Oracle® Health Sciences Central Designer

Security Guide

Release 7.0

F56111-03

ORACLE

Oracle Health Sciences Central Designer Security Guide, Release 7.0

F56111-03

Contents

Preface

Documentation accessibility v
Related resources v
Diversity and Inclusion v
Access to Oracle Support v
Additional copyright information vi

1 Security overview

Application security overview 1-1
General security principles 1-1

2 Secure installation and configuration

Installation overview 2-1
- Transport Layer Security (TLS) 2-1
- Signing authorizations and deployment packages 2-1
- Use digital certificates issued by Certificate Authorities 2-2
- Configure strong database passwords 2-2
- Close all unused ports 2-2
- Disable all unused services 2-2

Post-installation configuration 2-2
- Restrict access to Oracle Health Sciences Central Designer server machines 2-3
- Configure strong user passwords 2-3
- Configure rights and roles 2-3
- Configure IIS to prevent clickjacking 2-3

3 Security features

User security features 3-1
- Password configuration for user security 3-1
- Passwords for new users 3-2
- Login security 3-2

No data loss after a session transaction 3-2
Automatically deactivated user accounts 3-2
Restricted access to the application 3-2
Security events logs 3-3
Application security features 3-3
Rights assigned to roles 3-3
Users assigned to roles 3-4
Default user 3-4
Data security features 3-4
Protecting study objects 3-5
Audit trails for data security 3-5

4 Secure development overview

API overview 4-1
Rule sandbox details 4-1
Trusted assemblies 4-3
Untrusted assemblies 4-4
Elevating CAS permissions 4-4
General principles 4-4
Guidelines for elevating CAS permissions 4-5
Preventing hijacking of elevated permissions 4-5
Preventing access to code outside the sandbox 4-6
Preventing untrusted code from leaving the sandbox 4-6
Limitations of the rule sandbox security model 4-6
Oracle-hosted Oracle Central Designer 4-7
SQL injections 4-7
XML injection 4-7

Preface

This preface contains the following sections:

- Documentation accessibility
- Related resources
- Diversity and Inclusion
- Access to Oracle Support
- Additional copyright information

Documentation accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Related resources

All documentation and other supporting materials are available on the Oracle Help Center.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through Support Cloud.

Contact our Oracle Customer Support Services team by logging requests in one of the following locations:

- English interface of Oracle Health Sciences Customer Support Portal (https://hsgbu.custhelp.com/)
- Japanese interface of Oracle Health Sciences Customer Support Portal (https://hsgbu-jp.custhelp.com/)

You can also call our 24x7 help desk. For information, visit http://www.oracle.com/us/support/contact/health-sciences-cloud-support/index.html or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Additional copyright information

This documentation may include references to materials, offerings, or products that were previously offered by Phase Forward Inc. Certain materials, offerings, services, or products may no longer be offered or provided. Oracle and its affiliates cannot be held responsible for any such references should they appear in the text provided.

1

Security overview

In this chapter:

- Application security overview
- General security principles

Application security overview

To ensure security in the Oracle Health Sciences Central Designer application, carefully configure all system components, including the following third-party components:

- Web browsers
- Firewalls
- Load balancers
- Virtual Private Networks (VPNs)

General security principles

Keep software up to date

Keep all software versions and patches up to date.

Keep up to date on the latest Critical Patch Updates

Oracle continually improves its software and documentation. Critical Patch Updates are the primary means of releasing security fixes for Oracle products to customers with valid support contracts. They are released on the third Tuesday of January, April, July, and October (they were previously published on the Tuesday closest to the 17th day of January, April, July, and October).

Oracle highly recommends that customers apply these patches as soon as they are released.

