Oracle Applications

INTEGRIGY

April 12, 2007

Security Advisory

Oracle Applications 11i
Encrypted Password String Disclosure

OVERVIEW

An undisclosed security vulnerability exists in Oracle Applications 11i that may allow an unauthenticated, internal attacker to obtain Oracle Applications' user account encrypted password strings, which in turn can be decrypted using previously published information. An attacker can potentially obtain either any user's password or the Oracle Applications' main database account password (APPS). The attacker must have direct SQL*Net access to the database (e.g., SQL*Plus) and to exploit the vulnerability neither of the Oracle Applications security features "Managed SQL*Net Access" and "Server Security" can be enabled.

In a majority of Oracle Applications implementations, neither "Managed SQL*Net Access" nor "Server Security" are enabled. "Managed SQL*Net Access" is enabled by default beginning with 11.5.10, although, it is commonly disabled due to the complexity of managing permitted hosts and the limitations in only allowing a small number of hosts direct access to the database. "Sever Security" is not enabled by default in any version of Oracle Applications and seldom is enabled as the purpose and security benefits of this feature are poorly understood. **All Oracle Applications implementations should enable at least "Server Security" and preferably also enable "Managed SQL*Net Access".**

The underlying issue is that Oracle Applications passwords can be easily decrypted using methods previously published. [1] There are a number of ways an attacker (most likely an insider) may obtain encrypted password strings, including through ad-hoc query access, from cloned instances like development, or through SQL injection vulnerabilities in the application or standard database packages. This advisory relates to an additional method of obtaining encrypted passwords strings through exploitation of a specific undisclosed security vulnerability.

ORACLE APPLICATIONS PASSWORD DECRYPTION

Oracle Applications 11i is vulnerable to a significant security weakness in the encryption of passwords within the application where an insider may be able to obtain application account passwords or the APPS database account password. The fundamental issue is that the Oracle Applications 11i application account passwords are stored in the database encrypted using the APPS database password as the encryption key rather than using a strong, one-way hash algorithm. In order to provide access to the APPS database password upon login and for other processes, the

APPS password is stored encrypted in the database for each application account using the account username and password as the encryption key.

The method to decrypt the application passwords and the APPS database account password has been published along with easy to execute SQL commands. [2,3,4,5] The passwords can be decrypted by means of any access to the encrypted passwords strings usually through ad-hoc query access or a less secure cloned instance where the passwords have not been change.

More information on the password decryption issue can be found in Integrigy's January 2007 release "Oracle Applications 11i Password Decryption".

RISK MITIGATION STEPS

1. SET SERVER SECURITY TO SECURE [CRITICAL]

Oracle Applications Server Security when set to SECURE requires servers connecting to the application to provide a secure server ID. This is only used when actually logging into Oracle Applications through a client application (e.g., ADI or Discoverer) using SQL*Net and is not related to database authentication. By setting Server Security to SECURE may prevent an attacker from obtaining the encrypted password strings under certain circumstances. See the section "Administering Server Security" of the *Oracle Applications 11.5.10 System Administrator's Guide – Configuration* manual for detailed instructions on setting up Server Security. Server Security was introduced in the minipack FND.D (1932070) and is available in 11.5.7 and onwards. Enabling Server Security will prevent an unauthorized attacker from obtaining encrypted password strings by exploiting the security vulnerability referenced in this advisory assuming the attacker cannot obtain a Server ID key through other means.

If Server Security is enabled, then client applications such as Application Desktop Integrator (ADI) and Discoverer require a Server ID key in order to connect to the application. The Server ID key should be treated as sensitive information and should not be posted on internal websites nor documented in ADI installation notes provided to users. If ADI or other client applications are being used with Server Security, it may be possible for these users to obtain encrypted user passwords using the Server ID key provided to them. This is especially problematic in that these users have a valid login for the application, which is one of the essential components in decrypting the APPS password and other users' passwords when the GUEST password is not known or has been changed. With Sarbanes-Oxley (SOX) and Payment Card Industry (PCI) compliance, this possibility may be unacceptable and will require the deployment of these client applications on a central server (see step #3).

2. IMPLEMENT MANAGED SQL*NET ACCESS [RECOMMENDED]

11.5.10 introduced a new security feature called Managed SQL*Net Access. Managed SQL*Net Access limits the hosts that can connect to the database server using SQL*Net by implementing

Oracle TNS Listener valid node checking. Valid node checking is a list of IP addresses or host names that are permitted to connect to the database server. See Metalink Note ID 291897.1 for more information on configuring this feature. Unfortunately, it is very difficult to implement this feature since a large number of hosts often require access to the database server for interfaces, management, and client applications such as ADI or Discover.

