Appendix M

Required Permissions

SecureSphere requires a number of permissions to access databases for various tasks required in classification and assessment, and reading the content of Stored Procedures. This chapter lists the minimum permissions required to be able to successfully conduct these actions and includes the following:

- Required Permissions for Data Classification on page 1046
- Required Permissions for Assessment on page 1051
- Required Permissions for Stored Procedures on page 1065
- Required Permissions for Database User Rights Management on page 1070

Notes on MSSQL 2008 and Higher 1046
Required Permissions for File Classification 1046
Required Permissions for Data Classification 1046
Required Permissions for Assessment 1051
Required Permissions for Stored Procedures 1065
Required Permissions for Database User Rights Management 1070

Notes on MSSQL 2008 and Higher

In MSSQL 2008, the concept of the metadata security layer was introduced. The purpose of this feature is to control the data returned from security metadata views to different users. For example, when a user queries a view that lists database objects, only those objects on which the user has been granted viewing permissions are returned. These permissions only grant access to the metadata of these objects.

In addition, these permissions may also be denied from a user or role. The only role that is not subject to permissions denial is the "sysadmin" role. For this reason, we recommend using this role for the login used for inspection tasks.

Since we understand that it is not always acceptable to use a highly privileged user, we provide a detailed list of minimal permissions required. Note that in addition to the permission denial issue, new databases added to the server instance won't be covered by the inspection process until the permissions are manually modified.

Required Permissions for File Classification

In order to scan the contents of File servers for data classification and to obtain other file details you require the following permissions to all files and folders:

READ

Required Permissions for Data Classification

To run data classification, SecureSphere requires a number of baseline permissions in order to access the relevant aspects of the databases it needs to assess. This section reviews permissions required by SecureSphere to conduct data discovery and classification.

Supported Database Types for Classification 1047

Supported Database Types for Classification

Note: DB2 and IMS on z/OS are not supported.

SecureSphere supports a number of databases for classification. These include:

Permissions for MSSQL 2008 and Higher 1048
Permissions for Oracle Database Classification 1048
Permissions for Sybase Database Classification 1049
Permissions for Sybase IQ Database Classification 1049
Permissions for Informix Database Classification 1049
Permissions for MySQL Database Classification 1049
Permissions for DB2 Database Classification 1049
Permissions for Postgre Database Classification 1050
Permissions for Netezza Database Classification 1050
Permissions for Progress Database Classification 1051
Permissions for Teradata Database Classification 1051
Permissions for SAP Hana Database Classification 1051

Permissions for MSSQL 2008 and Higher

The following system permission is required on the master database in order to view the metadata of all objects within the server instance:

```
VIEW ANY DEFINITION
```

The following object permissions are required. They should be granted on each database within the scan scope:

```
execute on sys.sp_tables
execute on sys.sp_columns
select on sys.schemas
select on sys.synonyms
```

In addition, select permissions should be granted on all objects of interest. This can be achieved per database with the following permission:

```
select on database::<database_name>
```

In addition, the following permissions are required for each database to be scanned:

```
select on dbo.sysobjects
select on dbo.syscolumns
select on dbo.sysusers
select on dbo.systypes
select on dbo.syscomments
```

In addition, select permissions should be granted on all objects of interest. This can be achieved by adding the user as a member of the database role db_datareader

Permissions for Oracle Database Classification

The following permissions are required:

```
create session
select on sys.v_$fixed_table
select on sys.proxy_users
select on sys.dba_policies
select on sys.dba_encrypted_columns
select on sys.all_tab_columns
select on sys.all_objects
select on sys.dba_synonyms
select on sys.all_users
select any table
select any dictionary
```

Permissions for Sybase Database Classification

The following object permissions are required on the "master" database:

```
select on sysdatabases
select on sysusers
```

The following object permissions are required on the "sybsystemprocs" database:

```
execute on dbo.sp_tables
execute on dbo.sp_mda
execute on dbo.sp_sql_type_name
execute on dbo.sp_jdbc_getcatalogs
execute on dbo.   sp_jdbc_primarykey
execute on dbo.sp_columns
```

In addition, select permissions should be granted on all objects of interest.

Permissions for Sybase IQ Database Classification

The following object permissions are required:

```
execute on dbo.sp_jdbc_columns
execute on sp_mda
execute on dbo.sp_jdbc_tables
execute on dbo.sp_jdbc_getschemas
```

In addition, select permissions should be granted on all objects of interest.

Permissions for Informix Database Classification

For each database in the Informix instance the following permission is required:

```
grant connect to <user_name>
```

In addition, select permissions should be granted on all objects of interest.

Permissions for MySQL Database Classification

The following object permissions are required:

```
grant select on *.* to <user_name>
```

The reason granting a select privilege on all tables in all databases is the behavior of the INFORMATION_SCHEMA. INFORMATION_SCHEMA provides access to database metadata. Each MySQL user has the right to access these tables, but can see only the rows in the tables that correspond to objects for which the user has the proper access privileges.

Permissions for DB2 Database Classification

The following permissions are required in order to view the metadata of all objects within the server instance:

- For DB2 LUW Version 9.7 and above:

  The user that executes the classification scan should have the following authorities:

```
CONNECT
DATAACCESS
```

- For DB2 LUW Versions earlier than 9.7

  The user that executes the classification scan should have the following authorities:

  ```
  CONNECT
  DBADM
  ```

  Note: Such users can scan all database tables except those in ‘sysibmadm’ schema. In order to scan also object on this schema select permission on the relevant tables should also be granted.

