INTEGRIGY

TECHNICAL WHITE PAPER

Oracle Business Intelligence Enterprise Edition (OBIEE): Security Examined

MARCH 2014

OBIEE: Security Examined

Version 1.0.0 – March 2014

Authors: Mike Miller, CISSP-ISSMP

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

Table of Contents

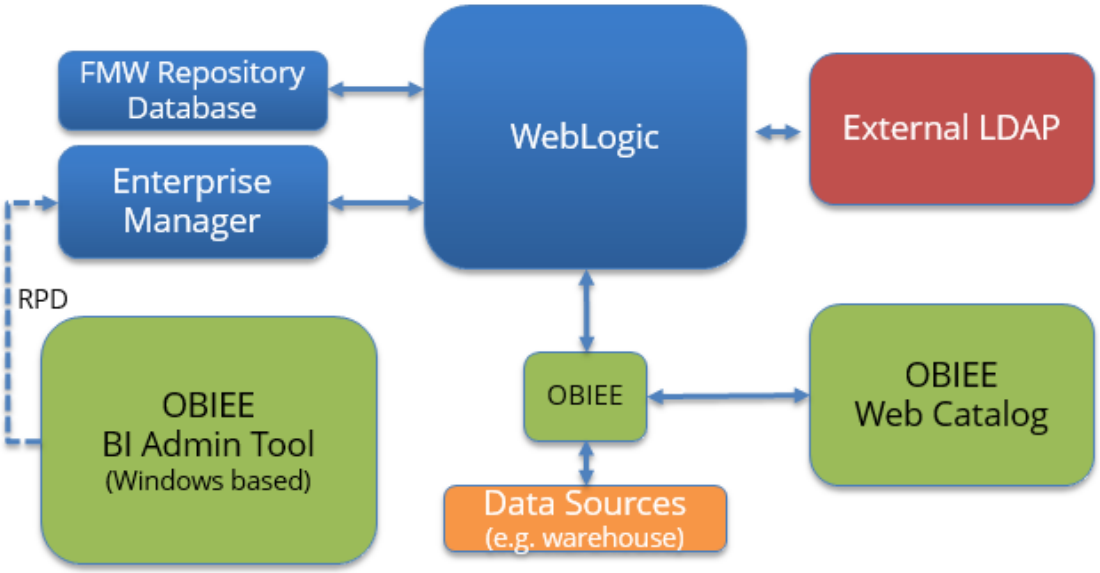
INTRODUCTION

Oracle Business Intelligence Enterprise Edition (OBIEE) 11g is a powerful tool for accessing data, however, this power means OBIEE security is imperative in order to protect the data.

This paper discusses the security features of OBIEE and uses the perspective of a manager or security professional. The intent is to present the key concepts and decisions that need to be made when considering how to secure OBIEE. As such this paper assumes a general familiarity with OBIEE and does not seek to describe OBIEE functionality in detail. A great many other sources of information can be found on OBIEE architecture, functionality and best practices for implementation.

As OBIEE is part of the Oracle Fusion Middleware product suite, a review of Oracle WebLogic's security features will be followed by a discussion of security features of each layer of the OBIEE technology stack. Figure 1 graphically depicts the architectural components of OBIEE. The sizes of the boxes are relative to the importance of OBIEE security.

Figure 1 - **OBIEE Security**

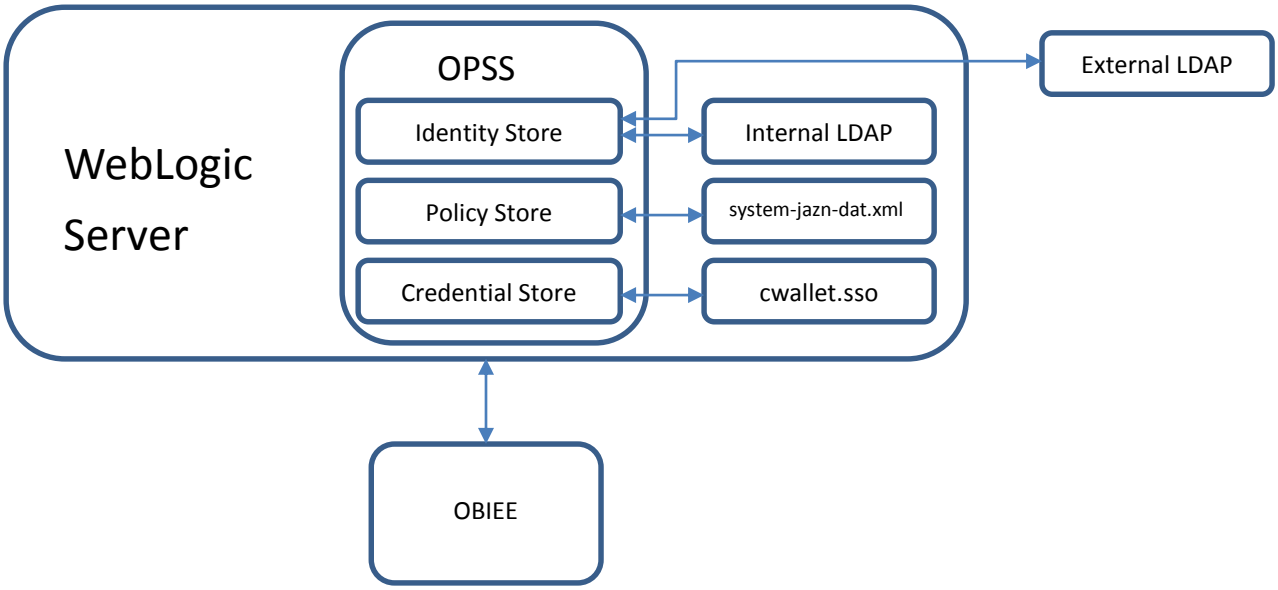


Audience and How to Read This Paper

The intended audience is IT security and internal audit professionals or those seeking a technical review of OBIEE's security features. A working technical knowledge of OBIEE, WebLogic and Oracle Databases is recommended. The review of WebLogic's security features may be read later if the reader needs only a quick review of OBIEE key security features and concepts.

WEBLOGIC

To discuss OBIEE security it makes sense to first look at WebLogic. As a Fusion Middleware 11g product, OBIEE 11g uses Oracle WebLogic for centralized common services, including a common security model. WebLogic itself is a scalable, enterprise-ready Java Platform, Enterprise Edition (Java EE) application server. The WebLogic Server infrastructure is based on a Service Oriented Architecture (SOA) which allows it to be the foundation for many different types of applications. Oracle Corporation describes the WebLogic Server security architecture as "providing a comprehensive, flexible security infrastructure designed to address the security challenges of making applications available on the Web."



SECURITY BASICS

To start the discussion of WebLogic security, WebLogic needs to be installed and run per security best practices. Passwords need to be safeguarded and well managed. Ensure that all relevant WebLogic passwords are complex and expiring periodically. Look for, and do not use, accounts used in sample applications such as weblogic/welcome1.

Production environments need to be physically separated from test and development environments. Development should not be done in a production environment. Servers on which WebLogic is to be installed need to be appropriately hardened, especially if supporting Internet connections, and sample and/or demonstration software should not be installed on production servers.

Network

Before installing WebLogic, have an expert review network services to ensure that a malicious attacker cannot access the operating system or system-level commands. Use a DMZ if exposing OBIEE functionality on the Internet. Also use SSL to protect client communications and configure all Fusion Middleware Applications to use SSL. The OHS web services commonly use 4443 (not 7777) on the firewall and 9704 internally for OBIEE reporting services.

Many OBIEE implementations take advantage of mobile access outside the firewall (e.g. iPad or tablets). The OBIEE mobile interface uses the same authentication within WebLogic. No additional authentication configurations are required.

OPERATING SYSTEM

The following reviews a few of the more important security issues for an operating system supporting WebLogic.

WebLogic User

WebLogic should be installed and started by a single operating user purposely created to support WebLogic. If possible avoid choosing an obvious name for this user. Do not use demo or sample user accounts and passwords.

Starting WebLogic

As a security best practice, WebLogic must not be run as a privileged user or as root. For UNIX installations this requires additional steps to be taken because with UNIX, only processes that run under a privileged user account (usually root) can bind to ports lower than 1024. Because WebLogic, as an Application Server is a long running process that needs to communicate on lower ports such as 80 or 443, there are two commonly used options:

- Start WebLogic under the privileged user account, bind to the privileged ports, and then change its user ID to a non-privileged account
- Start WebLogic using a non-privileged account and configure the firewall to use Network Address Translation (NAT) software to map protected ports to unprotected ones

File Permissions

Only install WebLogic on a host that can prevent unauthorized access to protected resources. For example, on a Windows computer, use only NTFS. Access to the WebLogic configuration files must be carefully restricted to the WebLogic administrators, ideally on an as needed basis only. No other operating system user should have read, write, or execute access to WebLogic Server product files or domain files.

When performing a security assessment for WebLogic, or any Fusion Middleware product such as OBIEE, the operating system accounts that WebLogic' is being run by must be verified along with a review of the WebLogic administration account. In addition, the file permissions for the following directories need to be reviewed to determine whether or not operating system access to the following directories is appropriately restricted:

- **Middleware Home directory** - this is the top-level directory for all Oracle Fusion Middleware products and is created when WebLogic Server is installed. By default, this directory is named Oracle/Middleware.
- **WebLogic Server product installation directory** – This contains all the WebLogic Server software components installed on the system, including program files. By default, this directory is a subdirectory of the Middleware Home and is named Oracle/Middleware/wlserver_10.3
- **WebLogic domain directories** - These contain the configuration files, security files, log files, Java EE applications, and other Java EE resources for a single WebLogic domain. By default, a domain is a

subdirectory of Middleware Home (for example, Oracle/Middleware/user_projects/domains/domain1). If multiple domains exist on a WebLogic Server host computer, each domain directory must be protected.

- **Persistence Store** - WebLogic's persistent store is a built-in solution for subsystems and services that require persistence. For example, it can store persistent JMS messages or temporarily store messages sent using the Store-and-Forward feature. The default persistent store is located in the data\store\default directory inside the servername subdirectory of a domain's root directory. During a security assessment, verify that operating system file access permissions appropriately restrict access to this directory.