Require complex and secure passwords

In the Oracle Health Sciences Central Designer Administrator application, an administrator should require that each user password meets the following requirements, which you set in the **Security** section, that you access by clicking **System Config**, and then **Settings** in the Oracle Health Sciences Central Designer Administrator application:

- Expires every 90 days. Configure this option in the **Passwords expire every** field.
- Has not been used recently. Configure the number of previously-used passwords that cannot be reused in the **Enforce password history** field.
- Contains a minimum of 8 characters. Configure this option in the **Minimum password length** field.
- Contains at least two of the following. Configure this option by setting the **Password complexity** setting to High.

- One letter and one number.
- One non-alphanumeric character.
- One upper-case and one lower-case letter, character, and at least either one number or special character.

For more information, see Configure strong user passwords.

Keep passwords private and secure

All users should change their passwords when they log in for the first time.

Tell users never to share passwords, write down passwords, or store passwords in files on their computers. For more information, see Passwords for new users.

Lock computers to protect data

Encourage users to lock computers that are left unattended. For more information, see Login security.

Provide only the necessary rights to perform an operation

Assign users to roles, and assign rights to roles so that users can perform only the tasks necessary for their jobs.

For more information, see:

- Rights assigned to roles.
- Users assigned to roles.

2

Secure installation and configuration

In this chapter:

- Installation overview
- Post-installation configuration

Installation overview

Use the information in this chapter to ensure the Oracle Central Designer application is installed and configured securely. For information about installing and configuring the Oracle Central Designer application, see the *Installation Guide*.

For more information, see:

- Transport Layer Security (TLS)
- Signing authorizations and deployment packages
- Use digital certificates issued by Certificate Authorities
- Configure strong database passwords
- Close all unused ports
- Disable all unused services

Transport Layer Security (TLS)

To encrypt the transmission of data between the application server and the client computers, you must enable Transport Layer Security (TLS) and obtain an X.509 certificate using your company certificate store or a third party.

For improved security, Oracle recommends that you disable SSL on the Oracle Health Sciences Central Designer application server and enable TLS 1.1 or above.

If you are deploying a study to an Oracle Health Sciences InForm server that uses TLS 1.1 or 1.2, run the following from an Administrator command prompt on the Oracle Health Sciences Central Designer application server to update the Windows registry:

```
reg add
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319" /v
SchUseStrongCrypto /t REG_DWORD /d 00000001
```

Signing authorizations and deployment packages

Signing web service authorizations and deployment packages is required. You must install the certificates used for signing on all application servers before you install the Oracle Health Sciences Central Designer application server. During the Oracle Health Sciences Central Designer application server installation, you are prompted to select a certificate for signing web service authorizations, and a certificate for signing deployment packages and Oracle Health Sciences InForm web service authorizations.

For more information, see the *Installation Guide*.

Use digital certificates issued by Certificate Authorities

A Certificate Authority (CA) assures users that the server information has been verified by a trusted source.

Oracle recommends that you use digital certificates that are issued by a Certificate Authority, and that do the following:

- Verify the server and domain.
- Provide at least a $1 million per year warranty.

Configure strong database passwords

When you install the Oracle Health Sciences Central Designer application, a system database administrator user is created. Only a system database administrator can perform the installation. Ensure all your database passwords are strong passwords.

Close all unused ports

Keep only the minimum number of ports open. You should close all ports not in use.

The Oracle Health Sciences Central Designer application uses the following ports:

- **80** —Used when the client applications are separated from the application servers and database server by a firewall, and do not use an SSL connection (HTTP).
- **443** —Used when the client applications are separated from the application servers and database server by a firewall, and use an SSL connection (HTTPS).
- **1521** —Used by the Oracle Listener service.
- **53000** —Used for communication between application servers behind a firewall.

Disable all unused services

The Oracle Health Sciences Central Designer application installs the job scheduler service on each application server. Make sure the job scheduler service is running, and disable unknown, unused services.

Post-installation configuration

In this section:

- Restrict access to Oracle Health Sciences Central Designer server machines
- Configure strong user passwords
- Configure rights and roles
- Configure IIS to prevent clickjacking

Restrict access to Oracle Health Sciences Central Designer server machines

Allow only administrator and system accounts access to the Oracle Health Sciences Central Designer application server and database server machines.