Database Classification on DB2 with the SecureSphere Agent for z/OS

To run data classification with the **SecureSphere Agent for z/OS**, make sure the AUTHID (IMPV001?) running the DB Classification Scan has permissions to:

- Connect to DB2 via DDF
- Execute required stored procedures SYSIBM.SQLTABLES and IBM.SQLCOLUMNS (GRANT EXECUTE ON PROCEDURE)
- SELECT privilege on all tables to be accessed during the database classification scan

Additionally, the DB2 database requires specific configuration to conduct DB Classification with the **SecureSphere Agent for z/OS**. This configuration is available by default. If you have problems with classification on z/OS, or if you modify the configuration of your DB2 database for any reason, see the Customer Portal article **Required Database Configuration for Classification Scan on z/OS** for instructions on how to verify your configuration.

Permissions for Postgre Database Classification

The following object permissions are required:

```
select on pg_attribute
select on pg_database
select on pg_tables
select on pg_views
```

In addition, select permissions should be granted on all objects of interest

Permissions for Netezza Database Classification

The following object permissions are required:

```
select on _v_table
select on _v_relation_column
select on _v_view
```

In addition, select permissions should be granted on all objects of interest.

Permissions for Progress Database Classification

The following object permissions are required:

```
select on sysprogress.SYSCOLUMNS
select on sysprogress.SYSTABLES
```

In addition, select permissions should be granted on all objects of interest.

Permissions for Teradata Database Classification

The following object permissions are required:

```
select on DBC.Tables
select on DBC.Columns
select on DBC.UDTInfo
select on DBC.Databases
```

In addition, select permissions should be granted on all objects of interest.

Permissions for SAP Hana Database Classification

The following object permissions are required:

```
select on SYS.SCHEMAS
select on SYS.TABLES
select on SYS.TABLE_COLUMNS
select on SYS.VIEWS
```

In addition, select permissions should be granted on all objects of interest

Required Permissions for Assessment

To conduct assessments, SecureSphere requires a number of baseline permissions in order to access the relevant aspects of the databases it needs to assess. This section reviews permissions required by SecureSphere to conduct assessments, and includes the following:

Supported Database Types for Assessments 1052
Permissions for Oracle Database Assessments 1053
Permissions for MSSQL 2008 and Higher Database Assessments 1057
Permissions for DB2 Database Assessments 1061
Permissions for Sybase Database Assessments 1062
Permissions for Informix Database Assessments 1064
Permissions for PostgreSQL Database Assessments 1065

Supported Database Types for Assessments

SecureSphere supports five primary database types that you may want to configure manually to provide access to. These include:

- DB2
- Oracle
- MySQL
- MSSQL
- Sybase
- Informix

Permissions for Oracle Database Assessments

Oracle database assessments can be executed using a dedicated role that has read permissions on the relevant database objects and a user that holds the role.

Note:

- For information on configuring SecureSphere for Oracle Stored Procedure access, see Permissions for Oracle Stored Procedure Analysis on page 1065.
- For a note on Use with Unix, see Notes on Use with Oracle OS Assessments on Unix on page 1057
- For instructions on working with Oracle 12 and Oracle 12c, scroll down further in this topic.

The user and role can be created using the following script:

```
create user imperva identified by assessment;
create role assessment_role;
grant assessment_role to imperva;
grant create session to assessment_role;
grant select on sys.registry$history to assessment_role;
grant select on sys.dba_db_links to assessment_role;
grant select on sys.dba_objects to assessment_role;
grant select on sys.dba_priv_audit_opts to assessment_role;
grant select on sys.dba_profiles to assessment_role;
grant select on sys.dba_role_privs to assessment_role;
grant select on sys.dba_roles to assessment_role;
grant select on sys.dba_scheduler_jobs to assessment_role;
grant select on sys.dba_stmt_audit_opts to assessment_role;
grant select on sys.dba_sys_privs to assessment_role;
grant select on sys.dba_tab_privs to assessment_role;
grant select on sys.dba_users to assessment_role;
grant select on sys.link$ to assessment_role;
grant select on sys.profile$ to assessment_role;
grant select on sys.profname$ to assessment_role;
grant select on sys.resource_map to assessment_role;
grant select on sys.role_tab_privs to assessment_role;
grant select on sys.sysauth$ to assessment_role;
grant select on sys.user$ to assessment_role;
grant select on sys.v_$controlfile to assessment_role;
grant select on sys.v_$database to assessment_role;
grant select on sys.v_$datafile to assessment_role;
grant select on sys.v_$logfile to assessment_role;
grant select on sys.v_$parameter to assessment_role;
grant select on sys.v_$pwfile_users to assessment_role;
grant select on sys.dba_data_files to assessment_role;
grant select on sys.dba_external_tables to assessment_role;
grant select on sys.dba_jobs to assessment_role;
grant select on sys.dba_obj_audit_opts to assessment_role;
grant select on sys.dba_source to assessment_role;
grant select on sys.dba_tables to assessment_role;
grant select on sys.dba_ts_quotas to assessment_role;
grant select on sys.v_$log to assessment_role;
grant select on sys.dba_registry to assessment_role;
grant select on sys.v_$fixed_table to assessment_role;
grant select on sys.proxy_users to assessment_role;
grant select on sys.dba_policies to assessment_role;
grant select on sys.dba_encrypted_columns to assessment_role;
grant select on sys.dba_indexes to assessment_role;
grant select on sys.dba_ind_partitions to assessment_role;
grant select on sys.dba_tablespaces to assessment_role;
grant select on sys.dba_tab_partitions to assessment_role;
grant select on sys.dba_users_with_defpwd to assessment_role;
grant select on sys.dba_proxies to assessment_role;
grant execute on sys.dbms_crypto to assessment_role;
grant execute on sys.utl_raw to assessment_role;
grant select on sys.dba_constraints to assessment_role;
grant select on sys.DBA_FREE_SPACE to assessment_role;
grant select on sys.DBA_SEGMENTS to assessment_role;
grant select on sys.dba_tab_columns to assessment_role;
grant select on sys.audit$ to assessment_role;
grant select on SYS.default_pwd$ to assessment_role;
grant select on sys.v_$encrypted_tablespaces to assessment_role;
grant select on sys.V_$OPTION to assessment_role;
grant select on sys.v_$tablespace to assessment_role;
grant select on sys.dba_repcatlog to assessment_role;
grant select on sys.dba_libraries to assessment_role;
grant select on sys.dba_col_privs to assessment_role;
grant select on sys.obj$ to assessment_role;
grant select on sys.objauth$ to assessment_role;
```