In addition, the security of the following directories is also of importance:

- The security LDAP database, which by default is in \domains\domain-name\servers\server-name\data\ldap\ldapfiles
- The directory and filename location of the private keystore
- The directory and filename location of a Root Certificate Authority (CA) keystore

For the above, at a minimum, consider using "umask 066", which denies read and write permission to Group and Others. The risk of failing to properly restrict file system access is that knowledgeable operating system users may be able to bypass WebLogic server security.

Security Realms

As a Fusion Middleware 11g product, OBIEE 11g uses Oracle WebLogic for centralized common services, including a common security model. WebLogic Security Realms define the security configurations required to protect the application(s) deployed within WebLogic and consist of definitions of users, groups, security roles and polices.

A security assessment for OBIEE needs to closely examine each Security Realm for recommended configurations and alignment to policy and compliance requirements. Settings such as time-out variables and logon attempts are set in the Security Realm. The installation of OBIEE creates a default Security Realm. Whether or not the default realm can be used or a new one created is an important implementation decision.

A WebLogic server can have multiple security realms, but only one realm can be active for a given domain (a domain is one or more servers or cluster of servers). As OBIEE can only have one repository on a single WebLogic server, the realm governing OBIEE must also be carefully identified during any security assessment.

If at all possible, Integrigy Corporation recommends using the default realm as a baseline to configure a new Realm for OBIEE. Integrigy Corporation highly recommends that each security realm attribute be thoroughly understood. For example, Security Realms can be configured to establish trust between two WebLogic servers. Enabling cross domain security can potentially open a WebLogic sever to man-in-the-middle attacks.

ORACLE PLATFORM SECURITY SERVICES (OPSS)

To implement Security Realm configurations, all Fusion Middleware applications use a security abstraction layer within WebLogic called the Oracle Platform Security Services (OPSS). OPSS is not the same as WebLogic security. WebLogic consumes OPSS services and frameworks (for example authentication). OPSS provides three key services:

- An **Identity Store**, to define and authenticate users that by default is set to use the embedded WebLogic Server LDAP server
- A **Credential Store**, to hold the usernames, passwords and other credentials that system services require. For example, for JDBC data sources
- A **Policy Store**, containing details of user groups and application roles, application policies and permissions. The policy store is used to authorize users after they are authenticated.

OPSS Identity Store

The use of OPSS identity store means that OBIEE user accounts are defined with WebLogic or optionally a third party LDAP server configured for WebLogic to use. The advantage of using a corporate LDAP server is that user accounts potentially do not need to be duplicated in WebLogic.

WebLogic does not support or certify a particular external LDAP server. Any v2 or v3 compliant LDAP server should work with WebLogic. WebLogic offers the following authentication options:

- Default embedded "built-in" LDAP server
- External LDAP server - (iPlanet, Active Directory, Open LDAP, Novel, Generic (v2/v3)
- External database through WebLogic JDBC data source
- Simple text file – this is not recommended for a production environment

Each authentication provider holds a LoginModule that performs the actual authentication, and if the realm uses multiple authentication providers, multiple definitions of the same user exist. While this allows for Federated lookups of user profile information across multiple identity stores, it does present a potential security risk. Integrigy Corporation recommends a go-live check, and also during any security assessment, to audit and reconcile all active authentication providers, groups and users. Especially if the OBIEE environment has been upgraded, the risk of legacy authentication providers might allow users to login with an obsolete or no password.

OPSS Credential Store

The credential store securely stores the usernames, passwords and other credentials that system services require. This includes credentials that are both user supplied and system generated. In 11g, the credential store is managed using the FMW Enterprise Manager and is stored in an Oracle Wallet (**cwallet.sso file)**.

The credential store stores passwords for deployed RPDs, BI Publisher data sources and BISystem users. Instead of using an Oracle Wallet, there is also the option for the credential store to use LDAP, but only Oracle Internet Directory is supported right now.

OPSS Policy Store

The domain policy store is the repository of system and application-specific policies. Think of these polices as the "who can do what" lists. The Identity store defines users, the credential store holds passwords and the policy store defines where the users can go and do. For each WebLogic domain there is one store that defines all policies and credentials of all applications deployed in the domain. During a security assessment the policy store should be reviewed. It is specified in the file jazn-data.xml.

WEBLOGIC SCRIPTING TOOL

The WebLogic Scripting Tool (WLST) is a command-line scripting environment that is used to create, manage, and monitor WebLogic. It is based on the Java scripting interpreter, Jython, version 2.2.1. In addition to supporting standard Jython features such as local variables, conditional variables, and flow control statements, WLST provides a set of scripting functions (commands) that are specific to WebLogic Server.

From a security risk perspective, consider WLST analogous to how DBAs use SQL to manage an Oracle database. Who is using WLST and how they are using it needs to be carefully reviewed as part of any WebLogic security assessment.

WLST uses the WebLogic Security Framework to enforce the same security rules as when using the WebLogic user interface. WLST scripts, similar to SQL scripts, are created and edited using any text editor and the operating system user running a WLST script can easily be different than the user referenced in the script. WLST scripts can be run in either on or offline mode and, aside from modifying and copying configurations, (e.g. to create a test server), they can be used to add, remove, or modify users, groups, and roles.

Securing the WLST Connection

Both Integrigy Corporation and Oracle recommend that when using WLST only connect through the administration port. The **administration port** is a special, secure port that all WebLogic Server instances in a domain can use for administration traffic.

By default, this port is not enabled, but it is recommended that administration port be enabled in production. Separating administration traffic from application traffic ensures that critical administration operations (starting and stopping servers and changing configurations) do not compete with application traffic on the same network connection.

The administration port is required to be secured using SSL. As well, by default, the demonstration certificate is used for SSL. The demo SSL certificate should not be used for production.

Writing and Reading Encrypted Configuration Values

Some attributes of a WebLogic Server configuration are encrypted to prevent unauthorized access to sensitive data. For example, JDBC data source passwords are encrypted. It is highly recommended to follow the WebLogic scripting tool documentation for specific instructions on working with encrypted configuration values however WLST is used - manually (ad-hoc), in scripts, offline and on

line. A security assessment should include a discussion, if not a review, of WLST scripts that set or manipulate encrypted values.

Running WLST Scripts

WLST scripts permit unencrypted passwords at the command line. WebLogic security policies need to address how WLST scripts should provide passwords. Storing passwords incorrectly can easily and needlessly expose passwords in scripts, on monitor screens and in logs files. When entering WLST commands that require an unencrypted password, the following precautions should be taken:

- Enter passwords only when prompted. If a password is omitted from the command line, it is subsequently prompted for when the command is executed
- For scripts that start WebLogic Server instances, create a boot identity file. The boot identity file is a text file that contains user credentials. Because the credentials are encrypted, using a boot identity file is much more secure than storing unencrypted credentials in a startup or shutdown script.
- For WLST administration scripts that require a user name and password, consider using a configuration file. This file, can be created using the WLST storeUserConfig command and contains:
    - User credentials in an encrypted form
    - A key file that WebLogic Server uses to unencrypt the credentials

AUDITING

WebLogic offers a robust suite of auditing functionality. The decision to audit a particular event can be based on specific audit criteria and/or severity levels. The records containing the audit information may be written to output repositories such as an LDAP server, database or a simple file.

Key features of the Audit Framework include:

- A uniform system for administering audits across a range of Java components, system components, and applications
- Common audit record format
- Common mechanism for audit policy configuration
- Extensive support for Java component auditing, which includes the ability to search for audit data at any application level
- Capturing authentication history/failures, authorization history, user management, and other common transaction data
- Flexible audit policies, pre-seeded audit policies, tree-like policy structure simplifies policy setup
- Prebuilt compliance reporting features
- Audit record storage - Data store (database) and files are both available. Using a data store allows the generation of reports with Oracle Business Intelligence Publisher.

KEYSTORE AND KEYS

To support SSL, including private keys, digital certificates, and trusted CA certificates WebLogic provides two types of keystores for keys and certificates:

- JKS-based keystore and truststore, the default JDK implementation of Java keystores provided by Sun Microsystems
- Oracle wallet, a container for PKCS#12-based credentials. Oracle wallets are used for Oracle Internet Directory and other products such as Oracle HTTP Server, Oracle Web Cache.

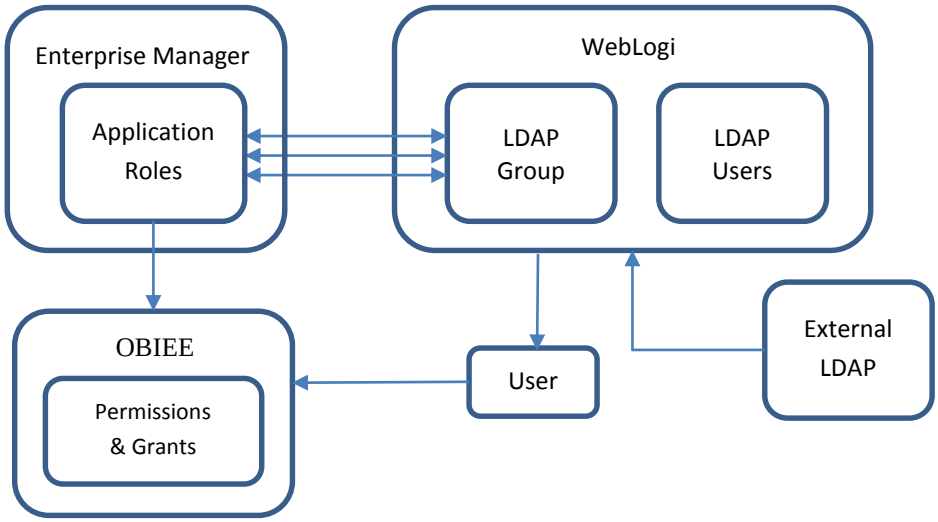
Both Integrigy Corporation and Oracle Corporation recommend using separate keystores for identity and trust. The identity keystore (private key/digital certificate pairs) and should not be allowed to risk compromising the trust keystore (trusted CA certificates) and vice versa. A further recommendation is to restrict access to the identity keystore password to as few people as possible. Certainly the number of people with access to both keystores is a security risk that should be mitigated.

