



Transmission and Reliability

TRO Application Security

User Guide

Release 0.0.0



TRO Control Systems
January 17, 2007

Table of Contents

| | |
|---|---|
| Administration | 1 |
| The Application Security Administration Tool | 1 |
| Adding an Application to TRO Application Security | 1 |
| Managing Roles | 1 |
| Creating a Role | 1 |
| Assigning Users to a Role | 2 |
| Modifying a Role | 2 |
| Deleting a Role | 2 |
| Managing Users | 3 |
| Adding a User | 3 |
| Adding a TVA Customer (External Company) | 3 |
| Deactivating a User Account | 3 |
| Unlocking a User Account | 3 |
| Removing a User | 4 |
| Responding to Requests for an Account | 4 |
| Responding to Requests for Access | 4 |
| Managing Groups | 4 |
| Adding a Group | 4 |
| Modifying a Group | 5 |
| Deleting a Group | 5 |
| Implementation | 6 |
| Implementing security on Web applications | 6 |
| Implementing security on Windows applications | 6 |
| Simple page-level implementation | 6 |
| Simple control-level implementation | 8 |
| Advanced Implementation | 9 |

Administration

1 The Application Security Administration Tool

The Administrator Interface for the development environment is located at <http://chadesoweb/TroApplicationSecurity/>. Any changes made using this interface will only affect the application in development.

2 Adding an Application to TRO Application Security

The default tab in the Application Security Administration Tool is the Application tab. All applications are added to the TRO Application Security using this tab.

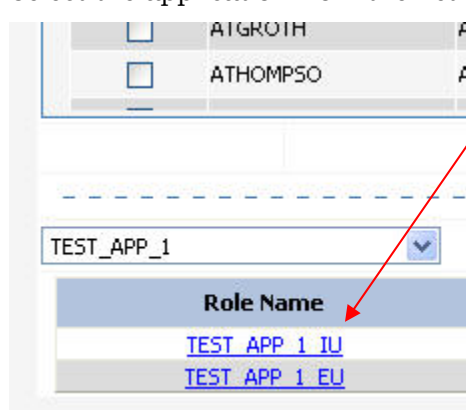
1. Navigate to the Application tab.
2. Enter the Application Name and the Application Description.
3. Click Save.

NOTE: When saving a new application, all spaces in the Application Name are converted to underscores (_), and all characters in the Application Name are changed to UPPER CASE.

3 Managing Roles

3.1 Creating a Role

1. Navigate to the **Roles** tab.
2. Select the application from the list at the bottom of the tab.



The screenshot shows the 'Roles' tab in the Application Security Administration Tool. At the top, there is a list of applications with checkboxes and names: 'ATGROTH' and 'ATHOMPSO'. Below this, there is a search box containing 'TEST_APP_1'. A red arrow points to the 'Role Name' field, which displays 'TEST_APP_1 IU' and 'TEST_APP_1 EU'.

3. Enter the **Role Name** and **Role Description**. To determine if a Role already exists, enter a keyword into the search box and click **Search**. All Roles that contain the keyword in the Role

Name or the Role Description are displayed. Click **Show All** to clear the search box and display all roles for the selected application.

NOTE: Users and Groups can be assigned as part of creating a new Role. Refer to the [Assigning Users to a Role](#) section of this document for detailed instructions.

4. Click **Save**.

NOTE: When saving a new role, all spaces in the Role Name are converted to underscores (_), and all characters in the Role Name are changed to UPPER CASE.

3.2 Assigning Users to a Role

1. Navigate to the **Roles** tab.
2. Navigate to the **Assign Users** tab to assign users to this role. To locate a desired Role, enter a keyword into the search box and click Search. All Roles that contain the keyword in the Role Name or the Role Description are displayed. Click Show All to clear the search box and display all roles for the selected application.
3. Select the check box next to each user to be assigned to this role.
4. Navigate to the **Assign Groups** tab to assign groups of users to this role.

NOTE: Refer to the [Adding a Group](#) section of this document for detailed instructions for creating groups of users.