Limit the number of users with access to the server machines. Disable or delete any unnecessary users.

Configure strong user passwords

Configure password options to require a secure level of complexity. For example, a minimum required password length of 8 characters requires users to create more secure and complex passwords than a minimum required password length of 6 characters.

For more information, see Password configuration for user security.

Configure rights and roles

Assign users to roles, assign rights to roles, and assign user access to studies so that users can perform only the tasks necessary for their jobs.

For more information, see:

- Rights assigned to roles.
- Users assigned to roles.

Configure IIS to prevent clickjacking

To secure the web server and prevent clickjacking on the `http://<server name>/CentralDesignerInstall` page, from which you install the Oracle Health Sciences Central Designer and Oracle Health Sciences Central Designer Administrator applications, configure the HTTP response header in IIS.

For more information, see the *Installation Guide*.

3

Security features

In this chapter:

- User security features
- Application security features
- Data security features

User security features

In this section:

- Password configuration for user security
- Passwords for new users
- Login security
- No data loss after a session transaction
- Automatically deactivated user accounts
- Restricted access to the application
- Security events logs

Password configuration for user security

An administrator can define the following formatting, entry, and reuse requirements for passwords in the Oracle Health Sciences Central Designer Administrator software. For the recommended settings, see General security principles and the *Administration Guide*.

- Number of days before the password expires. Maximum recommended setting is 90 days.
- Number of recently used passwords that are remembered in the system and cannot be reused. Minimum recommended setting is four passwords.
- Minimum length of the password. Minimum recommended setting is eight characters.
- Number of login attempts allowed. Maximum recommended setting is three.
- Password complexity. Recommended setting is High.
- Amount of time that a user is locked out after exceeding the allowed number of login attempts. Recommended setting for the system and user accounts is 30 minutes.
- Length of time a user can be inactive before session timeout. Recommended setting is 20 minutes.
- Amount of time before a user must reauthenticate during a session. Recommended setting is four hours.

Passwords for new users

When you create new users, the users should change their passwords the next time they log in.

Login security

Users must enter their user names and passwords to log in. The application does not allow duplicate user names.

If either a user name or password is incorrect, an error message appears, but does not tell the user the value that is incorrect. Therefore, if someone else is using the account to attempt to log in, the message does not confirm either a user name or password.

No data loss after a session transaction

The Oracle Health Sciences Central Designer application is configured to require users to re-enter their user names and passwords after a defined period of inactivity. The user can log in and continue working in the application without losing data.

This security feature is controlled by the following settings in the Oracle Health Sciences Central Designer Administrator application:

- **Inactivity timeout** —Number of minutes of inactivity that can pass before the Oracle Health Sciences Central Designer application requires a user to log in again.
- **User must re-authenticate every** —Number of minutes that a session can be active before the Oracle Health Sciences Central Designer application requires a user to log in again.

Select values for these settings that work with your studies.

Automatically deactivated user accounts

The Oracle Health Sciences Central Designer application is configured to allow a defined number of attempts to log in correctly. When a user exceeds the number of allowed login attempts, which is defined in the Oracle Health Sciences Central Designer Administrator application, the user account is inactivated and the user cannot log in.

Only a user with the appropriate rights can activate an automatically inactivated account. Relevant rights include:

- Activate users.
- Terminate and deactivate users.

Restricted access to the application

You can restrict access to the application in the following ways.

- Terminate a user.
  Typically, you terminate users who leave the organization. Terminated users cannot log in to the application. All users, including terminated users, remain in the

study for audit purposes. A terminated user can never be activated or deactivated. If you terminate a user account, you can never use the account again.