Enterprise Manager and Application Roles

Application roles are new with OBIEE 11g and replace groups within OBIEE 10g. The migration of application roles out of OBIEE allows a common set of roles to be define across all Fusion Middleware products and applications.

Application roles and Application Policies are managed in Oracle Enterprise Manager - Fusion Middleware Control. This is where LDAP groups are mapped to application roles and detailed permissions are assigned to the application roles. The key concept is that LDAP groups can be assigned to both Fusion users and Fusion Application roles, LDAP users are never individually or directly assigned permissions and grants within OBIEE.

Figure 2 Application Roles and Users

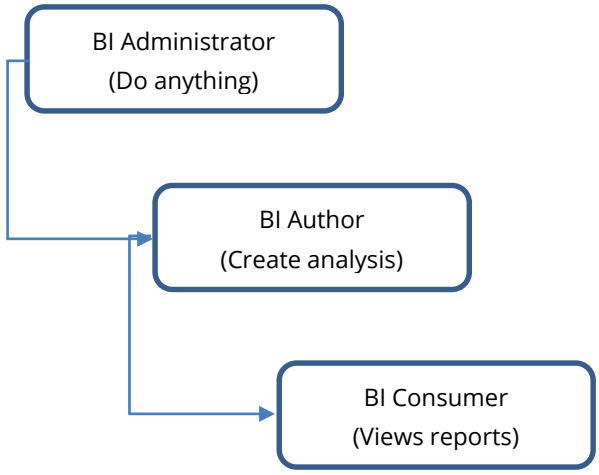


The out-of-the-box installation of OBIEE delivers three main application roles. These roles may be granted to individual users or to LDAP groups. During the implementation or at any time new roles can be created and existing roles changed.

Default OBIEE Application Roles		
Application Role	**LDAP Group***	**Description**
BIConsumer	BIConsumers	Base-level role that grants the user access to OBIEE analyses, dashboards and agents. Allows user to run or schedule existing BI Publisher reports, but not create any new ones
BIAuthor	BIAuthors	All BIConsumer rights, grants and permissions but also allows users to create new analyses, dashboards and other BI objects
BIAdministrator	BIAdministrators	All BIAuthor rights, grants and permissions (and therefore BIConsumer) as well as allows the user to administer all parts of the system, including modifying catalog permissions and privileges

*Note the naming convention difference of plural vs singular for Application Roles

Graphically the table above can be depicted as such:



Fusion Middleware Repository

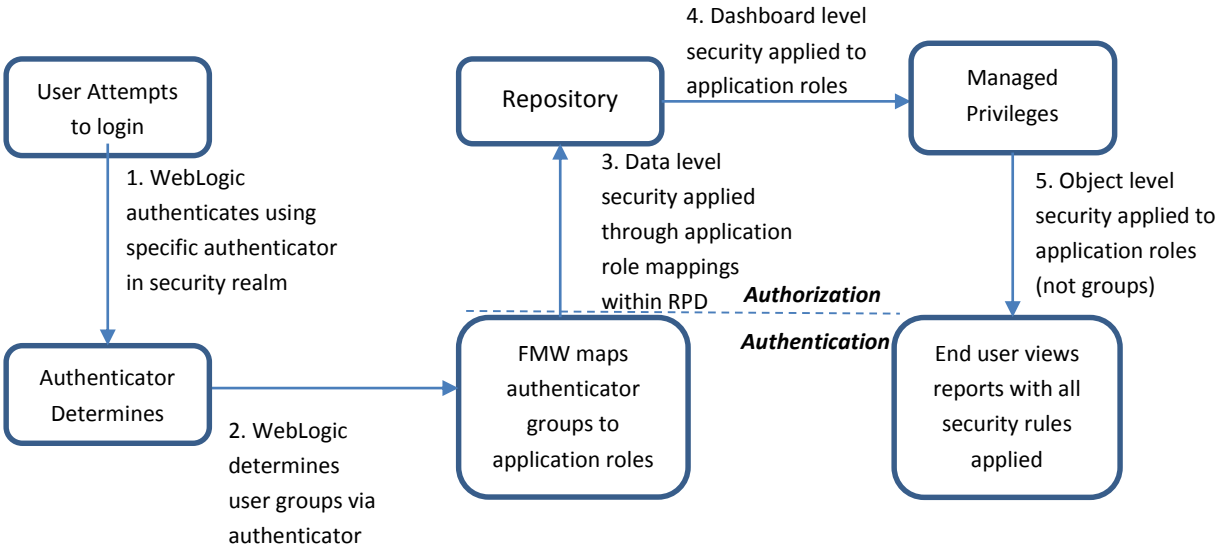
Fusion Middleware installations required a relational database. It is referred to as the repository. This repository should not be confused with the OBIEE repository (discussed later in this paper). Either an existing database or a new database can be used. The database need not be an Oracle database and several different vendors are supported - for example, SQL-Server.

The Fusion Middleware utility referred to as the 'Repository Creation Utility' is used to create a schema in the database for the purpose of storing metadata. Each installed Fusion Middleware product will have its own schema. For OBIEE, the BIPLATFORM and MDS schemas are used. These tables should not be manually accessed or edited.

During a security assessment the security of the repository database should be checked per the standard security checklist for the specific vendor.

OBIEE SECURITY

The review of OBIEE security to this point has identified how users are defined and authenticated within WebLogic, the major security concerns with WebLogic and how Fusion Application roles are defined and mapped to LDAP groups within Enterprise Manager. It is now time to discuss how OBIEE security and how authorization is accomplished.



REPOSITORY DATABASE

The OBIEE security model is almost entirely defined in the Repository Database and implemented in three ways:

- **Object-level security**: these are permissions on specific objects such as subject areas, presentation or physical tables and columns. Object level security is set in the RPD file.
- **Data-level security**: data filters to eliminate rows from result sets. Data level security is set in the RPD file.
- **Presentation Catalog security**: what reports and dashboards are available to specific users, application roles and LDAP groups. The presentation catalog defines its own security.

The OBIEE repository database, known as a RPD file because of its file extension, defines the entire OBIEE application. It contains all the metadata, security rules, database connection information and SQL used by an OBIEE application. The RPD file is password protected and the whole file is encrypted. Only the Oracle BI Administration tool can create or open RPD files and BI Administration tool runs only on Windows. To deploy an OBIEE application, the RPD file must be uploaded to Oracle Enterprise Manager. After uploading the RPD, the PRD password then must be entered into Enterprise Manager.

RPD Passwords

From a security assessment perspective, who has physical access to the RPD file and the RPD password is critical. If multiple OBIEE applications are being used, the RPD passwords should all be different. It is also

recommended that the RDP password be rotated per whatever policy governs critical database accounts and that production RPD passwords be different than non-production RPD passwords.

Once deployed through WebLogic, RPD file is located here:

For 11g:
```
ORACLE_INSTANCE/bifoundation/OracleBIServerComponent/coreapplication_obisn/repository
```

For 10g:
```
BI_ORACLE_HOME/server/repository
```

Initialization Blocks and Session Variables

Initialization blocks within the repository are used for instantiating a session when users connect. It is here where session variables are set that are used to implement many of OBIEE security rules. For example, when OBIEE is integrated with the Oracle E-Business Suite, the initialization block is where the responsibility_id and user_id variables are set in order to allow E-Business Suite responsibility security roles to be mapped to OBIEE security rules.

When performing a security assessment of OBIEE, to understand how security is being implemented within the RPD file, it will be necessary to know initialization blocks are being used and what variables they are setting. As well, of particular importance in a security assessment is to determine whether or not authentication is being done within the initialization block. Both Integrigy Corporation and Oracle Corporation recommend that authentication occur only within WebLogic.

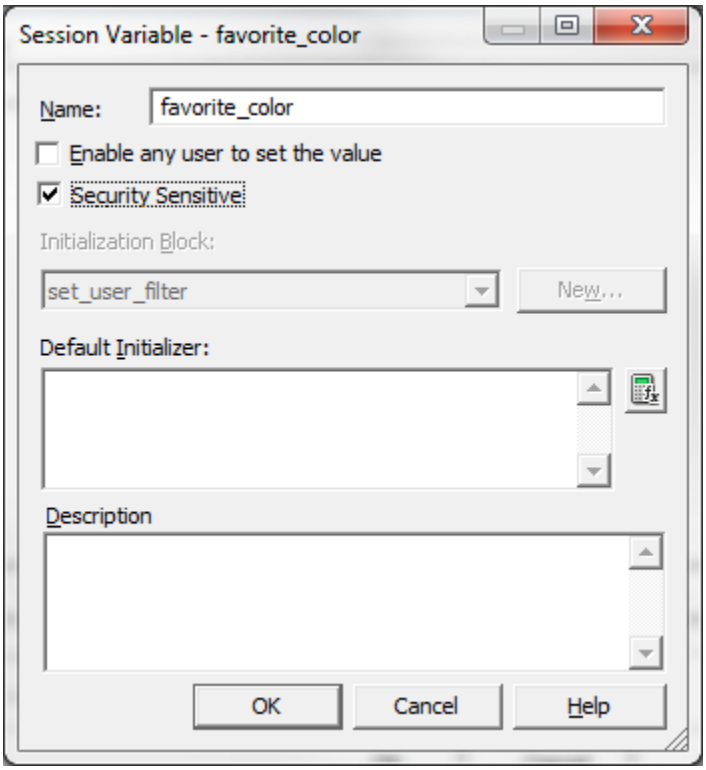
There are two types of session variables:

- **System Session variable –** These have reserved names that cannot be used for other kinds of variables, for example 'USERGUID'
- **Non System Session variable** – Defined by author of RPD file and commonly used in security filters

For session variables, there are two important security flags:

- **Enable any user to set the value** - allows session variable to be set after the initialization block has populated the value (at user login) by calling the ODBC stored procedure NQSSetSessionValue()
- **Security Sensitive** - Required when using row-level database security rules, most importantly when using Virtual Private Database (VPD). When filtering cache table matches, the Oracle BI Server looks at the parent database object of each column or table that is referenced in the logical request projection list. If the database object has the Virtual Private Database option selected, the Oracle BI Server matches a list of security-sensitive variables to each prospective cache hit. Cache hits would only occur on cache entries that included and matched all security-sensitive variables.