Oracle 12

When working with Oracle 12 you need to additionally include the following permissions:

```
grant select on sys.v_$version to assessment_role;
grant select on sys.audit_unified_policies to assessment_role;
grant select on sys.audit_unified_enabled_policies to assessment_role;
grant select on sys.dba_col_privs to assessment_role;
grant select on sys.table_privilege_map to assessment_role;
grant select on sys.registry$sqlpatch to assessment_role;
grant execute on sys.dbms_qopatch to assessment_role;
grant select on sys.dba_priv_audit_opts to assessment_role;
grant select on sys.dba_obj_audit_opts to assessment_role;
grant select on sys.dba_proxies to assessment_role;
grant select on lbacsys.dba_sa_audit_options to assessment_role;
```

Oracle 12c Multitenant

When working with Oracle 12c Multitenant, you need to create a new common user and common role

```
create user c##imperva identified by assessment;
create role c##assessment_role;
grant c##assessment_role to c##imperva container=all;
grant connect to c##assessment_role container=all;
grant create session to c##assessment_role container=all;
grant select on sys.dba_priv_audit_opts to c##assessment_role container=all;
grant select on sys.dba_obj_audit_opts to c##assessment_role container=all;
grant select on sys.dba_proxies to c##assessment_role container=all;
```

Privileges must be granted to the new common role using following syntax:

```
grant select on <table_name> to c##assessment_role container=all;
```

And you must grant the following permissions:

```
grant select on sys.dba_db_links to c##assessment_role container=all;
grant select on sys.v_$parameter to c##assessment_role container=all;
grant select on sys.v_$version to c##assessment_role container=all;
grant select on sys.v_$log to c##assessment_role container=all;
grant select on sys.dba_tab_privs to c##assessment_role container=all;
grant select on sys.dba_role_privs to c##assessment_role container=all;
grant select on sys.dba_users to c##assessment_role container=all;
grant select on sys.dba_sys_privs to c##assessment_role container=all;
grant select on sys.dba_tables to c##assessment_role container=all;
grant select on sys.dba_jobs to c##assessment_role container=all;
grant select on sys.dba_scheduler_jobs to c##assessment_role container=all;
grant select on sys.dba_external_tables to c##assessment_role container=all;
grant select on sys.dba_objects to c##assessment_role container=all;
grant select on sys.profname$ to c##assessment_role container=all;
grant select on sys.dba_profiles to c##assessment_role container=all;
grant select on sys.dba_tables to c##assessment_role container=all;
grant select on sys.dba_repcatlog to c##assessment_role container=all;
grant select on sys.dba_users_with_defpwd to c##assessment_role container=all;
grant select on sys.dba_data_files to c##assessment_role container=all;
grant select on sys.dba_stmt_audit_opts to c##assessment_role container=all;
grant select on sys.audit_unified_policies to c##assessment_role
container=all;
grant select on sys.audit_unified_enabled_policies to c##assessment_role
container=all;
grant select on sys.dba_roles to c##assessment_role container=all;
grant select on sys.dba_segments to c##assessment_role container=all;
grant select on sys.dba_free_space to c##assessment_role container=all;
grant select on sys.dba_tablespaces to c##assessment_role container=all;
grant select on sys.v_$tablespace to c##assessment_role container=all;
grant select on sys.dba_col_privs to c##assessment_role container=all;
grant select on sys.dba_libraries to c##assessment_role container=all;
grant select on sys.user$ to c##assessment_role container=all;
grant select on sys.obj$ to c##assessment_role container=all;
grant select on sys.objauth$ to c##assessment_role container=all;
grant select on sys.table_privilege_map to c##assessment_role container=all;
grant select on sys.v_$database to c##assessment_role container=all;
grant select on sys.v_$encrypted_tablespaces to c##assessment_role
container=all;
grant select on sys.dba_encrypted_columns to c##assessment_role container=all;
grant select on sys.dba_tab_columns to c##assessment_role container=all;
grant select on sys.dba_constraints to c##assessment_role container=all;
grant select on sys.default_pwd$ to c##assessment_role container=all;
grant select on sys.v_$option to c##assessment_role container=all;
grant select on lbacsys.dba_sa_audit_options to c##assessment_role
container=all;
```

Notes on Use with Oracle OS Assessments on Unix

- Most OS assessments can be executed using the Oracle installation account.
- For HP-UX servers (other UNIX servers might also have the same issues), stty and tset commands as part of the profile scripts might cause the assessment tests to fail. This is because these commands are terminal commands and the scripts run non-interactively.