- Deactivate a user.
  Typically, a user is automatically deactivated when the user fails to log in after the number of attempts set in the Oracle Health Sciences Central Designer Administrator software. After the user account is deactivated only an administrator can manually reactivate the user. The user must be reactivated before the user can work in the application.

Security events logs

The Oracle Health Sciences Central Designer application is configured to log the following security events:

- Successful logins.
- Failed logins.
- Password changes.
- Unauthorized access attempts.
- Unexpected failed validations of SAML tokens (indicating attempted bypass of validation).
- Changes to password management policies.

The following information is logged for every security event:

- Date and time.
- IP address.
- User name.
- Computer name.
- Event message, where applicable.

The data is captured in the PM_AUDIT_EVENT table. Because this table might grow rapidly over time, make sure to periodically export it, and then either truncate the table, or delete older rows.

Application security features

In this section:

- Rights assigned to roles
- Users assigned to roles
- Default user

Rights assigned to roles

A right is the permission to perform a specific activity. A role can have a library, study, or application scope. Each scope has a set of rights that you can grant to the role.

Rights grant access to different parts of the Oracle Health Sciences Central Designer and Oracle Health Sciences Central Designer Administrator applications. Entire parts of the application are hidden when users do not have the rights to work in those areas.

When a new user is created in the Oracle Health Sciences Central Designer application, an administrator with the right to modify user information assigns the user to a role in the library, study, or application scope, providing the user permissions to perform specific activities.

For example, a user can be assigned to the Study Collaboration role, which contains the right to create and assign tasks. The individual create and assign tasks right is static, but the group of rights assigned to the Study Collaboration role are configurable.

For more information, see the *Administration Guide*.

Users assigned to roles

After you review the rights that are assigned to roles and make any necessary changes, you can assign users to roles. A user assigned to a role has the rights that are granted to that role. Changes to a role are immediately applied to all users assigned to the role.

In addition, for each library and study role, a corresponding team exists. When you assign a user to a role, the user is also assigned to the team for that role. To assign a user to a team associated with a role, you rust first assign the user to the role.

A user can be assigned to a role that has one of the following scopes:

- **Library** —A user assigned to a library role is granted the rights associated with the role only in libraries where that user is also a member of the library team for the role.
- **Study** —A user assigned to a study role is granted the rights associated with the role only in studies where the user is also a member of the study team for the role.
- **Application** —A user assigned to an application role is granted all of the rights that are associated with the role, without restrictions.

You can also grant users the rights to perform administrative tasks such as configuring users, roles, rights, and system configuration settings. Administration users can also have unlimited rights in the Oracle Health Sciences Central Designer application. Ensure that you limit the users who have administration rights. For a description of administration rights, see the *Administration Guide*.

Default user

The Oracle Health Sciences Central Designer application installs the system user by default. During the installation, you configure a password for this user. In addition, you can configure the lockout time for the system user separately from all other users. By default, this user is assigned the superuser and DesignerAdministrator roles.

Oracle recommends that you create administrator accounts for individual users, and delete the system user after the initial application configuration.

Data security features

In this section:

- Protecting study objects
- Audit trails for data security

Protecting study objects

You can protect a library or a study to prevent users from making changes to study objects that you do not want to be modified.

When you protect a study or library, changes cannot be made to study objects or to the structure of the study or library.

When a study object is protected, its icon changes to reflect its protected state.

For more information, see the *Administration Guide*.

Audit trails for data security

Audit trails are comprehensive records that include information about each change that occurs in the Oracle Health Sciences Central Designer application.

The audit trail for the Oracle Health Sciences Central Designer application records each change, and for each change:

- Person who made the change.
- Date and time of the change.

You cannot modify data in an audit trail. For more information, see the *User Guide*.

4
Secure development overview

In this chapter:

- API overview
- Rule sandbox details
- Oracle-hosted Oracle Central Designer
- SQL injections
- XML injection

API overview

The Oracle Central Designer application provides an API for creating .NET DLLs to execute custom code (user-defined functions) called by rules through the Oracle Central Designer rule engine.