Security Sensitive

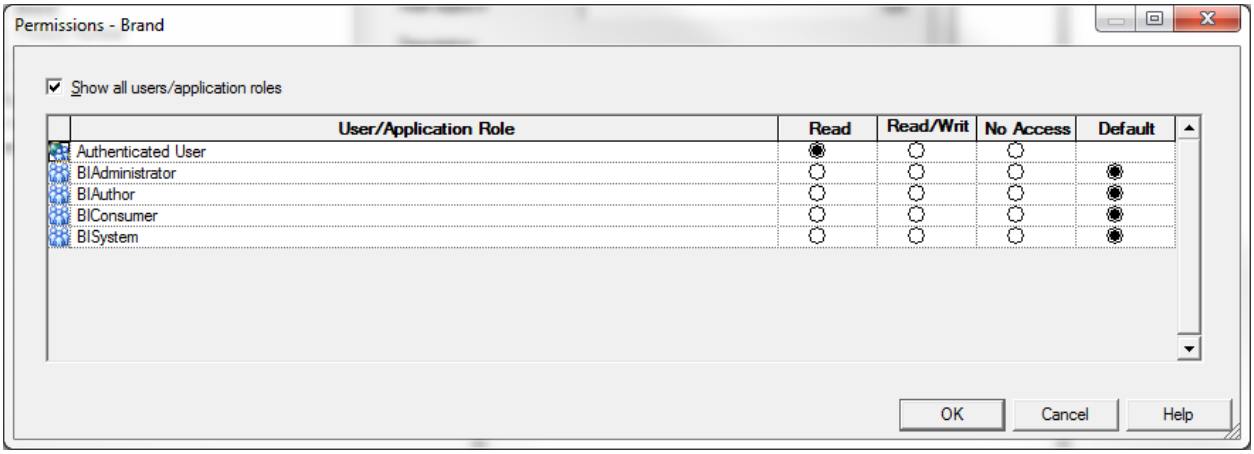


Three Layers

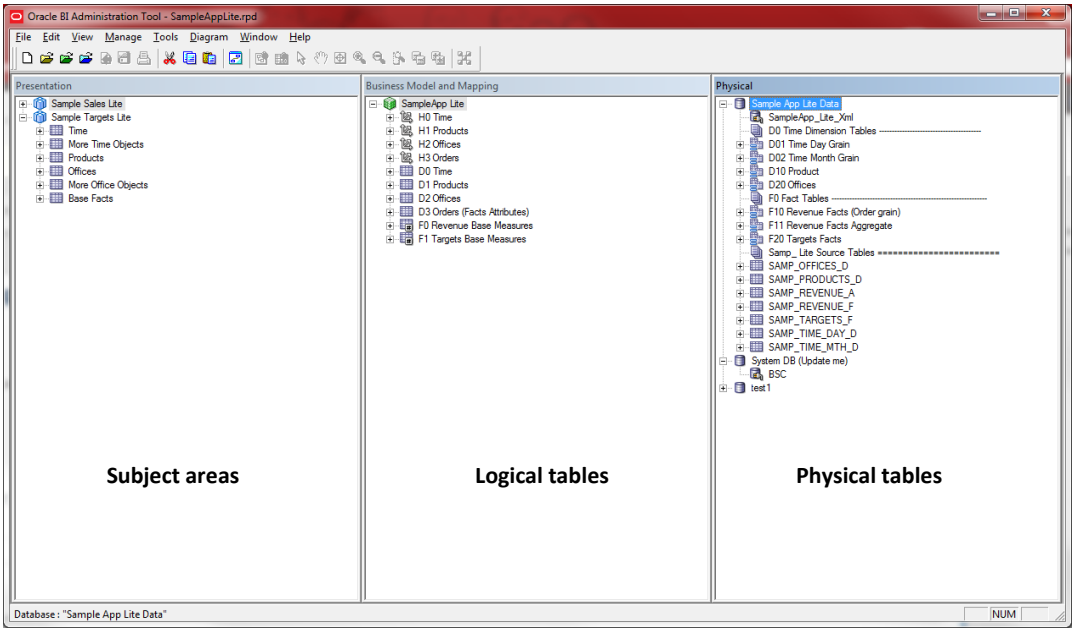
The OBIEE Repository is comprised of three layers. A very simplistic summary is below:

- **Physical layer**: Defines all database or data source connections (user id and passwords are entered and stored here), the physical table and columns, primary and foreign key relationships.
- **Business Model Mapping layer (BMM)**: Referencing the physical layer, here is where logical structures are built and aggregation rules are defined. The BMM is really the heart of an OBIEE application
- **Presentation layer**: Referencing the BMM, this layer presents the tables and columns to end users. For example, remove unwanted columns or rename awkwardly named columns. Most importantly from a security perspective, the presentation layer defines security rules for the BMM objects and maps them to Fusion Middleware application roles.