  In order to avoid this, add the condition below to wrap the commands for the user that runs the assessment tests (or any general profile file, such as /etc/profile). This condition determines whether the script has a controlling terminal, and if it doesn't, it won't run the terminal related commands.

```
---
if tty > /dev/null
then
   <stty/tset commands>
fi
---
```

- If "set -u" is used in the profile scripts, some OS tests might fail. This is due to the fact that the tests check variables that were not necessarily set in the environment.
- The test "Oracle Software Installation Account Granted Excessive Privileges" needs read access on /etc/sudoers, which is typically granted only to root

Notes on Use with Oracle OS Assessments on Windows

Oracle OS assessments on Windows need to be conducted using an account with administrative privileges.

Permissions for MSSQL 2008 and Higher Database Assessments

Database Assessments using MSSQL 2008 can be executed using a dedicated role that has permissions on the relevant database objects and a user that holds the role. The user and role should be created for each database in the server instance. Then, those users are mapped to a login that is used for the assessment process.

Notes:

- When you run the DISA (STIG) assessment policy, if you do not use a login that has the Control Server permission or is a member of the SYSADMIN server role, you can execute the scan using the minimum permissions explained below. In such a case, there may be errors from a few of the policy's tests.
- To run the seven tests in the policy **Password Strength Assessments for MS-SQL 2005 and Above**, the user must use a login that has the Control Server permission or is a member of the SYSADMIN server role, otherwise errors will be generated.
- In the commands below, the square brackets and their contents are not variable markers but are part of the syntax of the command set.

Creating a Login

The login can be created using the following script:

```
CREATE LOGIN [SecureSphere_login] WITH PASSWORD=N'assessment',
DEFAULT_DATABASE=[master]
```

Notes:

- If the database is defined as working with complex passwords, the password in this example will fail. In this case, you must use a complex password.
- [SecureSphere_login] should should be replaced with an actual Windows/SQL Login

Creating Users and Roles

The user and role for each database can be created using the following script:

```
CREATE SCHEMA [SecureSphere_schema]

go

CREATE USER [SecureSphere_user] FOR LOGIN [SecureSphere_login] WITH
DEFAULT_SCHEMA=[SecureSphere_schema]

go

ALTER AUTHORIZATION ON SCHEMA::[SecureSphere_schema] TO [SecureSphere_user]

go

CREATE ROLE [SecureSphere_role] AUTHORIZATION [SecureSphere_user]

go

ALTER ROLE SecureSphere_role ADD MEMBER [SecureSphere_user]

go
```

View Server State

When working with MSSQL 2012 Database Assessments, you additionally need to use the following script used to View Server State.

```
CREATE SERVER ROLE [SecureSphere_Server_Role]

GO

GRANT VIEW SERVER STATE TO [SecureSphere_Server_Role]

GO

ALTER SERVER ROLE [SecureSphere_Server_Role] ADD MEMBER [SecureSphere_login]

GO
```

Required Permissions for MSSQL 2008 and Higher Database Assessments

There are a number of types of required permissions for MSSQL 2008 and higher assessments. They include:

Required Login Permissions for MSSQL 2008 and Higher 1059

Required Permissions on Any Database on MSSQL 2008 and Higher 1059

Required Permissions on 'msdb' Database on MSSQL 2008 and Higher 1059

Required Permissions on 'master' database on MSSQL 2008 and Higher 1060

Required Login Permissions for MSSQL 2008 and Higher

The following permission is required in order to be able to view the metadata of all objects in the server instance:

```
grant VIEW ANY DEFINITION to SecureSphere_login
```

An alternative can be to manually grant view definition on all server and database objects, but this approach is not scalable since permission need to be maintained over time.

The function is executed during the assessment process and requires the following permission:

```
grant ALTER TRACE to SecureSphere_login
```

Required Permissions on Any Database on MSSQL 2008 and Higher

The following permissions are required on any database:

```
grant select on dbo.sysobjects to SecureSphere_role
grant select on dbo.sysprotects to SecureSphere_role
grant select on dbo.sysusers to SecureSphere_role
grant select on sys.asymmetric_keys to SecureSphere_role
grant select on sys.database_files to SecureSphere_role
grant select on sys.database_permissions to SecureSphere_role
grant select on sys.database_principals to SecureSphere_role
grant select on sys.database_role_members to SecureSphere_role
grant select on sys.key_encryptions to SecureSphere_role
grant select on sys.objects to SecureSphere_role
grant select on sys.procedures to SecureSphere_role
grant select on sys.schemas to SecureSphere_role
grant select on sys.sql_modules to SecureSphere_role
grant select on sys.symmetric_keys to SecureSphere_role
grant select on sys.system_objects to SecureSphere_role
```

