



**Oracle Unbreakable?
Part I: Ten Hot Security Spots
in Oracle Applications 11i**
March 18, 2002

Presented by:

Martin Schlafer
Answerthink, Inc.

Michael Ackerman
Answerthink, Inc.

Introduction

This paper focuses on ten out of the box security vulnerabilities that exist within the Core Apps and the Self- Service applications. These vulnerabilities can be found in a wide range of areas from Oracle Alerts, Hosted Concurrent Programs, to Agents and Procedure Calls, as well as in the web security functionality and the work flow manager.

As we explore these various areas, we will review each vulnerability and help determine the impact to organizations, as well as provide fixes and recommendations to address these security issues.

Overview

At the Fall 2001 Comdex, Larry Ellison proclaimed that Oracle is “unbreakable”. He has since clarified what that statement meant but the fact remains that while Oracle applications are some of the best in the market, the fact exists there are a significant number of security vulnerabilities within the standard release.

Oracle Alerts Manager

Overview: Oracle Alerts allows businesses to monitor the state of the information in the database and notify individuals when a change occurs.

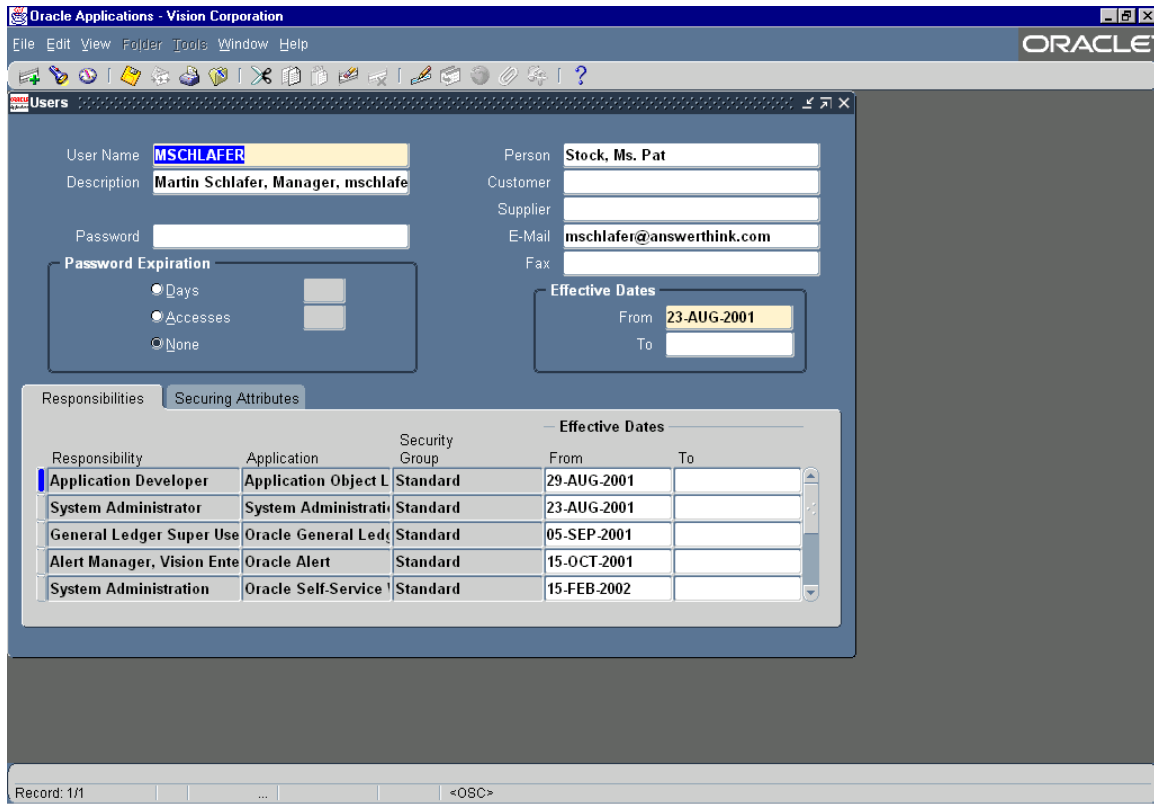
There are two types of alerts:

- Event driven alerts will notify a user when a specific action takes place within the database or applications. These alerts can also be configured to kick off other processes when they are triggered such as sending an email, running a concurrent program, or executing a SLQ script.
- Periodic alerts run as schedule times and will look for changes in the data that meet its conditions and then notify the user that the specified criteria have been met. Again as with Event alerts periodic alters can also be configured to launch a separate process.