Rules and user-defined functions can be executed in the following places:

- On the Oracle Central Designer application client during interactive rule testing.
- On the Oracle Central Designer application server during study validation that executes rule tests.
- On the Oracle Health Sciences InForm application server during rule execution.

Because any .NET API can be called from these DLLs (file access, database access, and so on), developers must follow secure guidelines while developing the code. In addition, the Oracle Central Designer application provides a certificate-based mechanism to prevent untrusted DLLs from executing code that requires elevated permissions.

For more information on the .NET Code Access Security (CAS) model, see https://msdn.microsoft.com/en-us/library/dd233102(v=vs.100).aspx.

For secure coding guidelines for .NET applications, see https://msdn.microsoft.com/en-us/library/d55zzx87(v=vs.90).aspx.

Rule sandbox details

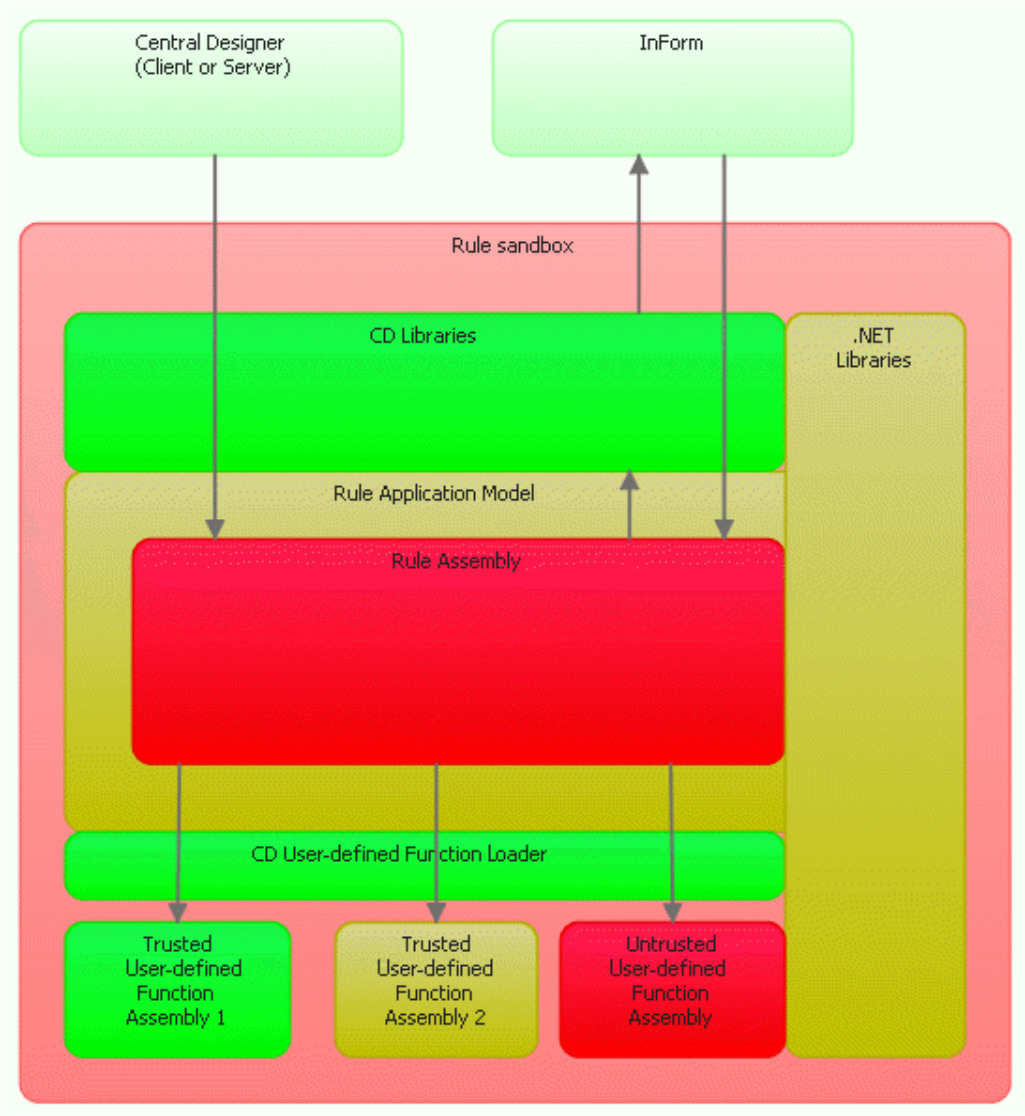
All user-defined function DLLs executed by the Oracle Central Designer rule engine run in a sandbox. The rule sandbox is a .NET AppDomain. You can think of it as a light-weight process inside the OS process.