Required Permissions on 'msdb' Database on MSSQL 2008 and Higher

In addition to the permissions required for any database, the following permissions are required on the 'msdb' database:

```
grant select on msdb.dbo.sysjobs to SecureSphere_role
grant select on msdb.dbo.sysproxies to SecureSphere_role
grant select on msdb.dbo.sysproxysubsystem to SecureSphere_role
grant select on msdb.dbo.sysjobhistory to SecureSphere_role
grant select on msdb.dbo.syssubsystems to SecureSphere_role
grant select on msdb.dbo.sysproxylogin to SecureSphere_role
grant select on msdb.sys.database_principals to SecureSphere_role
grant execute on msdb.dbo.sp_enum_login_for_proxy to SecureSphere_role
grant execute on msdb.dbo.sp_enum_proxy_for_subsystem to SecureSphere_role
```

Required Permissions on 'master' database on MSSQL 2008 and Higher

In addition to the permissions required for any database, the following permissions are needed on the 'master' database:

```
grant select on INFORMATION_SCHEMA.COLUMNS to SecureSphere_role
grant select on INFORMATION_SCHEMA.TABLE_PRIVILEGES to SecureSphere_role
grant select on sys.configurations to SecureSphere_role
grant select on sys.credentials to SecureSphere_role
grant select on sys.databases to SecureSphere_role
grant select on sys.fn_trace_geteventinfo to SecureSphere_role
grant select on sys.fn_trace_getinfo to SecureSphere_role
grant select on sys.linked_logins to SecureSphere_role
grant select on sys.master_files to SecureSphere_role
grant select on sys.master_key_passwords to SecureSphere_role
grant select on sys.server_permissions to SecureSphere_role
grant select on sys.server_principals to SecureSphere_role
grant select on sys.server_role_members to SecureSphere_role
grant select on sys.servers to SecureSphere_role
grant select on sys.service_broker_endpoints to SecureSphere_role
grant select on sys.soap_endpoints to SecureSphere_role
grant select on sys.sql_logins to SecureSphere_role
grant select on dbo.spt_values to SecureSphere_role
grant select on dbo.sysconfigures to SecureSphere_role
grant select on dbo.syscurconfigs to SecureSphere_role
grant select on dbo.sysdatabases to SecureSphere_role
grant select on dbo.syslogins to SecureSphere_role
grant select on INFORMATION_SCHEMA.SCHEMATA to SecureSphere_role
grant execute on sys.fn_isrolemember to SecureSphere_role
grant execute on sys.sp_dbfixedrolepermission to SecureSphere_role
grant execute on sys.sp_executesql to SecureSphere_role
grant execute on sys.sp_get_distributor to SecureSphere_role
grant execute on sys.sp_helpdbfixedrole to SecureSphere_role
grant execute on sys.sp_helpreplicationdboption to SecureSphere_role
grant execute on sys.sp_helprolemember to SecureSphere_role
grant execute on sys.sp_helprotect to SecureSphere_role
grant execute on sys.sp_helpsrvrolemember to SecureSphere_role
grant execute on sys.sp_helpuser to SecureSphere_role
grant execute on sys.sp_server_info to SecureSphere_role
grant execute on sys.xp_loginconfig to SecureSphere_role
```

Permissions for DB2 Database Assessments

User authentication in DB2 is managed using an external mechanism (for example an OS), with authorization being managed internally. The examples in this document used with DB2 databases integrate a user that has been created in the external mechanism and lists the user permissions needed to perform various tasks.

Note: For information on configuring SecureSphere for DB2 Stored Procedure access, see Permissions for DB2 Stored Procedure Analysis on page 1067.

While assuming the database user accessing the database for assessment is named "ASSESS", the following permissions are needed:

```
GRANT CONNECT ON DATABASE TO USER ASSESS;
GRANT EXECUTE ON FUNCTION SYSPROC.ENV_GET_INST_INFO TO USER ASSESS;
GRANT SELECT ON SYSIBM.SYSTABAUTH TO USER ASSESS;
GRANT SELECT ON SYSIBM.SYSCOLAUTH TO USER ASSESS;
GRANT SELECT ON SYSIBM.SYSDBAUTH TO USER ASSESS;
GRANT SELECT ON SYSIBMADM.DBMCFG TO USER ASSESS;
GRANT SELECT ON SYSIBMADM.DBCFG TO USER ASSESS;
GRANT SELECT ON SYSIBM.SYSDBAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.PACKAGEAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.TBSPACEAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.DBAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.SEQUENCEAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.INDEXAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.TABLES TO USER ASSESS;
GRANT SELECT ON SYSIBM.SYSSCHEMAAUTH TO USER ASSESS;
GRANT SELECT ON SYSIBM.SYSTABAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.LIBRARYAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.TABAUTH TO USER ASSESS;
GRANT SELECT ON SYSIBM.SYSROUTINEAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.SCHEMATA TO USER ASSESS;
GRANT SELECT ON SYSIBM.ROUTINES TO USER ASSESS;
GRANT SELECT ON SYSCAT.INDEXES TO USER ASSESS;
GRANT SELECT ON SYSCAT.SCHEMAAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.PACKAGES TO USER ASSESS;
GRANT SELECT ON SYSCAT.VIEWS TO USER ASSESS;
GRANT SELECT ON SYSCAT.TRIGGERS TO USER ASSESS;
GRANT SELECT ON SYSCAT.PASSTHRUAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.ROUTINEAUTH TO USER ASSESS;
GRANT SELECT ON SYSCAT.TABLESPACES TO USER ASSESS;
GRANT SELECT ON SYSCAT.SEQUENCES TO USER ASSESS;
GRANT SELECT ON SYSCAT.ROUTINES TO USER ASSESS;
GRANT SELECT ON SYSCAT.INDEXES TO USER ASSESS;
GRANT SELECT ON SYSCAT.PACKAGES TO USER ASSESS;
GRANT SELECT ON SYSCAT.SCHEMATA TO USER ASSESS;
GRANT SELECT ON SYSCAT.TRIGGERS TO USER ASSESS;
GRANT SELECT ON SYSCAT.TABLES TO USER ASSESS;
GRANT SELECT ON SYSCAT.VIEWS TO USER ASSESS;
GRANT SELECT ON SYSCAT.ROUTINES TO USER ASSESS;
GRANT EXECUTE ON FUNCTION SYSPROC.MON_GET_CONTAINER TO ASSESS;
```

