OVERVIEW

The Cizer .NET Reporting Security Model is based upon the core security concepts of authentication and authorization. Authentication can be defined as the process used for obtaining a users credentials and validating the user to the application. Authorization is the process by which users are permitted access to resources within the application. Developed on these core concepts, the security module used by this Cizer product provides an administrator with a variety of methods for validating users to an application and for restricting the application resources granted to its users.

AUTHENTICATION

The Cizer security model allows an administrator to choose from three providers for collecting user credentials to identify them in the application. These options are available for selection from the web management interface and can be set at the individual user level.

1. Windows

The Windows security provider works with the existing Windows domain account and password to identify a user in the application. This model requires that the Windows account be added to the Cizer security database through the Cizer web management interface or batch process and be granted application permissions. The Windows account is stored solely for the authorization of user rights with all authentications provided through the Windows domain.

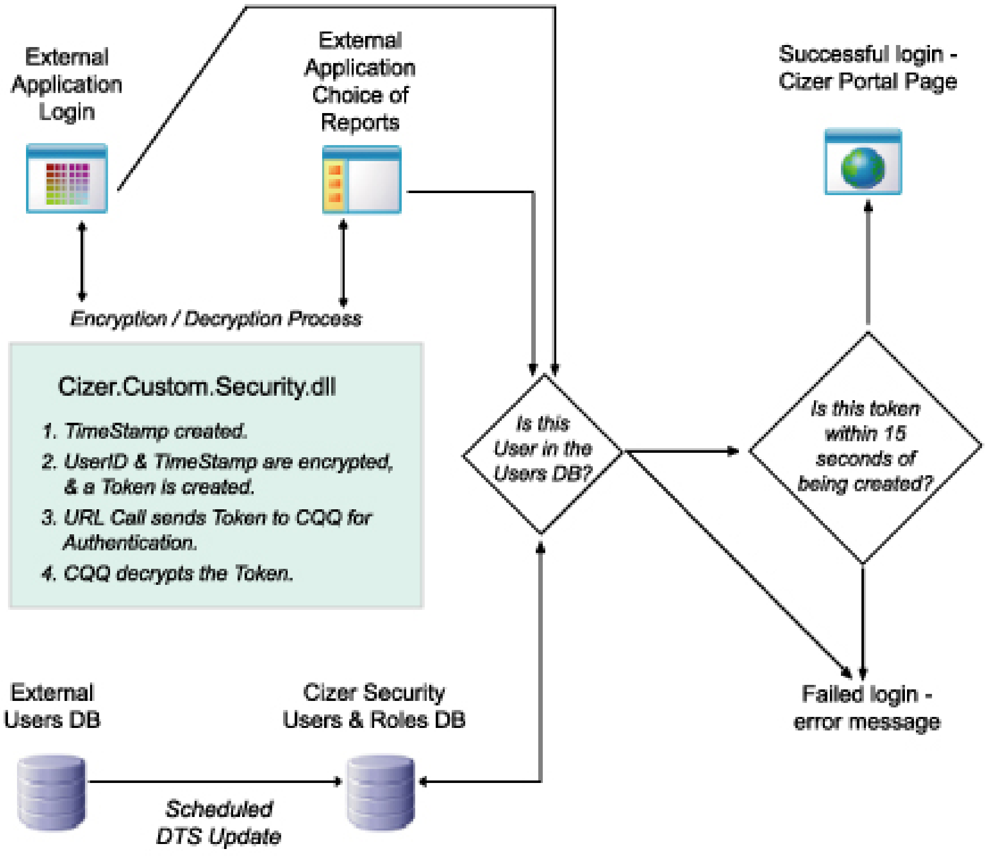
2. Forms

The Forms security provider relies on the Cizer Security database for all account management and authentication. User accounts, passwords, and activation information are stored in the security database and administered through the Cizer web management interface. Account passwords are automatically expired on creation, forcing the user to change their password with first login and automatically setting the next account expiration date. Invalid authentication attempts are logged by the system and the account is expired after reaching an invalid attempt threshold.