The rule sandbox has the minimum set of permissions possible. This means the permission to execute code (SecurityPermissionFlag.Execution), which is the maximum restriction possible in .NET.

There are two types of user-defined function assemblies:

- **Trusted**—When loaded into the sandbox, code in trusted assemblies executes as safe-critical code or as security-critical, and can elevate permissions to perform extra operations, such as DB access.
- **Untrusted**—Untrusted user-defined function assemblies have the same permissions as a rule: only basic operations, which are not outlined in the *Rules Reference Guide* as requiring special permission. Untrusted code or transparent code can access only safe-critical code in assemblies marked with the attribute AllowPartiallyTrustedCallers.

The following diagram shows the relationship between the sandbox and the main application (Oracle Health Sciences InForm or Oracle Central Designer), and the assembly structure inside the sandbox.



Color codes:

- **Green**—Good, trusted code that does not allow partially trusted callers (security-critical).

- **Yellow**—Trusted code that can be called by untrusted code (security safe-critical).
- **Red**—Untrusted code (transparent code).

The two light-green boxes on top represent AppDomains for the main application: Oracle Central Designer or Oracle Health Sciences InForm. These domains run in full trust. AppDomain technology isolates them from the sandbox. Communication between AppDomains (represented by arrows on the diagrams) is done with remoting calls (inter-process communication or RPCs).

If you follow a call from the Oracle Central Designer box to invoke the rule from the rule assembly:

1. The call is first created in the sandbox.
2. The call loads Oracle Central Designer libraries inside the sandbox and uses a proxy to invoke the method from Oracle Central Designer libraries (calls a green code).
3. The Oracle Central Designer libraries load the rule assembly (red code) and invoke the method from the rule assembly.
4. The rule assembly can make calls through the Rule Application Model into Oracle Central Designer libraries (arrow going up).
5. The rule assembly can make calls through the Rule Application Model into user-defined functions (arrows going down).

> **Note:**
>
> The code executed when any of these calls are made is restricted to the set of sandbox permissions. It does not matter if it is trusted or untrusted code, unless trusted code chooses to temporarily elevate the permissions. .NET does not propagate elevated permissions to untrusted code.

For more information, see:

- Trusted assemblies
- Untrusted assemblies
- Elevating CAS permissions
- Preventing access to code outside the sandbox
- Preventing untrusted code from leaving the sandbox
- Limitations of the rule sandbox security model

Trusted assemblies

The rule sandbox has the following classes of trusted assemblies:

- Oracle Central Designer assemblies built by Oracle and signed by the corporate code signing tool are trusted (security-critical and security safe-critical).
- User-defined function assemblies with their public keys registered in the machine certificate store are loaded in the sandbox as yellow or green.

> **Note:**
>
> The decision to trust these assemblies is based on a manual procedure of installing certificates in the certificate store. A manual procedure creates the possibility of human error by registering incorrect public keys and establishing trust for a bad assembly.

While trusted assemblies run in the highly restrictive sandbox, they can elevate permissions. For the procedure and guidelines for such elevations, see Elevating CAS permissions.