Note: Linux/UNIX OS Assessments should be executed using the DB2 installation account.

Permissions for Sybase Database Assessments

Sybase database assessments can be executed using a dedicated group that has permissions on the relevant database objects and a user that holds the group. The user and group should be created for each database in the server instance. Finally, those users are mapped to a login that is used for the assessment process.

Note that several Assessments execute stored procedures can only be accessed by members of the server role 'sso_role'. The rest of the assessments can be executed using the permissions given below.

Note: For information on configuring SecureSphere for Sybase Stored Procedure access, see Permissions for Sybase Stored Procedure Analysis on page 1067.

The following Assessments can be executed only if the login holds the 'sso_role':

- Default Account Set to SA
- Login Auditing Options not Set to ON
- Disk Auditing Options Set to other than ON
- Logout Auditing Option Set to other than ON
- Failed Logins Attempts

In addition, new databases added to the server instance won't be inspected until permissions are manually modified. This can also be avoided by using "sso_role".

Creating Login

The login can be created using the following script:

```
exec sp_addlogin 'SecureSphere_login', 'assessment1', @defdb = 'master'
```

Creating Users and Groups

The user and group for each database can be created using the following script:

```
exec sp_addgroup 'SecureSphere_group'
go
exec sp_adduser 'SecureSphere_login' , 'SecureSphere_user' ,
'SecureSphere_group'
```

Required Permissions for Sybase

There are a number of types of required permissions for Sybase assessments. They include:

Required Permissions on Any Database for Sybase 1063
Required Permissions on 'master' database for Sybase 1063
Required Permissions on 'sybsystemprocs' database for Sybase 1063
Required Permissions on 'sybsecurity' database for Sybase 1064

Required Permissions on Any Database for Sybase

The following permissions are required on any database:

```
grant select on dbo.sysusers to SecureSphere_group
grant select on dbo.sysobjects to SecureSphere_group
grant select on dbo.sysprotects to SecureSphere_group
grant select on dbo.syscomments to SecureSphere_group
grant select on dbo.sysalternates to SecureSphere_group
```

Required Permissions on 'master' database for Sybase

In addition to the permissions required for any database, the following permissions are required on the 'master' database:

```
grant select on master.dbo.spt_limit_types to SecureSphere_group
grant select on master.dbo.spt_values to SecureSphere_group
grant select on master.dbo.sysattributes to SecureSphere_group
grant select on master.dbo.sysconfigures to SecureSphere_group
grant select on master.dbo.syscurconfigs to SecureSphere_group
grant select on master.dbo.sysdatabases to SecureSphere_group
grant select on master.dbo.sysloginroles to SecureSphere_group
grant select on master.dbo.syslogins to SecureSphere_group
grant select on master.dbo.sysremotelogins to SecureSphere_group
grant select on master.dbo.sysresourcelimits to SecureSphere_group
grant select on master.dbo.sysroles to SecureSphere_group
grant select on master.dbo.sysroles to SecureSphere_group
grant select on master.dbo.sysservers to SecureSphere_group
grant select on master.dbo.syssrvroles to SecureSphere_group
```

Required Permissions on 'sybsystemprocs' database for Sybase

In addition to the permissions required for any database, the following permissions are required on the 'sybsystemprocs' database:

```
grant execute on sybsystemprocs.dbo.sp_loginconfig to SecureSphere_group
```

Note: This permission should be granted only when Sybase is installed on Windows.

Required Permissions on 'sybsecurity' database for Sybase

In addition to the permissions required for any database, the following permissions are required on the 'sybsecurity' database:

```
grant select on dbo.systhresholds to SecureSphere_group
grant select on dbo.syssegments to SecureSphere_group
```

Permissions for Informix Database Assessments

Informix database assessments can be executed using a dedicated role that has read permissions on the relevant database objects and a user that holds the role.

Note: For information on configuring SecureSphere for Informix Stored Procedure access, see Permissions for Informix Stored Procedure Analysis on page 1069.