Risk: High. APPS database objects and APPLMGR Unix files are compromised.

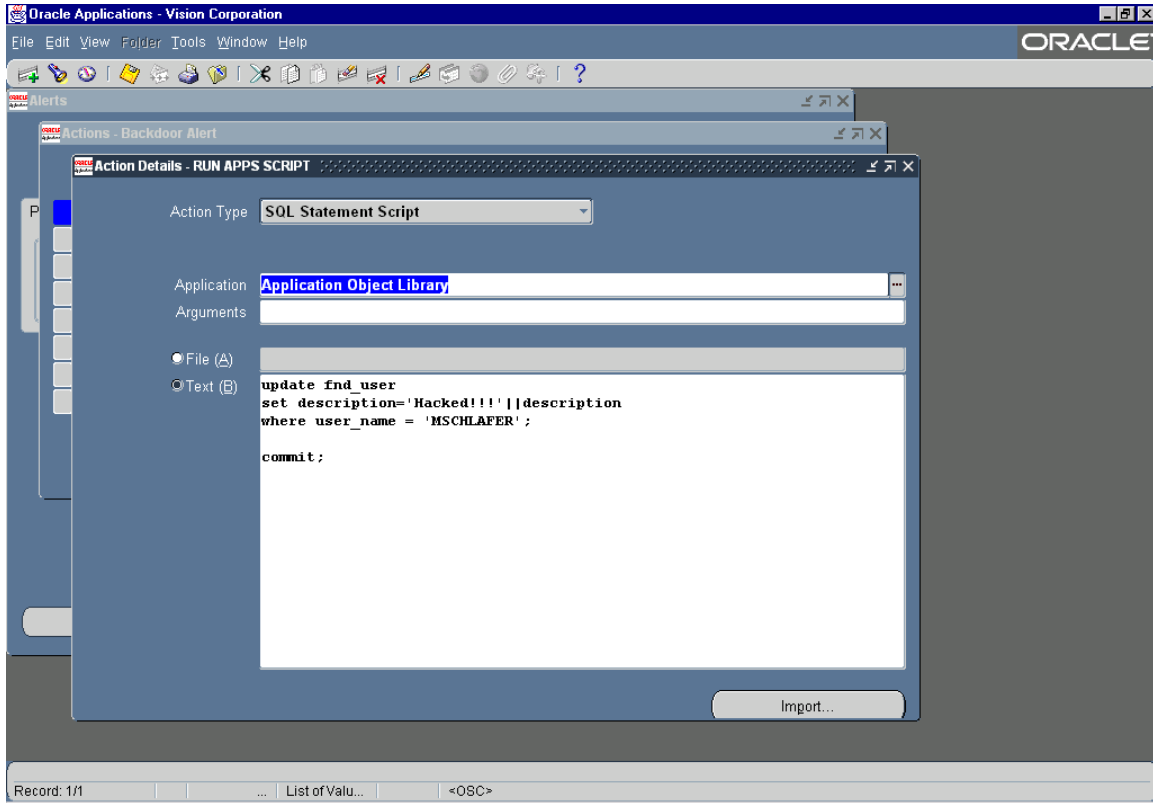
Vulnerability (1): This application module allows for the execution of an arbitrary UNIX command line as if the user was logged in using the APPLMGR account.

Vulnerability (2): It also grants the ability to run illicit SQL statements as the APPS account. Both of these actions can be taken without having to supply a password and without having direct access to the database and APPS account or UNIX and APPLMGR account.