3. Custom

The custom security provider removes the authentication process from the Cizer application and requires the client to assume this responsibility. The custom provider model requires that the client application authenticate the user and then request a security token from the Cizer security module. The security module returns an encrypted token comprised of the userid and current timestamp. The client then passes the encrypted security token as a URL call to the Cizer Security Module, at which point the token is decrypted and the expiration timestamp is validated. This model requires that the custom account be added to the Cizer security database through the Cizer web management interface or batch process and be granted application permissions.

Login Security Diagram:

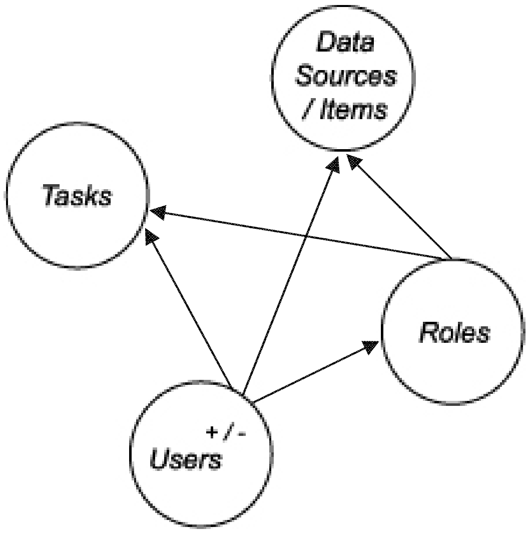

External Application Login
External Application Choice of Reports
Successful login - Cizer Portal Page
Encryption / Decryption Process
Cizer.Custom.Security.dll
1. TimeStamp created.
2. UserID & TimeStamp are encrypted, & a Token is created.
3. URL Call sends Token to CQQ for Authentication.
4. CQQ decrypts the Token.
Is this User in the Users DB?
Is this token within 15 seconds of being created?
External Users DB
Cizer Security Users & Roles DB
Failed login - error message
Scheduled DTS Update

AUTHORIZATION

1. Users and Roles

A highly flexible and secure role-based model is used for authorizing user access to Cizer application resources. The **role** is the highest level of authorization detail in the security model and is comprised of a collection of tasks and items. **Tasks** define a specific action within the application and represent permissions such as publish, save, run, or admin. The **item** defines specific objects in the application such as tables, views, SQL text, parameters, queries, and reports. User permissions are granted at both the role and user level by allowing the administrator to assign users to application roles and override role-defined task and item permissions at the user level. The authorization of application tasks and items plays an integral part in the security process and is the primary reason that clients choosing the Windows or custom security model must add their users to the Cizer security database even while authenticating through a different model.

Users & Roles in Cizer Security Administration:



Roles are given access to a list of Tasks & DataSources/Items.

Tasks = Rights within Cizer.NET.

DataSources = Databases you can see.

*Items = Field Level Security on Items within the Database.**

** Row level security can be acheived thru silent parameterization within the SQL statement*

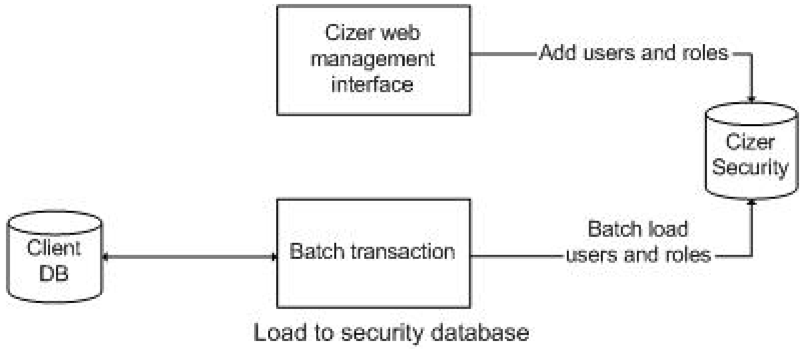
Users are assigned to defined Roles.