Presentation layer security rule



OBIEE BI RPD Repository

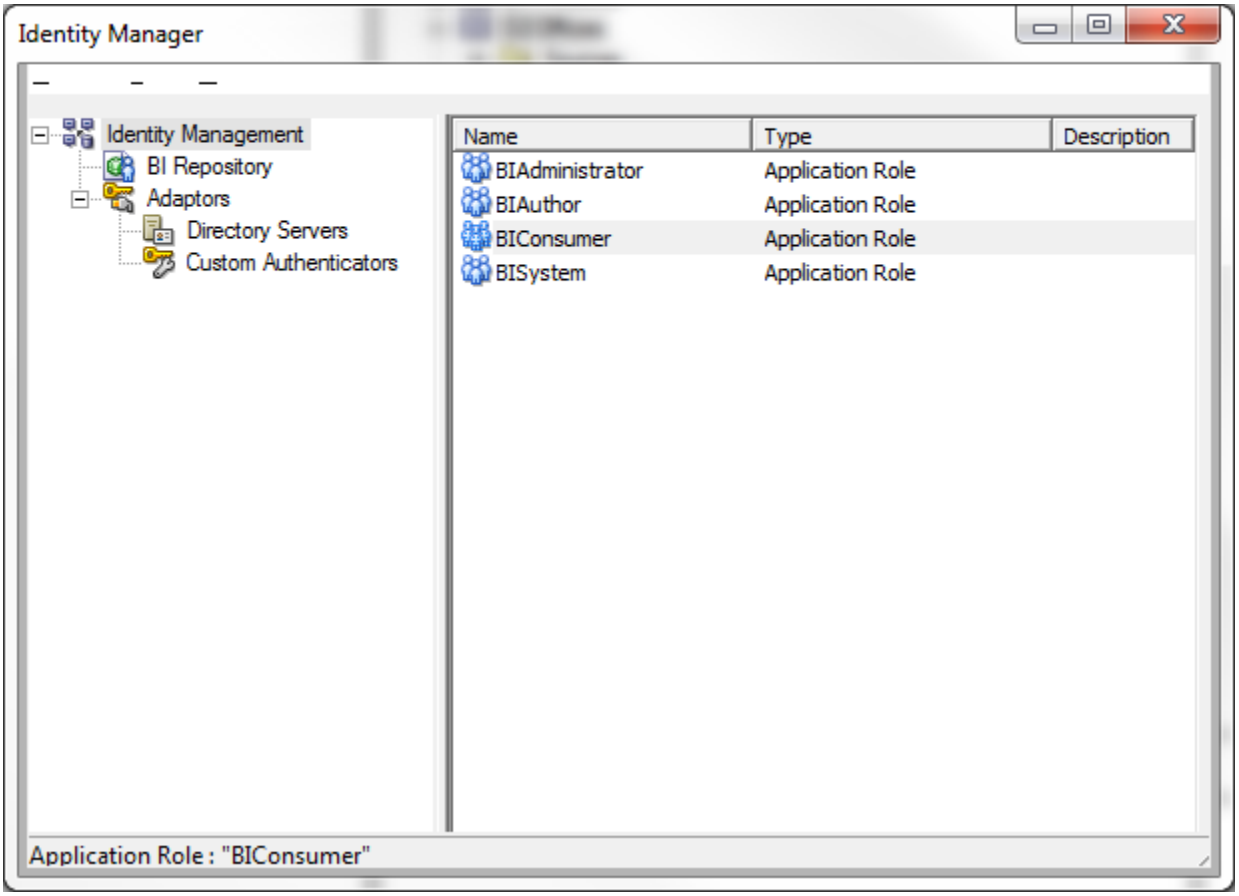


Object and Data Level Security

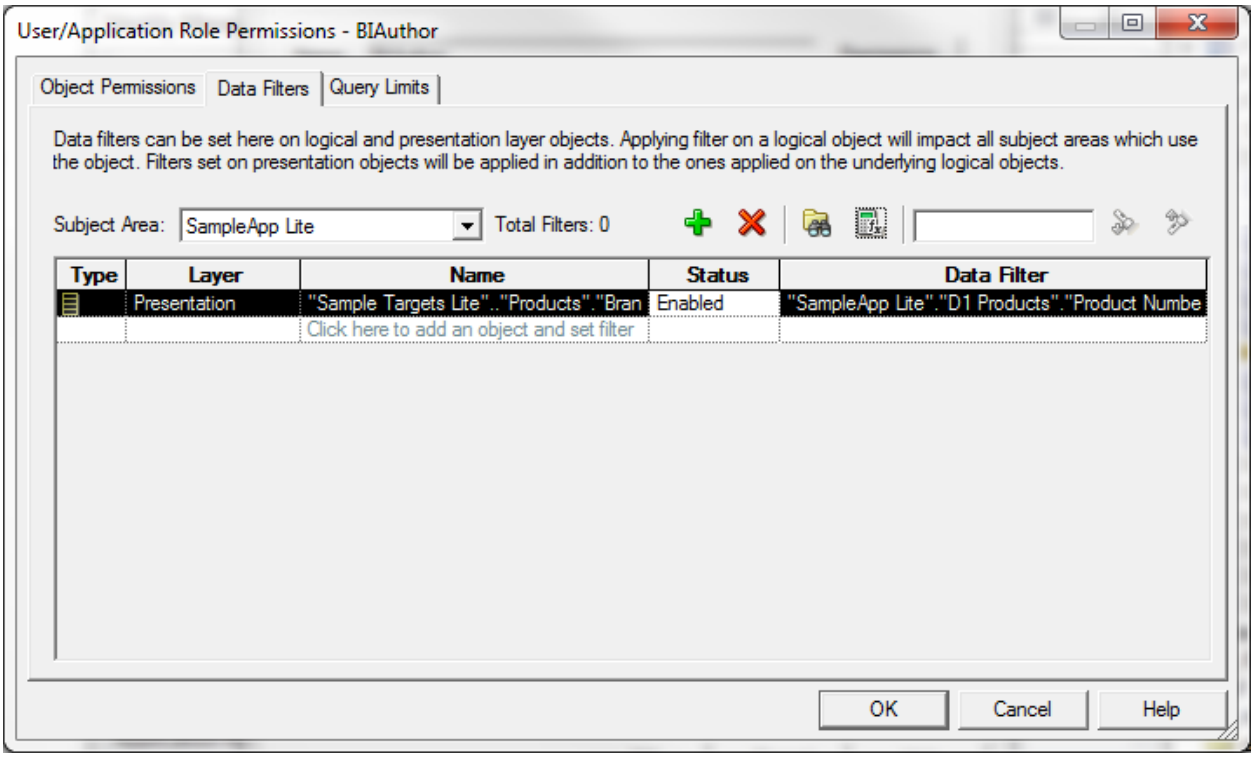
Object (Physical layer) and Data (BMM) level security is defined within the identity manager in the Repository. Object security can be set to either allow or deny access to a physical table or column. Data security allows rules to be applied to logical tables or columns (BMM layer). These rules can use static values as well as session variables.

Navigation: Open identity manager within the RPD -> select user or role -> click on permissions

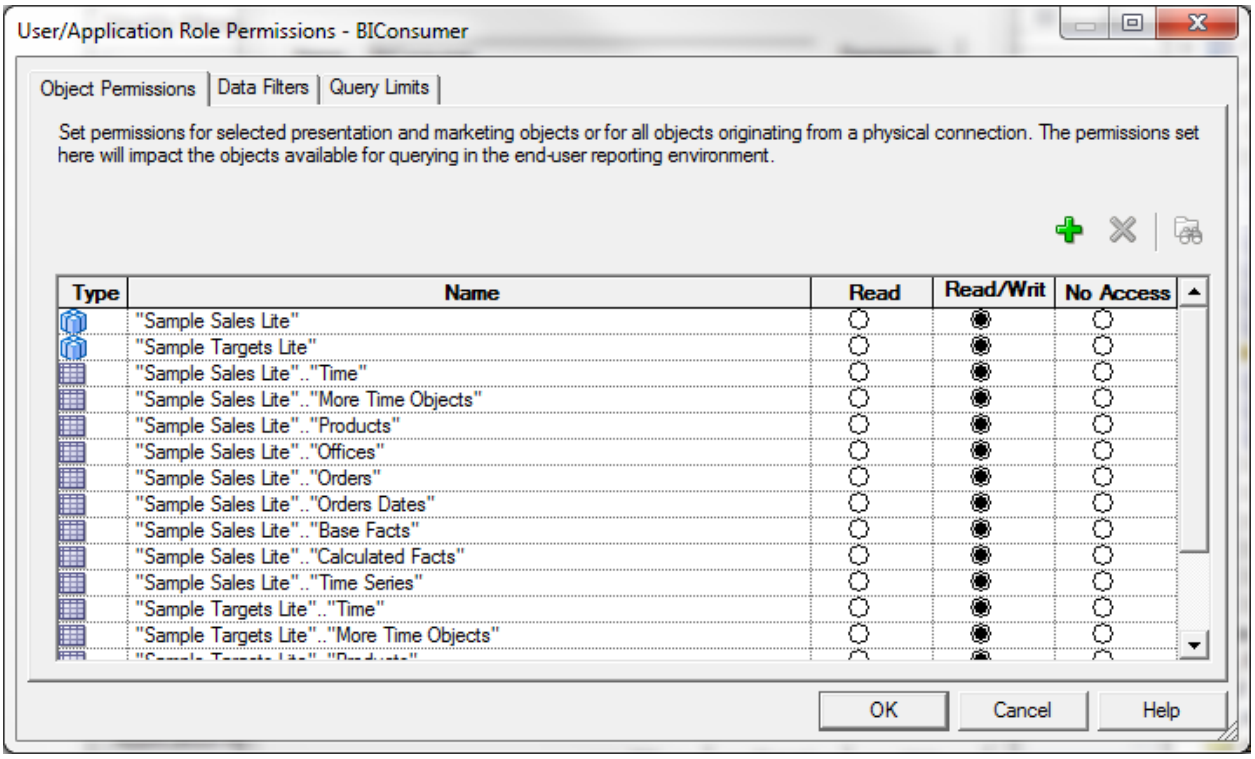
Identity Manager



Data Filter


User/Application Role Permissions - BIAuthor
Object Permissions
Data Filters
Query Limits
Data filters can be set here on logical and presentation layer objects. Applying filter on a logical object will impact all subject areas which use the object. Filters set on presentation objects will be applied in addition to the ones applied on the underlying logical objects.
Subject Area:
SampleApp Lite
Total Filters: 0
Type
Layer
Name
Status
Data Filter
Presentation
"Sample Targets Lite"."Products"."Bran
Enabled
"SampleApp Lite"."D1 Products"."Product Numbe
Click here to add an object and set filter
OK
Cancel
Help

Object Filter



PRESENTATION CATALOG

The presentation catalog (Web Catalog) stores the content that users create within OBIEE. While the Catalog uses the presentation layer objects, do not confuse the presentation layer within the RPD with the presentation catalog. The presentation catalog includes objects such as folders, shortcuts, filters, KPIs and dashboards. These objects are built using the presentation layer within the RPD.

The difference between RPD and Catalog security is that Repository level restrictions give the most flexibility as they can be either course-grained or fine-grained based on the data. Catalog level restrictions are more course-grained as they are applied to entire subject areas and/or objects.

To access an object in the catalog users must have security and can use either the BI client or web user interface. The BI client for the Web Catalog is installed along with the BI Admin client.

Administration Manager Privileges

ORACLE Business Intelligence

Administration

Manage Privileges

This page allows you to view and administer privileges associated with various components of Oracle Business Intelligence.

Access	Access to Dashboards	BI Consumer Role
	Access to Answers	BI Author Role
	Access to BI Composer	BI Author Role
	Access to Delivers	BI Author Role
	Access to Briefing Books	BI Consumer Role
	Access to Mobile	BI Consumer Role
	Access to Administration	BI Administrator Role, BI Consumer Role
	Access to Segments	BI Consumer Role
	Access to Segment Trees	BI Author Role
	Access to List Formats	BI Author Role
	Access to Metadata Dictionary	BI Author Role
	Access to Oracle BI for Microsoft Office	BI Consumer Role
	Access to Oracle BI Client Installer	BI Consumer Role
	Access to KPI Builder	BI Author Role
	Access to Scorecard	BI Consumer Role
Actions	Create Navigate Actions	BI Consumer Role
	Create Invoke Actions	BI Author Role
	Save Actions containing embedded HTML	BI Administrator Role
Admin: Catalog	Change Permissions	BI Author Role
	Toggle Maintenance Mode	BI Administrator Role
Admin: General	Manage Sessions	BI Administrator Role
	Manage Dashboards	BI Author Role
	See sessions IDs	BI Administrator Role
	Issue SQL Directly	Authenticated User, BI Administrator Role, BI Author Role, BI Consumer Role, BI System Role
	View System Information	BI Administrator Role, BI Author Role
	Performance Monitor	BI Administrator Role
	Manage Agent Sessions	BI Administrator Role
	Manage Device Types	BI Administrator Role
	Manage Map Data	BI Administrator Role

Access Control Lists (ACL) are defined for each object in the catalog. Within the file system the ACLs are stored in the *.ATR files which may be viewed through a HEX editor. A 16-digit binary representation is used similar to UNIX (e.g. 777). There are six different types of permissions for each object:

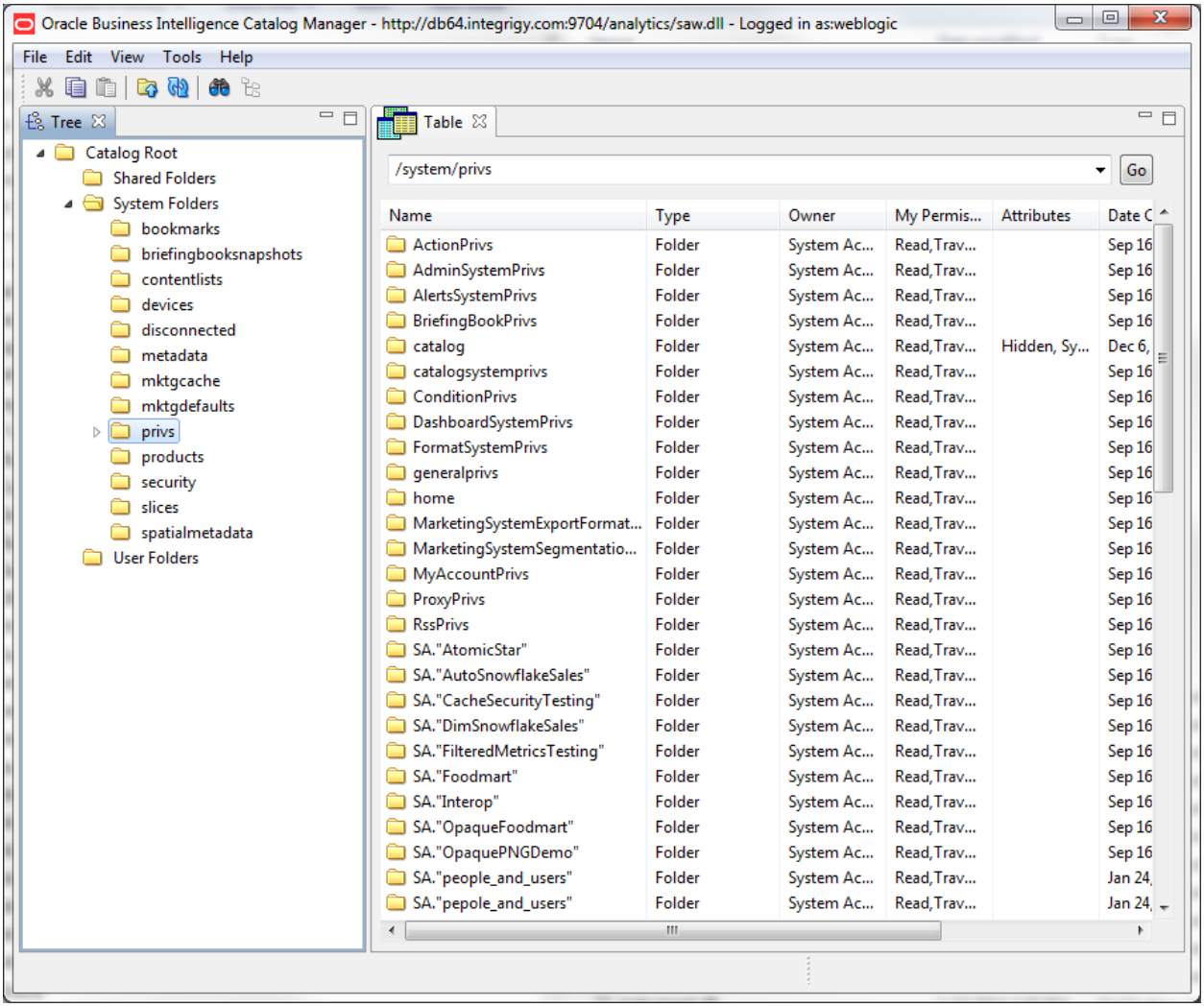
- Full control
- Modify

- Open
- Traverse
- No Access
- Custom

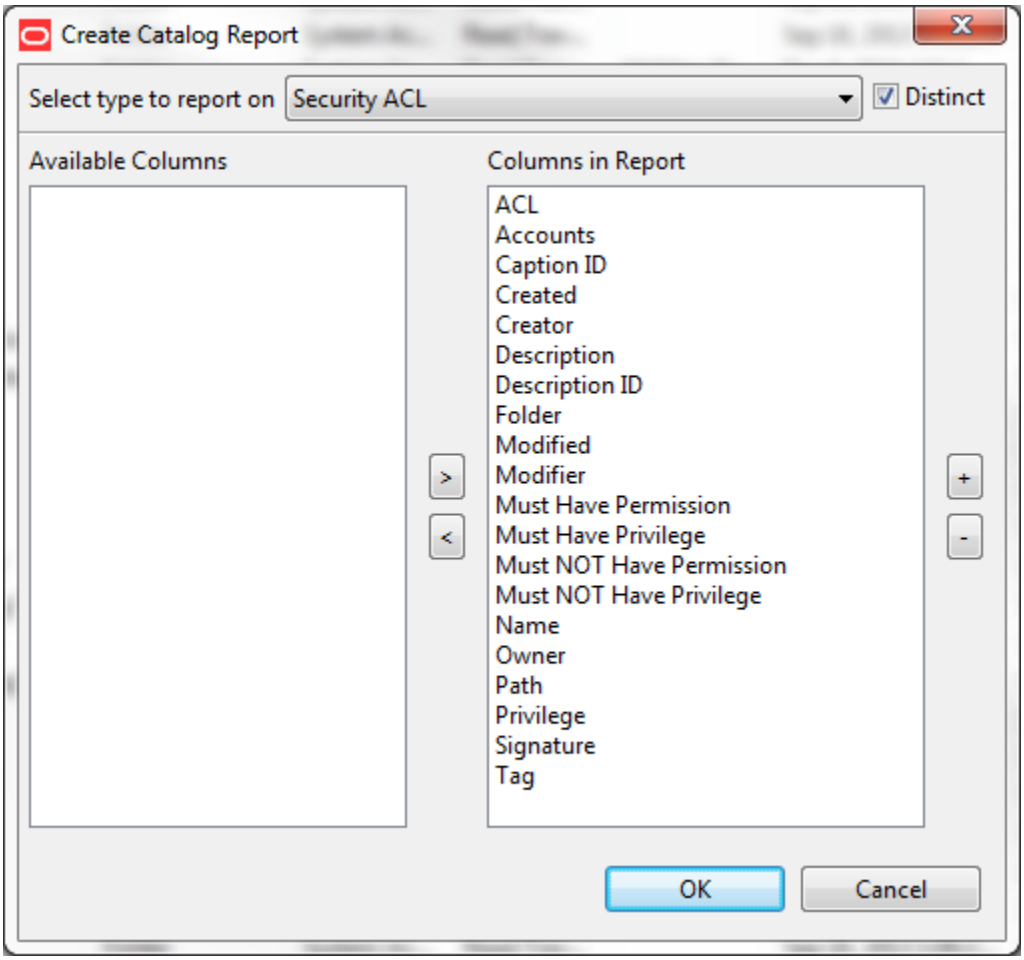
In 11g the catalog is located here:
`$ORACLE_INSTANCE/bifoundation/OracleBIPresentationServicesComponent/catalog`

From a security perspective, the permission reports that are able to be generated from the Web Catalog client tool are very valuable and can be exported to Excel for further analysis. For example, these reports can provide system generated reports for who can avoid OBIEE security and issue Direct SQL or has rights to Write-Back to the database. The security ACL will report on who has such Administration privileges.

OIBEE Web Catalog Client



OIBEE Web Catalog Report Parameters



LOGGING

OBIEE offers standard functionality for application level logging. This logging should be considered as one component of the overall logging approach and strategy. The operating system and database(s) supporting OBIEE should be using a centralized logging solution (most likely syslog) and it is also possible to parse the OBIEE logs for syslog consolidation.

For further information on OBIEE logging refer to the Oracle Fusion Middleware System Administrator's Guide for OBIEE 11g (part number E10541-02), chapter eight.

To configure OBIEE logging, the BI Admin client tool is used to set the overall default log level for the RPD as well as identify specific users to be logged. The log level can differ among users. No logging is possible for a role.

OBIEE Log Level	Description
Level 0	No Logging
Level 1	Logs the SQL statement issued from the client application. Also logs the following: ▪ Physical query response time — time for a query to be processed in the back-end database. ▪ Number of physical queries — the number of queries that are processed by the back-end database. ▪ Cumulative time — sum of time for all physical queries for a request (that is, the sum of all back-end database processing times and DB-connect times). ▪ DB-Connect time — time taken to connect to the back-end database. ▪ Query cache processing — time taken to process the logical query from the cache. ▪ Elapsed time —time that has elapsed from when the logical query is presented to the BI Server until the result is returned to the user. Elapsed time can never be less than response time, because elapsed time takes into account the small extra time between the logical queries being presented to the BI Server to the start of preparation of the query. In cases where this difference in time is negligible, the elapsed time equals the response time. ▪ Response time — time taken for the logical query to prepare, execute, and fetch the last record. This matches the TOTAL_TIME_SEC that is logged in usage tracking, as described in Section 9.3, "Description of the Usage Tracking Data." ▪ Compilation time — time taken to compile the logical query. For each query, logs the query status (success, failure, termination, or timeout), and the user ID, session ID, and request ID.
Level 2	Logs everything logged in Level 1 Additionally, for each query, logs the repository name, business model name, presentation catalog (called Subject Area in Answers) name, SQL for the queries issued against physical databases, queries issued against the cache, number of rows returned from each query against a physical database and from queries issued against the cache, and the number of rows returned to the client application.
Level 3	Logs everything logged in Level 2 Additionally, adds a log entry for the logical query plan, when a query that was supposed to seed the cache was not inserted into the cache, when existing cache entries are purged to make room for the current query, and when the attempt to update the exact match hit detector fails.
Level 4	Logs everything logged in Level 3 Additionally, logs the query execution plan.
Level 5	Logs everything logged in Level 4 Additionally, logs intermediate row counts at various points in the execution plan.

OBIEE Log Level	Description
Level 6	Not Used
Level 7	Not Used

Query Log content

The OBIEE query log has the following sections:

- **SQL Request** —SQL statement that is issued
- **General Query Information** —Lists the repository, the business model, and the subject area from which the query was run
- **Database Query** —Records the SQL statement that was sent to the underlying databases
- **Query Status** — The query success entry in the log indicates whether the query completed successfully, or failed

OBIEE log files

BI Component	Log File	Log File Directory
OPMN	debug.log	ORACLE_INSTANCE/diagnostics/logs/OPMN/opmn
OPMN	opmn.log	ORACLE_INSTANCE/diagnostics/logs/OPMN/opmn
BI Server	nqserver.log	ORACLE_INSTANCE/diagnostics/logs/ OracleBIServerComponent/coreapplication_obis1
BI Server Query	nquery<n>.log <n>=data and timestamp for example nqquery-20140109-2135.log	Oracle BI Server query Log ORACLE_INSTANCE/diagnostics/logs/OracleBIServer Component/coreapplication
BI Cluster Controller	nqcluster.log	ORACLE_INSTANCE/diagnostics/logs/ OracleBIClusterControllerComponent/coreapplicat ion_obiccs1
Oracle BI Scheduler	nqscheduler.log	ORACLE_INSTANCE/diagnostics/logs/ OracleBISchedulerComponent/coreapplication_obis ch1
Usage Tracking	NQAcct.yyymmdd.h hmmss.log	STORAGE_DIRECTORY parameter in the Usage Tracking section of the NQSConfig.INI file determines the location of usage tracking logs
Presentation Services	sawlog*.log (for example, sawlog0.log)	ORACLE_INSTANCE/diagnostics/logs/ OracleBIPresentationServicesComponent/ coreapplication_obips1 The configuration of this log (e.g. the writer setting to output to syslog or windows event log) is set in instanceconfig.xml
BI JavaHost	jh.log	ORACLE_INSTANCE/diagnostics/logs/ OracleBIJavaHostComponent/coreapplication_objh1

USAGE TRACKING

Knowing who ran what report, when and with what parameters is helpful not only for performance tuning but also for security. OBIEE 11g provides a sample RPD with a Usage Tracking subject area. The subject area will report on configuration and changes to the RPD as well as configuration changes to Enterprise Manager. To start using the functionality, one of the first steps is to copy the components from the sample RPD to the production RPD.