The only high-level restriction that you can specify at the assembly level is the SecurityTransparentAttribute, which prohibits all elevations in the assembly. If the assembly needs to elevate permissions, the only guards against over-elevating are code review and code analysis tools.

Untrusted assemblies

Untrusted assemblies consist of:

- One rule assembly generated by the Oracle Central Designer application from the study structure. The generated rule assembly includes rules written by users.
- User-defined function assemblies that do not come from Oracle Services. Such assemblies either do not have strong names or their public keys are not installed in the machine certificate store.

Oracle recommends that you use strong names for all user-defined function assemblies because it has many benefits besides establishing trust. The Oracle Central Designer application also supports existing assemblies that are not strongly named. The .NET application does not allow assemblies without strong names as trusted assemblies.

Elevating CAS permissions

Every elevated permission is a potential security hole. Untrusted code can use it to do something it cannot do directly (for example, by calling the .NET library directly). Ideally, there should be no permission elevations at all. However, developers may need access to restricted resources, such as config files, registry values, and calls using reflection. Access to restricted resources requires elevated permissions.

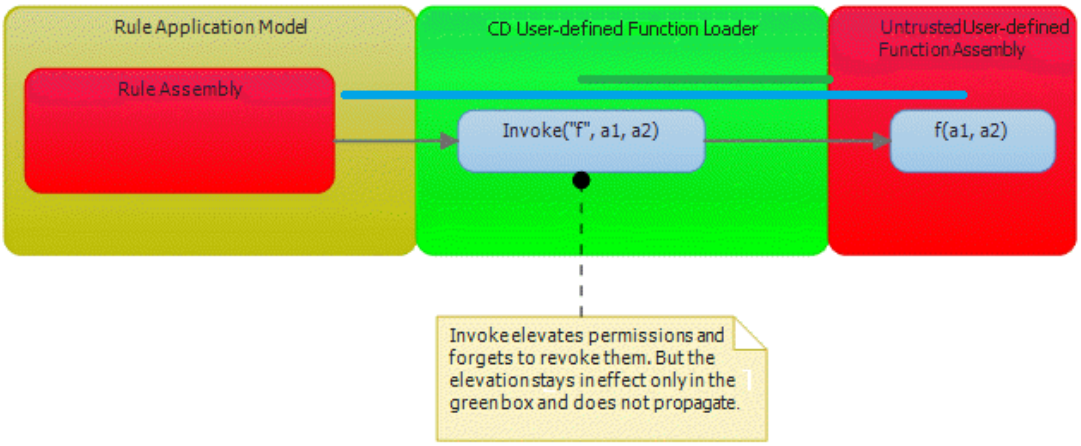
For more information, see:

- General principles
- Guidelines for elevating CAS permissions
- Preventing hijacking of elevated permissions

General principles

This diagram shows the permission flow. The blue line represents a basic set of permissions defined by the sandbox. The green line represents additional permissions added by the Invoke function.

The elevated permissions live only within the green box.



Guidelines for elevating CAS permissions

- Elevate permissions at the lowest level in the code, which is the closest point in the call stack that fails if the permission is not elevated.
- Elevate permissions only in the code specific to rule execution. If elevation is required at a lower level, move up in the call stack to the nearest call that is specific to rules.
- Elevate permissions on methods, and perhaps on classes, but not on the whole assembly.
- Be restrictive and specific. For example, if the code needs to read one registry key, elevate permission only for this key, not the whole registry.
- Use .NET attributes, not procedural code to elevate permissions. .NET attributes limit the scope of elevation to one method call or one class. In procedural code, you can forget to reset elevated permissions, and they can stay on. However, this sets a limit on how restrictive and specific you can be when you elevate permissions on methods or classes. For example, to limit the IO permission to one file, you must specify the complete path, which is known only at runtime, so this is impossible to do with an attribute.
- Add comments to each elevation explaining why it was necessary, and list the use case, unit test, or code path that requires that permission.