3. Deploy Oracle Client Applications Using Citrix or Terminal Server [Recommended]

All Oracle client applications such as ADI or Discoverer can be deployed from central servers using products like Citrix MetaFrame or Microsoft Terminal Server. Deploying client applications from a central server in combination with enabling Server Security and Managed SQL*Net access will mitigate the risk from this security vulnerability as long as ad-hoc query access is not also permitted from this central server or through other means.

See Metalink Note ID 277535.1 "E-Business Suite Recommended Set Up for Client/Server Products" for more information on configuring client applications to work Citrix MetaFrame or Microsoft Terminal Server.

Additional Steps

The Oracle Applications encrypted passwords must be protected in order to prevent decryption. The goal is to limit access to the FND_USER table and the encrypted passwords, just as should be done with the DBA_USERS view to prevent brute-forcing of the database account passwords.

General

1. Verify APPLSYSPUB does not have access to FND_USER_VIEW [CRITICAL]

Verify APPLSYSPUB and PUBLIC do not have SELECT privileges on the view APPS.FND_USER_VIEW. This is especially an issue with instances that were upgraded from 11.5.6 and prior. FND_USER_VIEW shows all application accounts and the ENCRYPTED_FOUNDATION_PASSWORD. Prior to 11.5.7, APPLSYSPUB may have been granted SELECT privileges on this view to support ADI. This view is not required by APPLSYSPUB, except for old, desupported versions of ADI.

2. Change passwords for all database accounts [Critical]

Change the passwords for every database account including all 250+ Oracle Applications schemas. Even though a module is not being used, the database account password must be changed. Use the FNDCPASS utility to change all the database passwords on a periodic basis. In 11.5.10 RUP3, FNDCPASS includes a new option (ALLORACLE) to change all the schema passwords in a single FNDCPASS call (see Metalink Note ID 398942.1).

3. Change passwords for all seeded Oracle Applications accounts [recommended]

Change the passwords for all Oracle Applications 11i seeded accounts (SYSADMIN, WIZARD, APPSMGR, etc.) even though these accounts may be already be disabled. At the same time, make sure all accounts except for SYSADMIN and GUEST are disabled (note a few accounts may be required by a specific module). See Metalink Note ID 189367.1 for the most up to date list of seeded user accounts. For years clients have questioned why we recommend always changing the seeded account passwords even though the accounts may be disabled – the ability to decrypt application passwords is the reason why.

4. Create all new application accounts with strong passwords [Recommended]

Create all new user accounts with unique and strong passwords. In 11.5.10, User Management (UMX) can be used to securely create new users with strong passwords.

5. Change GUEST account password

The password for the GUEST account should be changed from the default of ORACLE or GUEST. Follow the solution in Metalink Note ID 396537.1 for details on changing the password and check that the password was also changed in the System Profile Option "Guest User Password". Changing the GUEST account password can be problematic and the procedure should be thoroughly verified in a test instance prior to changing the password in production.

SQL Access

6. Limit access to FND_USER, FND_ORACLE_USERID, and FND_NODES [Recommended]

Limit access to the APPLSYS.FND_USER, APPLSYS.FND_ORACLE_USERID, and APPLSYS.FND_NODES tables by all non-DBA accounts including any query-only accounts. Often an APPSREAD or similar database account is created for support purposes or end-user ad-hoc query use. These accounts tend to be created with SELECT ANY TABLE system privilege, which allows access to FND_USER. Instead, all non-DBA accounts should be created with a limited set of database privileges for only those tables absolutely required for the business function.

Unfortunately, the FND_USER table is fundamentally required by many reporting and ad-hoc queries, thus it is difficult to directly exclude this table from such database accounts. Also, over 500 standard Oracle Applications views are dependent on FND_USER. A careful review of ad-hoc query privileges should be performed to determine the exact business requirements and privileges required.

Query accounts should not normally require access to FND_ORACLE_USERID, therefore, this table should be easy to exclude from such database accounts.

The FND_NODES table contains the "Server Security" SERVER_IDs, which should be protected. Access to a SERVER_ID may allow an attacker to obtain encrypted password strings.

CLONED DATABASES

All user and database passwords should be immediately changed in all cloned databases to prevent decryption of the production passwords.