Usage tracking can also be redirected to log files. The STORAGE_DIRECTORY setting is in the NQSConfig.INI file. This can be set if OBIEE usage logs are being sent, for example, to a centralized SYSLOG database.

The User Tracking Sample RPD can be found here:

```
{OBIEE_11G_Instance}/bifoundation/OracleBIServerComponent/coreapplication_obis1/sample
/usagetracking
```

Additional Security Discussion Items

The following discussion items are of value to anyone attempting to assess the security of an OBIEE implementation and/or understand the security capabilities of OBIEE.

Migrate Security Configurations

Creation of non-production instances is a regular occurrence for enterprise software. A security discussion of OBIEE should include an inquiry into policies and procedures for migrations. To migrate security configurations among non-production instances as well as from non-production to production, OBIEE requires specific steps to be taken:

- **Migrate Identity Stores** (users and groups in the WLS LDAP server) - use standard WebLogic functionality. Can use both WebLogic GUI and/or a WLST.
- **Migrate Policy Stores** (application roles and policies) – the policy store is defined within system-jazn-data.xml. Do not simply copy this file, specific steps are required using standard WebLogic functionality.
- **Migrate Credential Stores** (containing all of the stored usernames and passwords used by the BI Server and system accounts) – usually done incrementally using WLST.
- **Migrate Repositories** – upload using standard WebLogic functionality or WLST to place new repository in effect.

Use Permission Reports

Prior to 11g OBIEE did not have an option to report on security permissions and rules defined within the RPD. With 11g permission reports are able to be generated for presentation layer objects. These reports are able to be exported as CSV files and are of great value to understanding security within an OBIEE application.

WriteBack

OBIEE has an advanced feature which allows for data to be written (created or updated) back to the database. This feature is referred to as 'Write Back'. The connection pool needs to be configured to allow for write back as well as the physical layer needs to flag each table to be updated as 'uncacheable'. The granting of write back privileges is then given to users within the Presentation Layer.

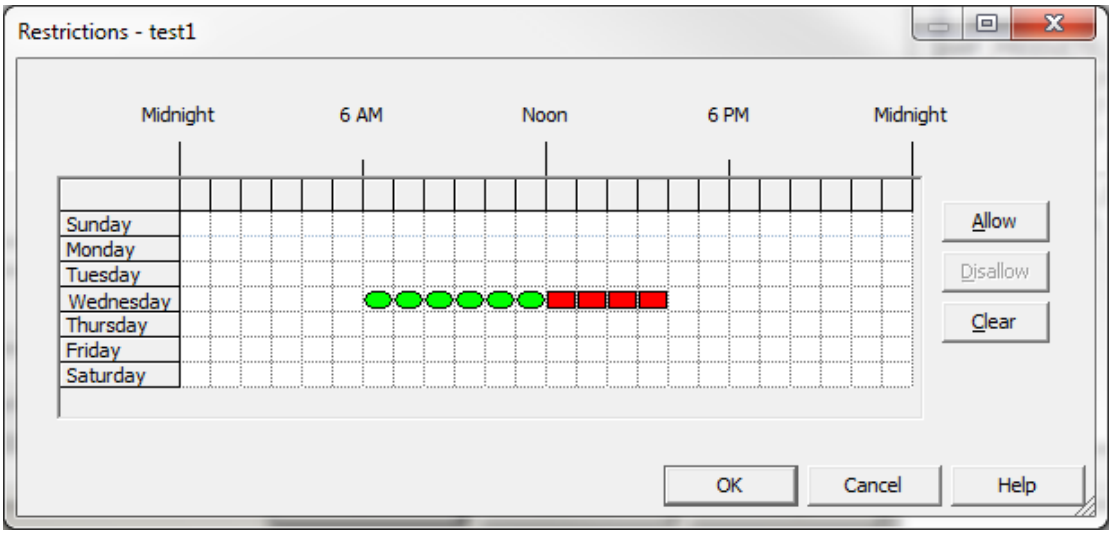
An example of WriteBack could be variables to assist with "what-if" modeling. The security risks if not set properly include data integrity if for example a user were to update their own salary.

Time limits

OBIEE, besides being able to set query limits for the number of rows to be returned, can also set limits for when a particular user or role can use OBIEE. For example, it is possible to define a rule to only allow a specific user or an application role to use the RPD during first shift and/or only Monday through Friday. Such rules may help build a more secure implementation.

Navigation: => Open identity manager within the RPD => Select role or user => Click on permissions => Click on query limits => Click on restrict

Time Restrictions for Users



Direct SQL Access

OBIEE allows for Direct SQL access to the repository database (data sources). While this can be invaluable for testing it can also be a large security risk, especially if full Data Manipulation Language (DML) is allowed. Direct SQL access can be globally enabled or disabled as well as set at a role or individual level.

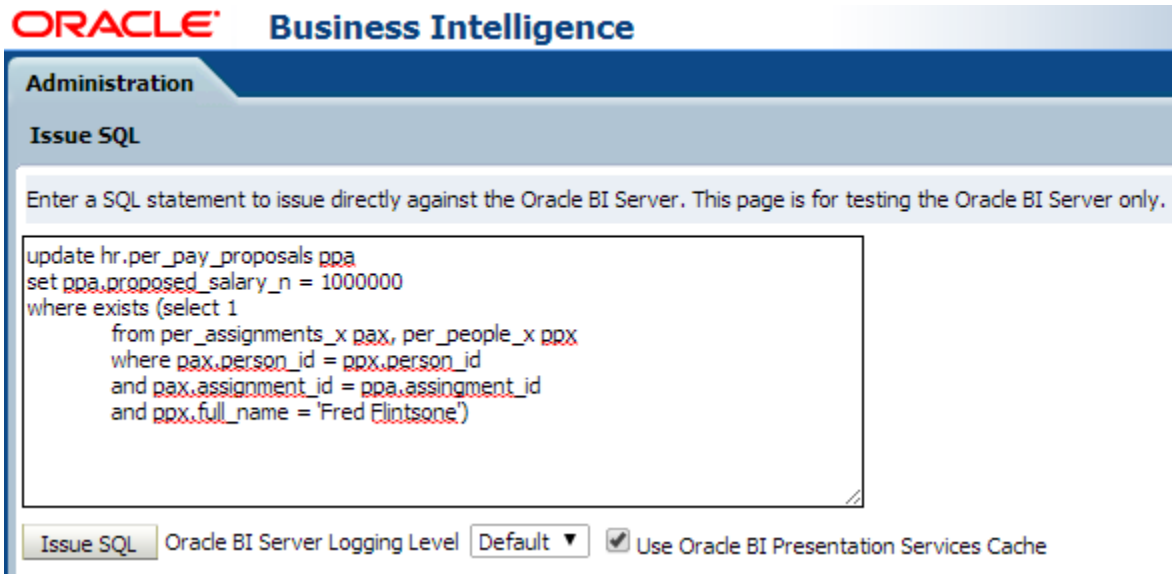
Any security assessment of OBIEE needs to review the usage of Direct SQL Access. The risks of allowing Direct SQL include:

- Confidentiality – bypassing data (object) level security defined within the repository (RPD)
- Integrity – modifying or deleting data
- Availability – overloading database with inordinate requests

To restrict access to direct SQL:

- Global: => Settings => Administration => Manage Privilege
- Role or User: => Open identity manager within the RPD => Select role or user => Click on permissions => Click on query limits => Select Allow, Disallow or Ignore for Execute direct database requests

Example of Direct SQL



Go URL SQL Access

The Oracle BI Presentation Services Go URL can be used to incorporate specific Oracle Business Intelligence results into external portals or applications. It has a number of optional parameters and arguments. It can also set session variables. More importantly, it can also issue SQL and return tabular results. For these two reasons alone, how the Go URL is being used should be kept in mind during a security assessment.

For example:

> http://testobiee:9704/analytics/saw.dll?Go&SQL=select+Region,Dollars+from+SupplierSales
> where the FROM clause is the name of the Subject Area to query

Alternatively, the command IssueRawSQL can be used to bypass the Web processing and issue SQL directly against the BI Server.

Act-As and Impersonation

OBIEE allows for two options for a user to assume the identity of another user. This functionality exists for several reasons, including testing, end-user support and system integration. Any security assessment of OBIEE needs to identify all users and systems able to act-as or impersonate another user.

- **Impersonation** - exists more for integration, is less secure than 'Act-As' and from a security perspective is a 'backdoor' risk. The default OBIEE user, BISystem comes out-of-the-box with Impersonation enabled. IT security should be consulted if this is enabled.
- **Act-As** – is the better choice to use support. It is more secure and is fully integrated into the user interface as standard functionality.

To check the Act-As and Impersonation settings, look at the system session variables PROXY and PROXYLEVEL.