Preventing hijacking of elevated permissions

- Use the .NET code transparency model. Keep the amount of Oracle Central Designer code that untrusted code can call to the minimum. In the Oracle Central Designer application, very few assemblies allow partially untrusted callers.
- Elevate permissions only when necessary and only in the code that is rule specific.
- Review the possible code path from untrusted code to the function that elevates permission to see if untrusted code can misuse the function.
- During the review process, watch for too generic or too powerful functions running with elevated permissions.
  - One example of such a function is RunExternalFunction(assemblyName, className, methodName, parameters). It allows untrusted code to execute ANYTHING, and it is executed in full trust.

Preventing access to code outside the sandbox

One of the security exploits listed on the web is the ability of untrusted code to call Assembly.Load(assemblyFullName) to gain access to the code in a different assembly. To address this risk, keep untrusted assemblies in a folder separate from the bin folder of the hosting application. Oracle does this in all rule execution environments.

Preventing untrusted code from leaving the sandbox

See http://www.contextis.com/files/are_you_my_type.pdf, which explains several exploits based on .NET serialization, where an untrusted object can be serialized inside the sandbox, cross AppDomain boundaries, and be de-serialized already in the fully trusted AppDomain.

The rule sandbox does not have a permanent, must-use communication channel with the parent AppDomain, but several things might involve serialization from the sandbox to the parent AppDomain:

- Exceptions thrown from rule execution.
- Results of rule tests returned from the Rule Test Engine, including exceptions if the test failed.
- Calls from a rule into the InForm application through the COM interface.

Sandbox exceptions are caught inside the sandbox, and new exceptions are thrown to the parent AppDomain. The assumption is that all caught exceptions are already logged, and creating new exceptions will not result in the loss of information about errors.

If a test fails and an exception occurs, the exception is rewritten or replaced with a string error message. The Rule Test Harness uses only an exception message.

The Oracle Health Sciences InForm COM interface comes into the main fully trusted AppDomain as .NET COM Interop. This reference then passes into the sandbox. Several methods in this interface have object type parameters, which potentially can allow any object to pass in. Untrusted code can create a serializable object, and pass it into the InForm interface. The object can be de-serialized into the main fully trusted AppDomain.

However, InForm Interop assemblies do not allow partially trusted callers, so untrusted code cannot call them directly. There is no way to pass indirectly arbitrary objects into these calls, because the Oracle Health Sciences InForm application assumes they are integers.

Limitations of the rule sandbox security model

The rule sandbox security model has several limitations:

- The following DoS attacks can occur:
  - Infinite sleep keeping the worker thread locked. Over time, such rules silently take all threads and starve the application.
  - Infinite loop consuming CPU cycles and slowing down the machine.
  - Writing a large amount of data to the event log and slowing down the application.

  - Allocating a large amount of memory.
- Clinical data corruption can use the legitimate interface of the Rule Application Model.
- Due to human error, the incorrect public key can be installed into the certificate store and establish trust for a bad user-defined function assembly.

Oracle-hosted Oracle Central Designer

The Oracle Services team is responsible for reviewing the source code for any user-defined function DLL that requires elevated permissions and that will be used by studies hosted by Oracle, even if the Oracle Central Designer instance is hosted by the customer.

The customer sends the source code to the Oracle Services team, who reviews and approves it, based on the security and coding standards maintained by Oracle.

The Oracle Services team then compiles the DLL and signs it using the Oracle signing tool (the same tool used to sign the Oracle Central Designer binaries). The signed DLL is then provided to the customer.

SQL injections

User-defined function DLLs can call any SQL statements that developers write, so you must adhere to secure SQL coding practices when writing SQL to be used by the user-defined function code.

XML injection

User-defined function DLLs consume any XML that developers need to use, so you must adhere to secure XML coding practices when manipulating XML in the user-defined function code.