For every database in the Informix instance the following permission is required:

```
grant connect to <user_name>
```

The Assessments tests use the following tables:

In sysmaster database:

```
sysdatabases
sysopendb
sysconfig
```

In every database:

```
sysusers -get error: cannot modify system catalog
systables
sysprocedures
sysprocauth
systabauth
sysroutinelangs
sysroleauth
syslangauth
```

In addition to granting the permissions listed above, you also need to perform the following actions:

- Clear all permission that were granted to public
- Check the error 'cannot modify system catalog', thus it relevant for all Informix version

Permissions for PostgreSQL Database Assessments

Note: Part of the assessment tests can be executed only with a user that has a superuser permission, otherwise errors will be generated.

The following object permissions are required:

```
create role <role_name> with login password '<password>';

grant select on pg_authid to <role_name>;
grant select on pg_shadow to <role_name> ;
```

Required Permissions for Stored Procedures

In order to access stored procedures and understand their content, SecureSphere requires a number of baseline permissions. This section reviews permissions required by SecureSphere to access stored procedures, and includes the following:

Permissions for Oracle Stored Procedure Analysis 1065
Permissions for MSSQL 2008 and Higher Stored Procedure Analysis 1066
Permissions for DB2 Stored Procedure Analysis 1067
Permissions for Sybase Stored Procedure Analysis 1067
Permissions for Informix Stored Procedure Analysis 1069

Permissions for Oracle Stored Procedure Analysis

Stored Procedure Analysis can be executed using a dedicated role that has read permissions on the relevant database objects and a user that holds the role.

Note: For information on configuring SecureSphere for Oracle Database Assessments, see Permissions for Oracle Database Assessments on page 1053.

The user and role can be created using the following script:

```
create user imperva identified by sp
create role sp_role
grant sp_role to imperva
grant create session to sp_role
grant select on all_objects to sp_role
grant select on dba_objects to sp_role
grant select on dba_source to sp_role
grant select on sys.obj$ to sp_role
grant select on sys.public_dependency to sp_role
grant select on sys.source$ to sp_role
grant select on sys.user$ to sp_role
grant select any table to sp_role
```

Permissions for MSSQL 2008 and Higher Stored Procedure Analysis

Stored Procedure analysis can be executed using a dedicated role that has permissions on the relevant database objects and a user that holds the role. The user and role should be created on each database in the server instance. Finally, those users are mapped to a login that is used for the assessment process.

Notes:

- Fetching stored procedure permissions require administrator privileges
- For information on configuring SecureSphere for MSSQL 2008 and higher Database Assessments, see Required Permissions for MSSQL 2008 and Higher Database Assessments on page 1058.

Creating Login

The login can be created using the following script:

```
CREATE LOGIN [sp_login] WITH PASSWORD=N'password', DEFAULT_DATABASE=[master]
```

Notes:

- If the database is defined as working with complex passwords, the password in this example will fail. In this case, you must use a complex password.
- [sp_login] should be replaced with an actual Windows/SQL Login

Creating Users and Roles

The user and role for each database can be created using the following script:

```
CREATE SCHEMA [sp_schema]
go
CREATE USER [sp_user] FOR LOGIN [sp_login] WITH DEFAULT_SCHEMA=[sp_schema]
go
ALTER AUTHORIZATION ON SCHEMA::[sp_schema] TO [sp_user]
go
CREATE ROLE [sp_role] AUTHORIZATION [sp_user]
go
ALTER ROLE sp_role ADD MEMBER [sp_user]
go
```

Required Permissions for MSSQL 2008 and Higher Stored Procedure Analysis

Required Login Permissions for MSSQL 2008 and Higher

The following permission is required in order to view the metadata of all stored procedures:

```
grant VIEW ANY DEFINITION to sp_login
```

Required Permissions on any database for MSSQL 2008 and Higher

The following permissions are required on any database.

```
grant showplan to sp_role
grant execute to sp_role
grant select on dbo.sysobjects to sp_role
grant select on dbo.syscomments to sp_role
```

Required Permissions on 'master' database for MSSQL 2008 and Higher

In addition to the permission required for any database, the following permissions are required on the 'master' database:

```
grant select on dbo.sysdatabases to sp_role
```

Permissions for DB2 Stored Procedure Analysis

User authentication in DB2 is managed using an external mechanism (for example an OS), with authorization being managed internally. The examples in this document used with DB2 databases integrate a user that has been created in the external mechanism and lists the user permissions needed to perform various tasks.

Note: For information on configuring SecureSphere for DB2 Database Assessments, see Permissions for DB2 Database Assessments on page 1061.

Assume the user used is named "SPFETCH". The following permissions are needed:

```
GRANT CONNECT ON DATABASE TO USER SPFETCH
GRANT SELECT ON SYSIBM.SYSPLANDEP TO USER SPFETCH
GRANT SELECT ON SYSIBM.ROUTINES TO USER SPFETCH
GRANT SELECT ON SYSCAT.ROUTINES TO USER SPFETCH
GRANT SELECT ON SYSCAT.ROUTINEDEP TO USER SPFETCH
GRANT SELECT ON SYSCAT.TABDEP TO USER SPFETCH
```

Permissions for Sybase Stored Procedure Analysis

Stored Procedure analysis can be executed using a dedicated group that has permissions on the relevant database objects and a user that holds the group. The user and group should be created on each database in the server instance. Finally, those users are mapped to a login that is used for the assessment process.