7. CHANGE ALL APPLICATION ACCOUNT PASSWORDS DURING CLONING [CRITICAL]

As part of the cloning process, change all application account passwords to a random string using a PL/SQL script that calls FND_USER_PKG.CHANGEPASSWORD. An operational issue then exists in that users of the cloned instance will need to obtain the new password. One solution is to use the "Reset Password Functionality" in 11.5.10 and UMX.H. Users needing access will then have to reset their password after each clone of a development or test database. See Metalink Note ID 399766.1 for more information on the UMX Reset Password Functionality.

8. CHANGE ALL DATABASE ACCOUNT PASSWORDS DURING CLONING [CRITICAL]

All database account passwords should be changed as part of the cloning process (with the exception of APPLSYSPUB). Even though a module is not being used, the database account password must be changed. In 11.5.10 RUP3, FNDCPASS includes a new option (ALLORACLE) to change all the schema passwords in a single FNDCPASS call (see Metalink Note ID 398942.1). Also, all standard database account passwords (CTXSYS, DBSNMP, etc.) should also be changed as part of each clone.

9. CHANGE THE GUEST ACCOUNT PASSWORD DURING CLONING

The GUEST password requires additional steps in order to change including updating the password in the AutoConfig XML file. As part of the cloning process, follow the steps in Metalink Note ID 396537.1 to change the GUEST password.

VULNERABILITY INFORMATION DISCLOSURE

In accordance with Integrigy's vulnerability disclosure guidelines, Integrigy will not publish any details or further information regarding the security vulnerability referenced in this advisory unless such information is made public in the future by Oracle or a third-party. The only reasonable and supported risk mitigation steps are those actions described in this advisory. These steps are fully supported and recommended by Oracle as the appropriate actions related to this vulnerability. Oracle has been notified of this vulnerability and recommends the actions described in this advisory until a permanent solution to the password decryption issue can be implemented. Oracle's

response to our releasing this advisory was "we have no problem with you issuing an advisory explaining good practice for the topics you mention in your email."

REFERENCES

1. Integrigy Corporation, "Oracle Applications 11i Password Decryption", 9 January 2007, http://www.integrigy.com/security-resources/whitepapers/Integrigy_Oracle_Apps_Password_Issue.pdf
2. Martin Schlafer and Michael Ackerman Michael, "Oracle Unbreakable? Part I: Ten Hot Security Spots in Oracle Applications 11i", March 2002, Spring 2002 OAUG Conference, http://www.oaug.com/conferencesandeducation/papers/spring2002/SCHLAFEW.pdf
3. http://www.erp100.com/viewthread.php?action=printable&tid=1592, November 2005, (Simplified Chinese -> English Translation)
4. http://521102yz.itpub.net/post/5095/84876, May 2006, (Simplified Chinese -> English Translation)
5. Johan Louwers, "Oracle Applications Passwords Decryption Vulnerability", December 2006, http://johanlouwers.blogspot.com/2006/12/oracle-applications-passwords.html
6. Oracle Corporation, "Best Practices for Securing the E-Business Suite 3.0.4", October 2006, Metalink Note ID 189367.1, https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=189367.1 [Sections of this document were written by Integrigy Corporation]
7. Oracle Corporation, "After RUP4 Unable To Login After Password Reset Or Password Expiration", 3 December 2006, Metalink Note ID 396537.1, https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=396537.1
8. Oracle Corporation, "Reset Password Functionality FAQ", 15 November 2006, Oracle Metalink Note ID 399766.1, https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=399766.1
9. Oracle Corporation, "11.5.10 New Features : Managed SQL*Net Access from Hosts", 21 January 2005, Metalink Note ID 291897.1, https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=291897.1
10. Oracle Corporation, "E-Business Suite Recommended Set Up for Client/Server Products", 24 February 2006, Metalink Note ID 277535.1, https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=277535.1

HISTORY

April 12, 2007 – Initial Version

About Integrigy

Integrigy Corporation is a leader in application security for large enterprise, mission critical applications. Our application vulnerability assessment tool, AppSentry, assists companies in securing their largest and most important applications. AppDefend is an intrusion prevention system for Oracle Applications and blocks common types of attacks against application servers. Integrigy Consulting offers security assessment services for leading ERP and CRM applications.

Integrigy Corporation
P.O. Box 81545
Chicago, Illinois 60602 USA
888/542-4802
www.integrigy.com

Authors: Stephen Kost and Jack Kanter

If you have any questions, comments or suggestions regarding this document, please send them via e-mail to alerts@integrigy.com.