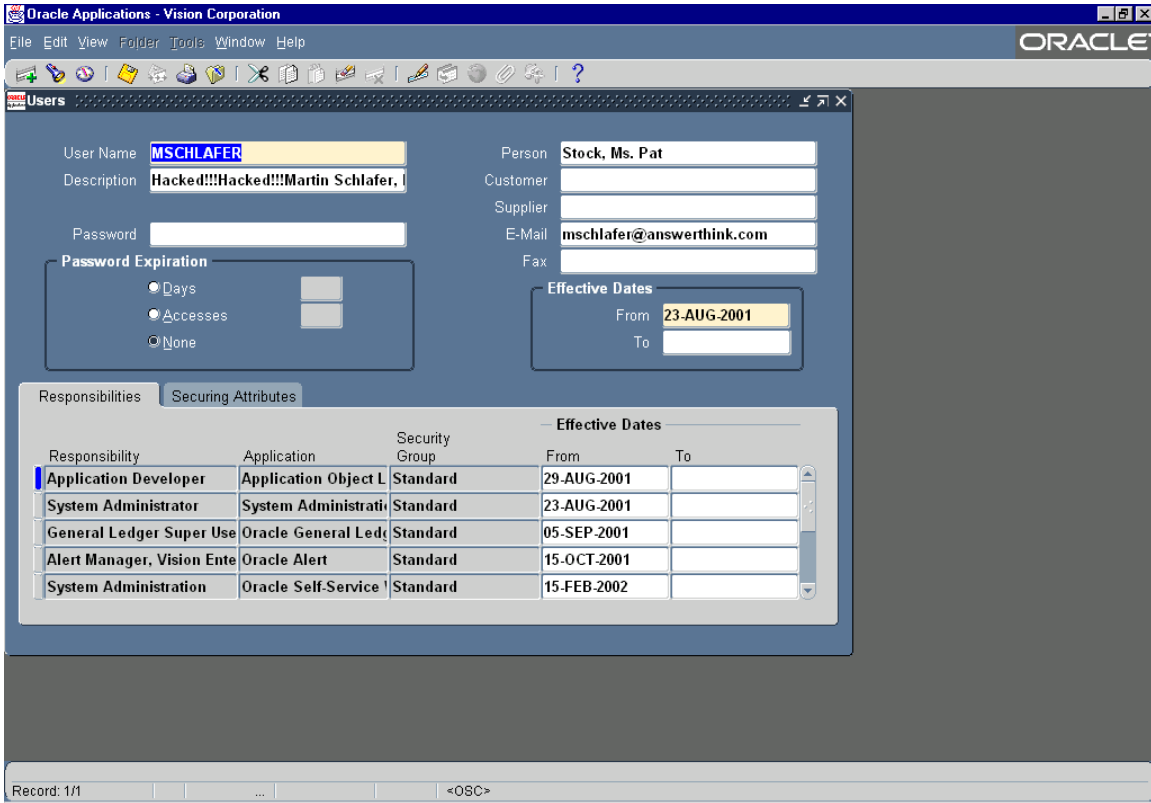
The following picture shows the user information prior to being hacked by the alerts code.

This screen shot shows the code that was used to modify the users record.



This same method could be used to add a new user, or add a new responsibility to an existing user and not leave an audit trail. PL/SQL stored procedures can be called directly by using the SQL execute command.

This shot shows the result. You can now see the “Hacked!!” text that was added to the user’s description.



Recommendation:

If alerts are used:

- Disallow power-users from ad-hoc development of alerts.
- Treat alerts as other custom development requiring review processes and code promotions.
- Monitor the Alert Manager responsibility as closely as the System Administrator responsibility.

If alerts are not used:

- Disable the Alert Manager responsibility.
- Monitor the Alert Manager responsibility as closely as the System Administrator responsibility.



Hosted Concurrent Programs

Overview: Concurrent programs are executables that run simultaneously with other programs allowing for the full utilization of the servers capacity. A Hosted Language Concurrent Program is a custom program that consists of compiled executables or programs that run operating system scripts.

Risk: High. APPS database account and objects and APPLMGR Unix files are compromised.