Note: For information on configuring SecureSphere for Sybase Database Assessments, see Permissions for Sybase Database Assessments on page 1062.

Creating Login

The login can be created using the following script:

```
exec sp_addlogin 'sp_login', 'password', @defdb = 'master'
```

Creating Users and Groups

The user and groups for each database can be created using the following script:

```
exec sp_addgroup 'sp_group'
go
exec sp_adduser 'sp_login' , 'sp_user' , 'sp_group'
```

Required Permissions

In the following sections we provide the minimum permissions that are required in order to run Stored Procedure Analysis on Sybase databases.

Notice that the script in section Required Execute Permissions on any database for Sybase on page 1069, grant execute permission on all stored procedures that currently exist. Therefore, when a new stored procedure is created the script should be executed again. For this reason we recommend using a system administrator login which holds the 'sso_role' instead.

Required Select Permissions on any database for Sybase

The following select permissions are required on any database:

```
grant select on dbo.sysusers to sp_group
grant select on dbo.sysobjects to sp_group
grant select on dbo.syscomments to sp_group
grant select on dbo.sysdepends to sp_group
grant select on dbo.syscolumns to sp_group
```

Required Select Permissions on 'master' database for Sybase

In addition to the permission required for any database, the following select permissions are required on the 'master' database:

```
grant select on master.sysdatabases to sp_group
```

Required Execute Permissions on any database for Sybase

The following code grants execute permission on all stored procedures in the current database, run it on each database:

```
--create temp table with all stored procedure in the current database
create table #stored_procedures (sp nvarchar(400), owner nvarchar(400))

insert into #stored_procedures
select o.name, u.name from dbo.sysobjects o, dbo.sysusers u where o.uid =
u.uid and o.type = 'P'

--iterate thru all stored procedures and grant execute permission
go
declare cursor_sp cursor
for select * from #stored_procedures
go
open cursor_sp

declare @curr_sp varchar(400)
declare @curr_owner varchar(400)

fetch cursor_sp into @curr_sp, @curr_owner
while (@@sqlstatus = 0)
begin
      declare @command1 varchar(1000)
    select @command1 = "grant execute on " + @curr_owner + "." + @curr_sp + "
to sp_group"
    exec (@command1)
    fetch cursor_sp into @curr_sp, @curr_owner
end

close cursor_sp
deallocate cursor cursor_sp

drop table #stored_procedures
```

Permissions for Informix Stored Procedure Analysis

Stored Procedure analysis can be executed by granting the resource privilege to the user on any database. The permission required is as follows:

```
grant RESOURCE to <user_name>
```

Note: You need to run this command for each database from which you want to retrieve stored procedures.

Required Permissions for Database User Rights Management

In order to access tables that contain information regarding User Rights and understand their content, SecureSphere requires a number of permissions.

Note: Connect permissions need to be granted to all databases you want to scan with User Rights Management.

This section reviews permissions required by SecureSphere to access these tables, and includes the following:

Permissions for User Rights in Oracle 1070
Permissions for User Rights in MSSQL 2008 and Higher 1070
Permissions for User Rights in DB2 1071

Permissions for User Rights in Oracle

The following select permissions are required for the following Oracle objects:

```
select on dba_objects
select on dba_users
select on dba_tab_privs
select on dba_sys_privs
select on dba_role_privs
select on proxy_users
select on v_$pwfile_users
select on dba_roles
```

Permissions for User Rights in MSSQL 2008 and Higher

The following select permissions are required for the following MSSQL 2008 and higher objects:

For the Master database in MSSQL 2008 and Higher

```
select on master.sys.databases
select on master.sys.server_principals
select on master.sys.server_permissions
select on master.sys.server_role_members
```

For all databases in MSSQL 2008 and Higher

```
select on sys.schemas
select on sys.all_objects
select on sys.database_principals
select on sys.database_permissions
select on sys.database_role_members
```

Additionally, the 'VIEW ANY DEFINITION´ permission should be granted to the login in order to be able to view the metadata of all objects in the server instance.

Permissions for User Rights in DB2

The following select permissions are required all DB2 databases:

Up to and including DB2 v9.0

```
select on sysibmadm.dbmcfg
select on sysibm.sysuserauth
select on syscat.tabauth
select on syscat.dbauth
select on syscat.packageauth
select on syscat.indexauth
select on syscat.schemaauth
select on syscat.schemata
select on syscat.tables
select on syscat.packages
```

From DB2 v9.5

In v9.5, for security purposes, it is recommended that you give privileges to roles, and then assign the users to be a member of those roles.

```
select on syscat.roles
select on syscat.roleauth
```

OS credentials Requirements for DB2

In addition to permissions to databases in DB2, you also need to configure OS credentials to obtain User Rights information. This depends on your DB2 implementation, as follows:

- **For DB2 on Unix based system (Linux, Solaris, Hp-ux, Aix)**: Need to define a local OS user which have an access permissions to the files: /etc/passwd, /etc/group. It should be configured under the Server Group on the Servers tab.
- **For DB2 on Windows based system**: Need to configure user credentials to connect to an active directory. It should be configured on the Windows Domain in the Server Group.