You can limit or augment the rights of individual users by removing or adding specific Tasks & DataSources/Items.

You can also define Users without involving any Roles.

2. User accounts from external sources (via DTS)

User accounts may be added to the Cizer security database using the Cizer web management interface or through a batch transaction. The web management interface enables the administrator to add and manage individual user accounts and roles. This module is used for defining the authentication type, assigning roles, and granting permissions to the application tasks and items. The batch transaction method allows an administrator to create a custom module or SQL statement for loading users and roles from a separate application. The batch process defines a security stored procedure that loads credentials to the security database and populates their corresponding roles and permissions.


Cizer web management interface
Add users and roles
Cizer Security
Client DB
Batch transaction
Batch load users and roles
Load to security database

3. Row Level Data Security:

The role-based authorization model also includes the ability to restrict data at the row level, providing an additional layer of information security. The data restriction is achieved by exposing the current user id (%UserID%) as a silent parameter that can be included as a SQL Where clause constraint in a Cizer "User Query" which functions like a view on the Cizer Reporting Services server.

The silent user id (%UserID%) data restriction can also be used to filter the parameter values available to the user, but including it in the parameter SQL Where clause.

This architecture enables an administrator to create cross-reference table that can provide filters to any enterprise data store. For example a cross-reference table for Finance Funds might simply have a column of UserID's and a column of FinanceFunds. Thus the user would only see those Finance Funds corresponding to their user id, and the Finance Funds would be included as a SQL constraint for all database queries.

For the following example, assume the user has logged into Cizer as fsmith or sjones, which is available within Cizer as %UserID%

Table: User_Fund

User	FinanceFund
fsmith	311
fsmith	210
sjones	311
sjones	245
sjones	248

→

Table: Fund_Balance

FinanceFund	FundBalance
210	1000.21
245	3245.76
248	21.55
311	1982.35

Example Parameter SQL to return Funds available to the user:

```
Select FinanceFund from User_Fund where User = %UserID%
```

Example Query SQL to return Balances available to the user:

```
Select * from Fund_Balance join User_Fund on (Fund_Balance.FinanceFund =
User_Fund.FinanceFund) where User_Fund.User = %UserID%
```

SECURITY CONFIGURATION OPTIONS

Option 1:

Mixed Mode Security (Windows Integrated and/or Anonymous access)

The Mixed Mode Security configuration option should be followed when Cizer .NET Reporting is installed on an enterprise network that may be accessed from the web using anonymous access or when impersonation has been disabled on the network.

Summary:

- Cizer or Windows accounts used for authentication
- Cizer security model used for authorization
- Reports are created using Everyone account in Microsoft Reporting Services
- Reports are viewed using Everyone account in Microsoft Reporting Services

Setup:

1. IIS Windows integrated security and Anonymous access enabled
2. Cizer .NET Reporting web.config impersonation set to false
3. Everyone account given Content Manager rights on Cizer Publish and Temp folders in Microsoft Reporting Services

Option 2:

Windows Integrated Security

The Windows Integrated Security configuration option should be followed when Cizer .NET Reporting is installed on an enterprise network that does not allow anonymous access from outside the domain.

Summary:

- Cizer or Windows accounts used for authentication
- Cizer security model used for authorization
- Reports are created using Windows logon credentials in Microsoft Reporting Services
- Reports viewed using Windows logon credentials in Microsoft Reporting Services

Setup:

1. IIS Windows integrated security enabled; Anonymous access disabled
2. Cizer .NET Reporting web.config impersonation set to true
3. Windows User/Role given Content Manager rights or Cizer Publish and Temp folders in Microsoft Reporting Services

CONCLUSION

The security model provides administrators with a scalable and secure environment for integrating Cizer applications with their existing products or deploying as a standalone application. The Windows and Custom providers define a flexible authentication process for gathering user credentials and identifying them to the application. The role-based security can be used to customize the application and restrict user access at the resource level.