Vulnerability (3): Hosted programs are key components to integration and customizing the Oracle Applications. As part of this ability, they are passed the APPS schema password. A “Trojan Horse” program can capture and report this password. UNIX processes can also be monitored and exposing their command line arguments, which can include passwords that are passed in clear text.

```
$ ls -l $FND_TOP/bin/showme*
lrwxrwxrwx 1 vislmgr vislmgr      44 Dec  6 21:40
/d50/oracle/vislappl/fnd/11.5.0/bin/showme ->
/d50/oracle/vislappl/fnd/11.5.0/bin/fndcpe
-rwxr-xr-x 1 vislmgr vislmgr      43 Dec  6 21:36
/d50/oracle/vislappl/fnd/11.5.0/bin/showme.prog
$ cat $FND_TOP/bin/showme.prog
#!/bin/ksh

echo 'The Oracle ID is: ' $1

$
```

Fix: The solution to this vulnerability is actually documented in the Oracle Applications System Administrator’s Guide but is normally not implemented.

There are two ways to close this gap.

- The first allows for the user name and password to be passed but encrypted. To encrypt the Oracle username and password you will have to configure the concurrent manager to pass it as an environment variable. In the Execution Options field of the concurrent programs window you need to enter the word “ENCRYPT”. This will result in the concurrent manager sending the username/password in the environment variable fcp_login.
- The second option results in the concurrent manger not passing the username / password. To configure the concurrent program in this manner enter “SECURE” in the Execution Options field of the Concurrent Programs window.

Vulnerability (4): All application tops are defined by environment variables. The variable identifier is associated to an application in the field BASE PATH on application registration form. Altering this to another environment variable such as REPORTS60_TMP (evaluates to /tmp) will cause concurrent programs to be run from a different insecure location other than the secured APPL_TOP. The application registration form is not limited to System Administrator responsibility but also is available in the Application Developer responsibility.

Fix: Examine all environment variables in use by the UNIX APPLMGR account. Alter environment variables such as REPORTS60_TMP to reference a secure directory location. APPL_TMP is an example of where this has already been corrected.

Recommendation:

- Monitor the Application Developer responsibility as closely as the System Administrator responsibility.
- Monitor the application definitions and environment variables present in UNIX APPLMGR account.



Application PL/SQL Agent

Overview: As a matter of course, SQL methods and stored procedures run with the privileges of the definer not the invoker. Definer rights routines are bound to the schema in which they were created. However, using the AUTHID clause enables a stored procedure and SQL methods to execute with all the privileges of the invoker, or current user. These invoker rights routines are not bound to a particular schema and can be run by numerous users. Additionally the definer does not necessarily need to know who the user will be.

Risk: High. APPS database objects are compromised.