	Act-As	Impersonate
Level of access	Full or read-only access, on a single user	Full access
Users whose identity can be assumed by the proxy user	Defined list of users	Any and all users, anytime
Access method	Standard functionality of UI	Construct URL manually
How to know if being used	Both proxy and Target are shown in the UI	No indication given
Security risk	Little to none	Credentials exposed in plain text when URL submitted

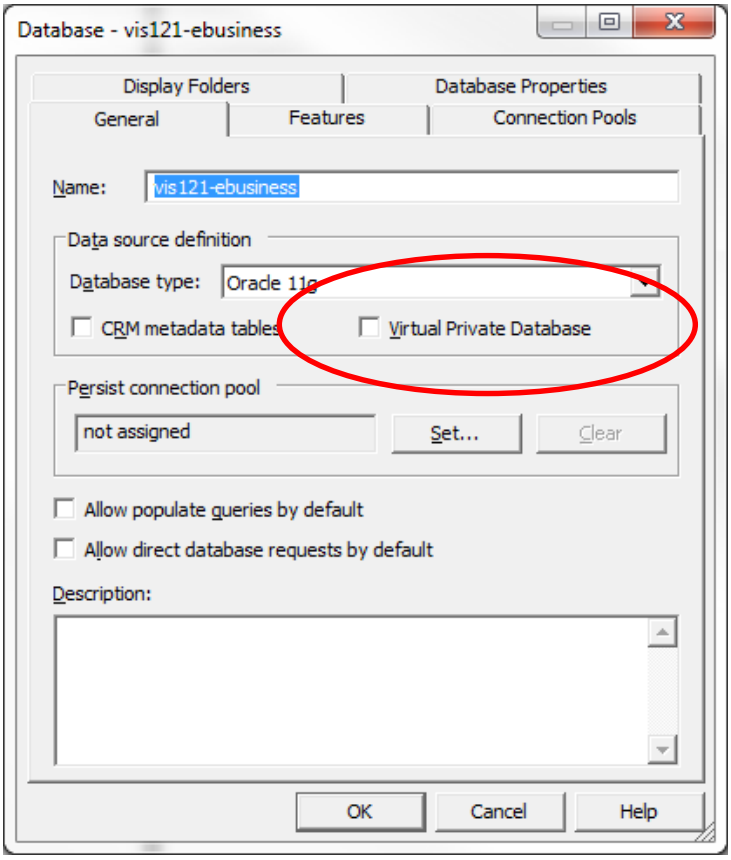
An example URL for impersonation is below. Please note, too, that the URL were exercised, the security risks could be increased because the user name password will then be captured and stored in the Apache log files.

http://server_name_or_ip_address/analytics/saw.dll?Go&NQuser=xxx&NQPassword=xxx

Virtual Private Databases

If a Virtual Private Database (VPD) is being used with an OBIEE connection, two things MUST be done. First, for the database connect, set the flag for Virtual Private Database to "Yes". Second, all variables supporting the security rules must be set to 'Security Sensitive'. Failing to properly set up VPD with OBIEE can result in VPD policies being bypassed due to the OBIEE cache incorrectly sharing result sets already in memory.

VPD Checkbox



Source Code Control

New with OBIEE 11g is the feature to store Repositories using a XML file format instead of the single RPD file. This allows for better support of source code control. Each component of the RPD file has its own XML file such that source code control tools such as Subversion can be used to check in or out each component. Best practice for development and security is to use source code control whenever possible.

Key Users and Passwords

The following is a list of users and accounts to be examined during any security assessment of OBIEE:

Account	Security Issue
OS owner of WebLogic	Ideally use user named something other than 'weblogic' and do not use welcome1 for a password as this is the default password for the sample OBIEE applications.
OS user that runs WebLogic	This can be any number of user accounts, but it should not be root or a privileged user. Do not hardcode this user's credentials in startup/shutdown scripts.
WebLogic gui user/administration user	End-user with full Administration rights to WebLogic
Repository password	Must be entered to open the RDP to edit. Password is also entered into Enterprise Manager to deploy the RPD file
BI Admin User	End-user with full Administration rights to OBIEE
BI System User	Administrator added by default and is used for service-to-service authentication. It is not intended to be used to write reports or administer the application. The password set during installation and stored in the credential store (Oracle Wallet). Do not change password without following the specific Oracle support instructions.
OracleSystemUser	Created during installation. User name can be change later but need to follow instructions.
Physical data source connection	These are stored in the RPD file.

ORACLE E-BUSINESS SUITE

There are two primary options for sharing authentication solutions with the Oracle E-Business Suite. The Oracle E-Business Suite and OBIEE both can take advantage of Oracle's Single Sign-On (SSO) solutions. If SSO is used, both OBIEE and the E-Business Suite would be subscribing applications.

The other option is for OBIEE to use the Oracle E-Business Suite for authentication. This solution requires that users first log into the E-Business Suite and from there exercise (click-on) a menu function to bring them into OBIEE without having to type a user name or password.

OBIEE AND ORACLE E-BUSINESS SUITE INTEGRATION

Configuring OBIEE to use the Oracle E-Business Suite for authentication is straight forward and can be completed in a test environment with only a small amount of effort. It is technically accomplished through the sharing of the E-Business Suite session cookie.

Further documentation on the specific steps to configure OBIEE to use the E-Business Suite for authentication can be found on Metalink as well as in the OBIEE documentation. A high level summary is as follows:

1. Using the BI Admin client tool, modify the RPD file to add a connection to the E-Business Suite database.
2. Add an initialization block to the RPD file that calls the E-Business Suite API APP_SESSION.validate_icx_session and then call FND_GLOBAL to collect the variables resp_id, resp_appl_id, security_group_id, resp_name, user_id, employee_id and user_name.
3. Edit the OBIEE configuration files authenicationschema.xml and instanceconfig.xml
4. Create a menu function to launch OBIEE. You must use the SSWA OracleOasis.jsp$mode=OBIEE
5. Populate the system profile option 'FND: Oracle Business Intelligence Suite EE base URL' with the url for OBIEE. For example: http://theobieeserver.yourcompany.com:9704
6. Upload the modified RPD file using Enterprise Manager and bounce all OBIEE services

Technical Summary

Authentication integration between OBIEE and the E-Business Suite is through a combination of a shared session cookie and a dynamic URL. The key to making it work are edits to OBIEE's instanceconfig.xml configuration file. It is in this file that OBIEE instructed is to look for the E-Business Suite session cookie.

With OBIEE looking for the E-Business Suite session cookie, when the user exercises the menu function with the Suite based on the SSWA OracleOasis.jsp, a URL is dynamically constructed using the profile option value for 'FND: Oracle Business Intelligence Suite EE base URL'. The URL is built as follows: [base_url]/analytics/saw.dll?[module_invoked]&acf=xxxxxxxxxx with xxxxxxxxxx being a random ten digit number.

Because OBIEE is listening for the E-Business Suite cookie, the random ten-digit number received by OBIEE triggers communication between the OBIEE and E-Business back-end servers. This communication uses the ten-

digit number to validate the request. Once the request is validated, additional authentication and authorization information is passed to OBIEE from the E-Business Suite.

With the secure communication established and validated using the ten- digit number, the E-Business Suite sends a session cookie to the user's browser for OBIEE to use.

PEOPLESOFT

In researching this paper OBIEE authentication through Peoplesoft was not attempted. For those considering options for OBIEE authentication through Peoplesoft the following sources will be of assistance:

- Understanding OBIEE Integration Within the PeopleSoft Framework
- "Integrating Oracle BI Enterprise with Peoplesoft", Oracle Metalink Note ID 1074402.1, Oracle Corporation, 20 August 2013 https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=1074402.1

REFERENCES

WEBLOGIC

1. “Oracle Fusion Middleware 11g Architecture and Management”, Shafii, Lee and Konduri, McGraw-Hill Oracle Press 2011
2. "Getting Started with Oracle WebLogic Server 12c: Developer's Guide ", Nunes and Oliveira, Packt Publishing, September 2013
3. “Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server 11*g* Release 1 (10.3.5) E13710”, Oracle Corporation, April 2011, http://docs.oracle.com/cd/E21764_01/web.1111/e13710/toc.htm
4. “Oracle Fusion Middleware Security Overview, 11*g* Release 1 (11.1.1) E12889-01”, Oracle Corporation, May 2009, http://docs.oracle.com/cd/E12839_01/core.1111/e12889/toc.htm
5. “Oracle Fusion Middleware Securing a Production Environment for Oracle WebLogic Server 11*g* Release 1 (10.3.4) E13705-04”, Oracle Corporation, January 2011, http://docs.oracle.com/cd/E17904_01/web.1111/e13705/toc.htm
6. “Oracle WebLogic Server WebLogic Scripting Tool10g Release 3 (10.3)”, Oracle Corporation, July 2008, http://docs.oracle.com/cd/E15051_01/wls/docs103/pdf/config_scripting.pdf
7. “Oracle Fusion Middleware, Introduction to Oracle WebLogic Server, 11g Release 1 (10.3.1) E13752-01”, Oracle Corporation, May 2009, http://docs.oracle.com/cd/E12839_01/web.1111/e13752/toc.htm

OBIEE

1. “Oracle Business Intelligence 11g Developers Guide”, Mark Rittman, McGraw-Hill Oracle Press, 2013
2. “Oracle Business Intelligence 11gR1 Cookbook”, Yilmaz, Packt Publishing , 2013
3. “Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition 11*g* Release 1 (11.1.1) E10543-07”, Oracle Corporation, February 2013, http://docs.oracle.com/cd/E28280_01/bi.1111/e10543/toc.htm
4. “Oracle Fusion Middleware Integrator's Guide for Oracle Business Intelligence Enterprise Edition 11*g* Release 1 (11.1.1) E16364-01”, Oracle Corporation, July 2010, http://docs.oracle.com/cd/E14571_01/bi.1111/e16364/title.htm

ORACLE SUPPORT

1. "Integrating Oracle Business Intelligence Applications (OIBEE 11g) With Oracle E-Business Suite”, Oracle Metalink Note ID 1343143.1, Oracle Corporation, 1 November 2013 https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=1343143.1

HISTORY

CHANGE HISTORY

1.0.0	March 2014	Initial Version

About Integrigy

Integrigy Corporation (www.integrigy.com)

Integrigy Corporation is a leader in application security for enterprise mission-critical applications. AppSentry, our application and database security assessment tool, assists companies in securing their largest and most important applications through detailed security audits and actionable recommendations. AppDefend, our enterprise web application firewall is specifically designed for the Oracle E-Business Suite. Integrigy Consulting offers comprehensive security assessment services for leading databases and ERP applications, enabling companies to leverage our in-depth knowledge of this significant threat to business operations.

INTEGRIGY
Integrigy Corporation
P.O. Box 81545
Chicago, Illinois 60681 USA
888/542-4802
www.integrigy.com