5. Select the check box next to each group to be assigned to this role.
6. Click **Save**.

3.3 Modifying a Role

1. Navigate to the **Roles** tab.
2. Select the application from the list at the bottom of the tab.
3. Select the Role by clicking on the desired **Role Name** at the bottom of the screen.
4. Modify the **Role Name** or **Role Description**, assign new users or groups, and unassign existing users or groups, as desired.
5. Click **Save**.

3.4 Deleting a Role

1. Navigate to the **Roles** tab.
2. Select the application from the list at the bottom of the tab.
3. Click the **Delete** link to the right of the **Role** that is to be deleted.
4. Click **OK** to confirm the deletion.

4 Managing Users

The TRO Application Security is preloaded with all Transmission and Reliability employee and contractor names. In addition, other names have been added to the security database. Before adding any user, verify that the user does not already exist in the security database.

4.1 Adding a User

All TRO users are included in the security application, retrieved from the roster list.

1. Navigate to the **Users** tab.
2. Select the **Is External** check box. This step is not required for users who have been assigned an NTID.
3. Select the **Company Name** where the user is employed.
4. Enter the **User Name** for the new user. For TVA employees and contractors, enter the user's NTID as the **User Name** and skip to 9.
5. Enter the user's **Password**. This step is not required for users who have been assigned an NTID.
6. Enter the user's **First Name** and **Last Name**. This step is not required for users who have been assigned an NTID.
7. Select a **Security Question** and enter the **Security Answer** for validating requests for password resets. For TVA employees and contractors, the security credentials from their NTID are used.
8. Enter the user's **Email** address and **Telephone** number.
9. To assign this new user to an existing role, click the Expand button (+) to expand the application roles, and then select the check box next to each role to which this user is to be assigned.
10. Click **Save**.

4.2 Adding a TVA Customer (External Company)

1. Navigate to the **Companies** tab.
2. Enter the **Company Name**.
3. Click **Save**.

4.3 Deactivating a User Account

1. Navigate to the **Users** tab.
2. Enter the user's name, or any portion of the name or NTID, in the Search box and click **Search**.
3. Click the **User Name** of the desired user.
4. Select the **Is Locked** check box.
5. Click **Save**.

4.4 Unlocking a User Account

1. Navigate to the **Users** tab.

2. Enter the user's name, or any portion of the name or NTID, in the Search box and click **Search**.
3. Click the **User Name** of the desired user.
4. Clear the **Is Locked** check box.
5. Click **Save**.

4.5 Removing a User

1. Navigate to the **Users** tab.
2. Enter the user's name, or any portion of the name or NTID, in the Search box and click **Search**.
3. Click the Delete link to the right of the **User Name** that is to be deleted.
4. Click **OK** to confirm the deletion.

4.6 Responding to Requests for an Account

1. Navigate to the **Account Requests** tab.
2. Click the **User Name** of the desired user.
3. Select the **Company** for this user, and then click **Approve** to add the user to the security database,

or

Enter the **Reason for Disapproval** and click **Disapprove** to reject the request for an account.

The user will be notified of your action in an email message.

Approving a user's request for an account will automatically generate a request for access. See the [Responding to Requests for Access](#) section of this document for detailed instructions on responding to these requests.

4.7 Responding to Requests for Access

1. Navigate to the **Access Requests** tab.
2. Click the **User Name** of the desired user.
3. Click **Approve** to assign the user to the requested role,

or

Enter the **Reason for Disapproval** and click **Disapprove** to reject the request for access.

The user will be notified of your action in an email message.

5 Managing Groups

5.1 Adding a Group

1. Navigate to the **Groups** tab.

2. Enter the **Group Name** and **Group Description**.

NOTE: When adding a group that is specific to an application, the group name should be prefixed with the application's acronym.

When adding a global group, use standard TVA/TRO names for the group (e.g., Balancing Authority, Market Participants, Fossil Plant Owners).

3. Users can be assigned to the group by selecting the check box to the left of each desired **User Name**.
4. This group can be assigned to roles by clicking the Expand button (+) for each desired **Application Name** and then selecting the check box to the left of each desired **Role Name**.
5. Click **Save**.