Vulnerability (5): The Oracle Applications PL/SQL agent permits execution of arbitrary PL/SQL code as the APPS schema. Oracle Applications provide some prevention with the OWA_CUSTOM package, but unfortunately it can be circumvented with invoker rights and adding the schema identifier in the URL.

```
CREATE OR REPLACE PACKAGE FND_WEB AUTHID CURRENT_USER AS
  PROCEDURE Ping;
END FND_WEB;

/

CREATE OR REPLACE PACKAGE BODY FND_WEB AS

PROCEDURE Ping IS
  user_id VARCHAR2(240);
BEGIN
  select user_id
  into user_id
  from fnd_user
  where user_name='MSCHLAFER';

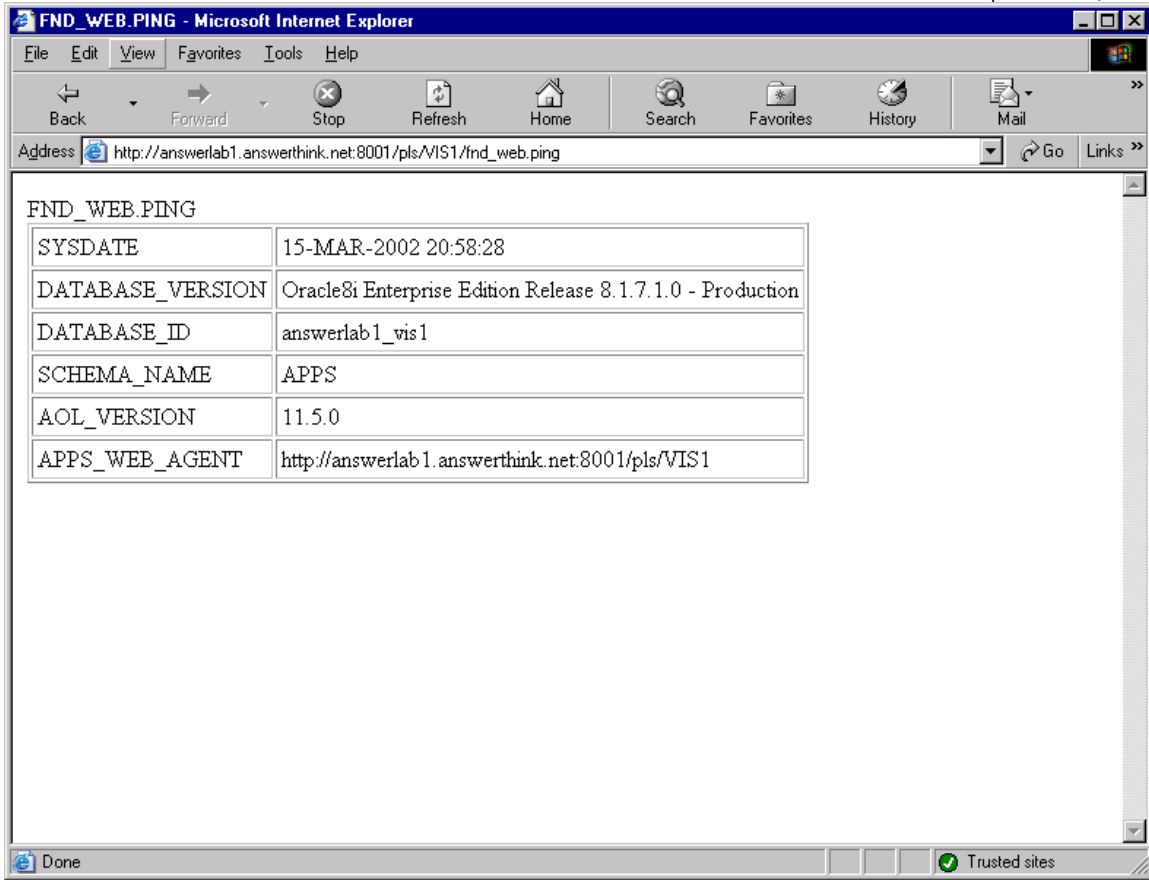
  HTP.P('The value is: '||user_id);

END Ping;

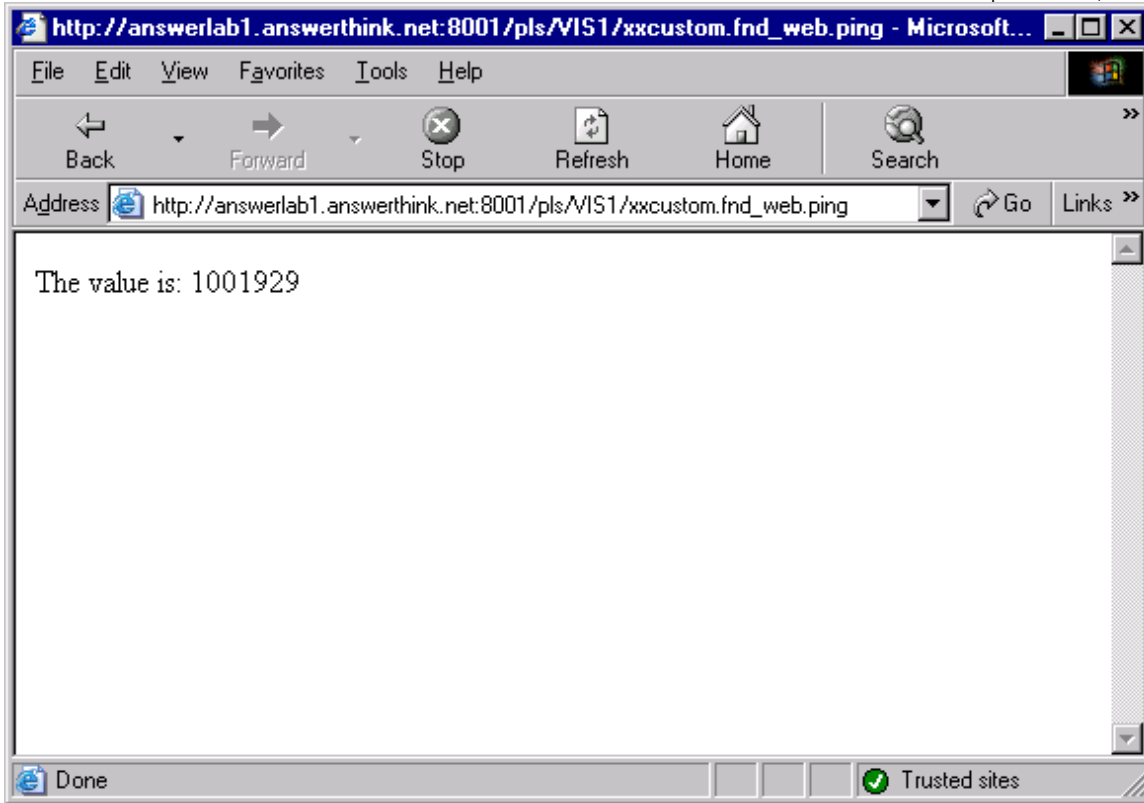
END FND_WEB;

/
```

