstances, however, major damage could have been done to the system since the employee involved had enough privileges to potentially cause real harm. You need to periodically review employee privileges. Be sure such issues are addressed as policies in the security plan.

When An Employee Leaves

Another threat to a system is the employee who either gives notice or is being terminated. Although there are many different philosophies on how to best handle a terminating employee, the security plan should outline exactly what actions are to be taken.

Termination types

The following are a few of the possible termination types to be addressed in a security plan:

- Immediate dismissal, including escorting the employee out of the building with or without severance pay in hand, because the employee has been involved in a major breach of conduct or ethics
- Giving the employee the option of working for a period of time—usually two weeks—before departing the organization; this is usually done in cases where an employee has voluntarily given notice to pursue other interests
- Giving the employee an extended amount of time to continue working and possibly even providing company assistance in a job search; this may be done when a large number of employees have been laid off (downsizing)

The policy should have clear directions on what steps are to be taken in each situation. For example, if an employee is being escorted out of the building immediately and both management and HR are in agreement about the dismissal, the policy might state that the company should revoke all access to all computer accounts and all databases before confronting the employee.

When an employee gives notice

When an employee gives notice, many different approaches may be appropriate. If the employee works directly with the systems or databases in a support capacity, the company might choose to offer two weeks severance pay and have the employee leave immediately rather than take the chance that the employee will do any of the following:

- Work himself to excess trying to clean up every possible task before departing and make careless, costly errors due to fatigue

- Develop a “short-timer” attitude and do little or nothing for the two weeks, which can be very demoralizing to those employees who are not leaving
- Cause damage to the system by inserting a time bomb or Trojan horse set to go off after his departure to sabotage the system or a specific manager’s data area

On the other hand, in the situation where an employee has proven his trustworthiness, you might consider maintaining his accounts even after he has left—for a specified amount of time. If there is a chance that the employee might someday return to the company, access to accounts might be locked instead of explicitly revoked or deleted from the system. This approach is particularly useful for workers such as consultants, resident contractors, or temporary hires who may work sporadically with the databases or systems.

The curious employee

Many people are curious. Some have a desire to know as much as they can about their surroundings, while others just love to poke around and see how much information they can access that they are not supposed to know. The curious employee poses a difficult type of security challenge; there may be absolutely no malicious intent involved, but such an employee might nevertheless violate the privacy rights of others. The decision about what penalties should be imposed on an employee who stumbles into data areas to which he should not have access should be considered and clearly defined in the policy.

User Tracking

In the last several sections, we’ve shown that an employee does not necessarily spend his entire professional life in the same job or the same area of the company. There is, therefore, a need to keep track of each employee’s location and current job requirements to ensure that the employee’s system and database access rights remain appropriate as he moves around or changes jobs. Let’s follow a bit of the employment history of our old friend, Ima Tickdoff.

When Ima came to work for her company, she started as a clerk in the mailroom. Initially, Ima needed access to only one database application to be able to look up employee locations. Ima was bright and very clever and learned quickly. Over a period of time, Ima attended night school, obtained other skills, and was eventually brought into the payroll area. At that point in time, Ima’s computer access needs changed. She now required access to three different databases to perform her varied tasks.

An employee, Ima Tickdoff, became unhappy with her boss. Her unhappiness grew to a point where she finally requested a transfer to the billing department. When she moved to her new department, it was no longer appropriate for her to have access to any payroll information but she now required access to the billing applications. What mechanism was used to ensure that Ima’s access to payroll information stopped and her access to billing information began? How was a user tracked in Ima’s company? How are users tracked in your company? What vehicle ensures that an employee does not have inappropriate access to sensitive data?

You can use your organization’s security plan as the vehicle for specifying how users and their privileges and accesses will be tracked within your company. In one organization, a user tracking application was written to populate a database nightly with current employee information. The application includes each operating system and identifies the databases on each system. Each application manager, DBA, and all developers associated with each database are recorded within the user tracking application, as well as everyone who has an account on each system and database. All privileges and roles for each database are housed within the user tracking application, and each user who has been granted each role and/ or privilege is recorded with the associated role or privilege

information. Once each quarter, a report is generated and emailed to each manager showing who has access to their application and what privileges are held. Before the user tracking application was written, the requirements for this application were carefully outlined in the company's security plan. Once the requirements were defined, the creation of the user tracking application became a simple task of implementing the requirements as specified. By basing the application on the security plan details, the finished user tracking application was both easy to create and completely in sync with the security plan.

Sample Security Plan Checklist

The following checklist is provided as an aid to ensure that you've identified and addressed all of the necessary areas of interest to your company. The checklist is designed to be a guide for you and your team to ensure that topics that need to be included in your security plan will not be overlooked.

Have You	Yes/No
Identified all of the key players?	
Obtained management buy-in (at all levels)?	
Collected all applicable system and database information?	
Identified the specific types of accounts required for each system—both operating system and database?	
Determined who will have authority to approve accounts?	
Determined who will create/delete/manage accounts?	
Determined a user tracking method and implementation?	
Decided how account approval will be performed: email, web site, hard-copy form, etc.?	
Identified all affected applications on each system?	
Identified a username and password structure?	
Determined what constitutes a security breach and the appropriate penalty for each breach?	
Identified all sensitive data on the system and created methods to protect that data?	
Determined what forms of monitoring will be used?	
Determined the forms of backup which will be used?	
Created recovery procedures to be followed?	
Determined the required availability for the database?	
Established standards for views and roles?	