NOTE: When saving a new group, all spaces in the Group Name are converted to underscores (_), and all characters in the Group Name are changed to UPPER CASE.

5.2 Modifying a Group

1. Navigate to the **Groups** tab.
2. Enter the desired group's name in the Search box and click **Search**.
3. Click on the desired **Group Name**.
4. To add a user to this group, select the check box to the left of the desired **User Name**.

To remove a user from this group, clear the check box to the left of the desired **User Name**.

5. To assign this group to a new role, click the Expand button (+) for each desired **Application Name** and then select the check box to the left of each desired **Role Name**.

To remove this group from an assigned role, click the Expand button (+) for each desired **Application Name** and then clear the check box to the left of each desired **Role Name**.

6. Click **Save**.

5.3 Deleting a Group

1. Navigate to the **Groups** tab.
2. Enter the desired group's name in the Search box and click **Search**.
3. Click the Delete link to the right of the **Group Name** that is to be deleted.
4. Click **OK** to confirm the deletion.

Implementation

Security can be implemented at page-level and control-level. If you implement security on a Web application that is making use of custom user controls, you may need to add security at a control-level.

6 Implementing security on Web applications

To add security to a web site, add a reference to **Tva.Web.dll**. This will automatically add references to the following DLLs:

- **Tva.Core.dll**
- **Tva.Security.dll**
- **Tva.IO.Compression.dll**

7 Implementing security on Windows applications

To add security to a web site, add references to the following DLLs:

- **Tva.Core.dll**
- **Tva.Security.dll**
- **Tva.IO.Compression.dll**

8 Simple page-level implementation

In most cases, simple page-level implementation will be sufficient for securing a web page. Advanced implementation of security must be used in cases when more granular control is needed over the login process.

Sample page-level implementation code:

```
Imports Tva.Security.Application

Partial Class _Default
    Inherits Tva.Web.UI.SecurePage

    Public Sub New()

        MyBase.New("test_app_1", SecurityServer.Development)

    End Sub

    Protected Sub Page_LoginSuccessful(ByVal sender As Object, _
        ByVal e As System.EventArgs) Handles Me.LoginSuccessful
```

```

        Me.LabelGreeting.Text = String.Format(Me.LabelGreeting.Text,
Me.SecurityProvider.User.FirstName)
        Me.LabelWebSite.Text = String.Format(Me.LabelWebSite.Text,
Me.SecurityProvider.ApplicationName)

        Me.SecurityProvider.SetValidRoles(Me.UltraWebTab1.Tabs(0), "test app 1 iu")
        Me.SecurityProvider.SetValidRoleAction(Me.UltraWebTab1.Tabs(0), ValidRoleAction.Visible)

        Me.SecurityProvider.SetValidRoles(Me.UltraWebTab1.Tabs(1), "test app 1 eu")
        Me.SecurityProvider.SetValidRoleAction(Me.UltraWebTab1.Tabs(1), ValidRoleAction.Visible)

    End Sub

End Class

```

Web pages that need to be secured must inherit from **Tva.Web.UI.SecurePage** and provide a default constructor that calls one of the constructors provided by **Tva.Web.UI.SecurePage**.

```

Public Sub New()

    MyBase.New("test_app_1",
SecurityServer.Development, True)

End Sub

```

SecurePage has multiple constructor overloads that can be used to initialize it. The constructor shown above takes 3 parameters:

1. Parameter 1: Name of application as in the security database
2. Parameter 2: The security server that you want to use
3. Parameter 3: Whether you want to cache user information once initialized

Any web page that is inheriting from **Tva.Web.UI.SecurePage** must at least call the constructor that takes an application name.

Tva.Web.UI.SecurePage inherits from **System.Web.UI.Page** and adds the following:

Properties:

- SecurityProvider – The **Tva.Security.Application.WebSecurityProvider** that handles the security. This property is available at the page's **PreInit** event.

Events:

- LoginSuccessful – Raised when the current user has access to the application
- LoginUnsuccessful – Raised when the current user does not have access to the application

The security provider (i.e.