The “Trojan Horse” PL/SQL code with invoker rights.



The standard URL to fnd_web.ping. The APPS code is executed.



The illicit call to xxcustom's fnd_web.ping defined with invoker rights. APPS executed the code and returned the user id from it's own fnd_user table!

Recommendation:

- Monitor database schemas for ability to create stored procedures.
- Monitor for stored procedures with invoker rights.

Database External Procedure Calls

Overview: External Procedure Calls provide a simple and easy method for the Oracle server and PL/SQL to callout to external programs and execute them. They can be called in a variety of ways including SQL, PL/SQL, and client side OCI. External procedures provide the foundation to extend Oracle 8i server so that it can interface with external systems, execute 3GL application code and supporting data cartridges.

Risk: High. Oracle Unix files are compromised.

Vulnerability (6): This integrates PL/SQL to compiled “C” routines. This enables PL/SQL to have a higher degree of functionality and diversity than it does stand-alone. The higher degree of functionality permits “C” code to execute arbitrary system calls as the ORACLE account. A library routine could be written that calls the C-routine “SYSTEM” in Unix without validating the string argument that is passed to the shell. A tainted string argument could be used to run a hostile program.

<p>NAME</p> <p> system - issue a shell command</p> <p>SYNOPSIS</p> <pre>#include <stdlib.h></pre> <pre>int system(const char *string);</pre> <p>DESCRIPTION</p> <p>The system() function causes string to be given to the shell as input, as if string had been typed as a command at a terminal. The invoker waits until the shell has completed, then returns the exit status of the shell in the format specified by waitpid(2).</p>
--

Fix: Do not setup the external procedure listener.

Recommendation: If external procedure calls are necessary:

- Monitor database schemas for the ability to create library objects.
- Monitor database stored procedures for calls to external C routines.



Application Web Security JAR File

Overview: Oracle uses java class and jar files that are copied to the java top during installation for web based security. The java code is version controlled in the apps.zip file contained in both the AU_TOP and JAVA_TOP directories. The executable jar files are generated in both the JAVA_TOP and <prod>_TOP directories

Risk: High. Application end-user accounts are compromised.

Vulnerability (7): The Java methods within can be called from the APPS schema to encrypt and decrypt end-user passwords. Reverse compilation of the security classes also discloses the complete encryption algorithm and private keys effectively rendering the current application security highly exposed.

Vulnerability (8): The algorithm and private key is common to all Oracle Applications installations of the same code base. Therefore someone could take the encrypted passwords from the secure system and decrypt it on another system (such as a Vision or Demo installation) using the known GUEST account and password.

Following is sample code calling the java security methods:

```
CREATE OR REPLACE PACKAGE XX_AOLSEC IS

    function encrypt(s in varchar2, s1 in varchar2, i in number) return
    varchar2;

    function decrypt(s in varchar2, s1 in varchar2) return varchar2;

END XX_AOLSEC;
/

CREATE OR REPLACE PACKAGE BODY XX_AOLSEC AS

function encrypt(s in varchar2, s1 in varchar2, i in number)
    return varchar2
    as language java name
'oracle.apps.fnd.security.WebSessionManagerProc.encrypt(java.lang.Strin
g,java.lang.String, int) return java.lang.String';

function decrypt(s in varchar2, s1 in varchar2)
    return varchar2
    as language java name
'oracle.apps.fnd.security.WebSessionManagerProc.decrypt(java.lang.Strin
g,java.lang.String) return java.lang.String';

END XX_AOLSEC;
/
```

Recommendation: Not much can be done here. Public routines used by different application components provide the encrypted password text including the GUEST account and the APPLSYSPUB account.