Tva.Security.Application.WebSecurityProvider) also makes it easy to manipulate controls in a web page based on roles. For example, you can make a control visible or enable it

only when the current user belongs to a specific role with the following 2 lines of code:

```
Me.SecurityProvider.SetValidRoles
    (Me.UltraWebTab1.Tabs(0),
     "test_app_1_iu")
Me.SecurityProvider.SetValidRoleAction
    (Me.UltraWebTab1.Tabs(0), ValidRoleAction.Visible)
```

Here we are telling the security provider that we want to make the control **UltraWebTab1.Tabs(0)** visible only if the current user belongs to the "test_app_1_iu" role. Multiple roles can be specified by delimiting them with a comma (,).

9 Simple control-level implementation

Control-level implementation is similar to that of page-level implementation with a few exceptions:

1. The custom control inherits from **Tva.Web.UI.SecureUserControl**
2. Having a default constructor that calls the **SecureUserControl** constructor is optional if the custom control is being used in a secure page (page that is directly or indirectly inheriting from **Tva.Web.UI.SecurePage**).

Sample control-level implementation code:

```
Imports Tva.Security.Application

Partial Class WebUserControl
    Inherits Tva.Web.UI.SecureUserControl

    ''' <summary>
    ''' This is optional when the custom control is being used in a secure web page.
    ''' </summary>
    Public Sub New()

        MyBase.New("test_app_1", SecurityServer.Development, True)

    End Sub

    Protected Sub Page_LoginSuccessful(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.LoginSuccessful

        Me.SecurityProvider.SetValidRoles(Me.ButtonSave, "test_app_1_iu")
        Me.SecurityProvider.SetValidRoleAction(Me.ButtonSave, ValidRoleAction.Visible)

    End Sub
End Class
```

Tva.Web.UI.SecureUserControl inherits from **System.Web.UI.UserControl** and adds the following:

Properties:

- **SecurityProvider** – The **Tva.Security.Application.WebSecurityProvider** that handles the security. This property is available at the page's **Init** event.

Events:

- **LoginSuccessful** – Raised when the current user has access to the application
- **LoginUnsuccessful** – Raised when the current user does not have access to the application

10 Advanced Implementation

More granular control over the login process can be achieved by subscribing to any of the following events raised by the **Tva.Security.Application.WebSecurityProvider**:

- **BeforeLogin** – Raised before the login process has started.
- **BeforeAuthenticate** – Raised before the user has been authenticated. User property has been initialized at this point.
- **AfterAuthenticate** – Raised when the current user has been authenticated and either one of **AccessGranted** or **AccessDenied** events have been raised.
- **AfterLogin** – Raised when the login process is complete.
- **AccessGranted** – Raised when the current user has been authenticated and is determined to have access to the application.
- **AccessDenied** – Raised when the current user has been authenticated and is determined not to have access to the application.

Sample advanced implementation code:

```
Imports Tva.Security.Application

Partial Class _Default
    Inherits Tva.Web.UI.SecurePage

    Private WithEvents m_securityProvider As WebSecurityProvider

    Public Sub New()

        MyBase.New("test_app_1", SecurityServer.Development, True)
        m_securityProvider = Me.SecurityProvider

    End Sub

    ...

    Protected Sub m_securityProvider_AccessDenied(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles m_securityProvider.AccessDenied
```

```
e.Cancel = True
Response.Redirect("CustomAccessDeniedPage.aspx")

End Sub

End Class
```

In order to subscribe to the events that are raised by the security provider we must declare a member variable with the **WithEvents** of type **Tva.Security.Application.WebSecurityProvider** and assign to the **SecurityProvider** property exposed by **Tva.Web.UI.SecurePage**. Once this is done, we can subscribe to any of the event described above. In the code sample above, we have subscribed to the **AccessDenied** event and want to stop further execution of the login process when the current user does not have access to the application and redirect to our custom “access denied” web page instead of the default access denied web page.

NOTE: Execution of the login process can be stopped from any of the events that provide **System.ComponentModel.CancelEventArgs** as the event argument.