Oracle Self-Service Workflow Notifications

Overview: Oracle workflow is a graphical tool that enables organization to control the flow of information and transactions according the user-defined rules. These rules can be continuously changed as the needs of the business evolve.

Risk: High. Access to expense reports, requisitions, etc. exposed to approval by unauthorized users.

Vulnerability (9): The email notifications bypass the application sign-on screen. They contain a URL that has the notification ID and a key. The key can be spoofed or cracked allowing for unauthorized approvals. Also failure to authenticate named users means that a compromised email account can also be used to gain unauthorized access by less sophisticated means.

Fix: Alter the seeded approval workflow notifications to redirect users to the sign-on URL instead of “trusting” the workflow notification that passes users directly to the approval screen for the particular self-service web application.

Oracle Seeded Passwords

Overview: Oracle Applications installs with seeded passwords at the application level and the database. Often DBAs and system administrators change the passwords, but forget the non-application schemas.

Risk: High. Unauthorized access to application and database accounts.

Vulnerability (10): Failure to alter the seeded passwords allows anyone access to the application environment. The degree of compromise depends on which account is accessed. This is most common security weakness, but the easiest to correct.

The following list of schemas has the DBA database role in a fresh 11i Prod installation:
CSMIG, CTXSYS, SYS, SYSTEM

The following list of schemas has the DBA database role in a fresh 11i Vision installation:
ADSEUL1153_US, APPLSYS, APPS, CTXSYS, EU, OSM, SYS, SYSTEM, WEBDB (Additionally, the MDSYS schema has a majority of the system privileges granted directly.)

Fix: Alter the seeded passwords during installation. With FNDCPASS, alteration of the massive amount of product schemas is not as complicated as it once was.

Recommendation:

- Ensure that the application configuration files with the APPS password are adequately protected.
- Ensure that the administration log files (located under \$APPL_TOP/admin) are adequately protected.
- Do not share APPS account outside application DBA administration.

Background

Martin Schlafer and Michael Ackerman are managers in the Oracle practice at Answerthink.

Mr. Schlafer is a manager of Technology for Oracle Solutions. He is responsible for the development and implementation of world class architectures incorporating Oracle Applications technologies and supporting software. He has over eight years of experience in the design, development, implementation, and support of information technology architectures and systems in a variety of hardware environments. He has spent the last five years concentrating exclusively on the delivery of Oracle Applications (R10.6 SC, R10.7 SC and NCA, R11, and R11i). His experience spans several operating systems and platforms including IBM RS6000 – AIX 4.1.4, HP – HPUX 11, Sun – Solaris 8, and Windows NT 4.0.

Mr. Ackerman has 6 years of experience as a project manager, 3 years as an application architect, 2 years in client server development, 2 1/2 years in business practice improvement and 6 months in managing corporate training and education product development.

Answerthink (NASDAQ: ANSR, www.answerthink.com) is a leading provider of technology-enabled business transformation solutions. We address our clients' strategic business needs by offering a wide range of integrated services and solutions including: benchmarking and business strategy, business application and technology integration.

Founded in 1997, Answerthink has more than 1,100 associates based in 17 cities around the world. Answerthink is a Certified Advantage Partner with Oracle. With 130 dedicated and experienced Oracle E-Business specialists on staff we have successfully completed over 150 Oracle Applications projects in with focus in:

- Full Suite of Financial Products
- Full Suite of Human Resource Products (HR and Payroll)
- Full Suite of Supply Chain/eProcurement Products
- Full Suite of Customer Relationship Management Products
- Multi Dimensional Decision Support Systems
- Oracle Technology Solutions

Martin Schlafer (mschlafer@answerthink.com, ph: 404-849-4637)
Michael Ackerman (mackerman@answerthink.com, ph: 678-428-3969)