Microsoft Conditional Access Custom Controls integration guide

Overview

Microsoft has a feature in their Azure stack called Conditional Access. This feature allows Azure customers to apply policies to either the log-in process to Office 365 or specific apps and tiles within Office 365/Azure. Using this feature, Azure customers can restrict access to applications, such as Outlook, SharePoint, and others, based on several different factors.

Recently, Microsoft added a function to Conditional Access called custom controls. Custom controls allow third-party integration into Conditional Access. This process involves having a registered application by the third party to be allowed globally by Microsoft and then providing OpenID Connect (OIDC) endpoints for use by the Azure customer to call out to the third party’s authorization process.

Prerequisites

You must ensure that you have the following items:

- Install a SecureAuth IdP appliance version 9.1 or 9.2 and configure one or more realms for that appliance (refer to the SecureAuth IdP Realm Guide).
- Configure the following tabs in the Secure Auth IdP Web Admin console before configuring any other tabs:
  - **Overview**: Define the description of the realm and SMTP connections.
  - **Data**: An enterprise directory must be integrated with SecureAuth IdP.
  - **Workflow**: Define how users access the target.
  - **Multi-Factor Methods**: Define the Multi-Factor Authentication methods that are used to access the target, if any.
- Gain administrative access for Microsoft Azure.
- Install and configure Internet Information Services (IIS) for Windows Server.
- Set up Modern Authentication in your server environment. See the Hybrid Modern Authentication overview and prerequisites for using it with on-premises Skype for Business and Exchange servers article on the Microsoft website.
- Contact support@secureauth.com, open a support ticket, and include "Tailoring - Conditional Access" if you will use this integration. Request the following items so you have them on hand during the configuration:
  - ASPX and code-behind pages (you will need this in the Configure Internet Information Server (IIS) for Windows Server task)
  - Import Rules (you will need this in the Configure Internet Information Server (IIS) for Windows Server task)
  - JSON file (you will need to provide the Client ID, to be created on the Post Authentication tab > OpenID Connect/OAuth 2.0 – Clients section) and the public URL of the OpenID Connect Post-Auth realm to SecureAuth Support
  - JSON file (you will need this for the task in Configure Microsoft Custom Control task)

SecureAuth IdP configuration steps

Create a SecureAuth IdP realm and configure it for use with Microsoft Conditional Access.

Configure Internet Information Server (IIS) for Windows Server

1. Log into your **SecureAuth IdP Admin** console.

2. Copy the ASPX and code-behind pages under the root of the newly-defined realm, which is located in D:\SecureAuth\SecureAuth\Realm_number, for example, D:\SecureAuth\SecureAuth5 (Contact SecureAuth Support per the Prerequisites steps, if you did not already request the ASPX and code-behind pages.)

A custom pre-authentication page is used to retrieve the user ID from Microsoft for this request. Microsoft sends a HTTP POST with the OIDC parameters and an additional parameter called id_token_hint. This parameter includes a JSON web token (JWT) and a number of claims, including the unique ID for the user and their user principal name (UPN). SecureAuth IdP must obtain that information and validate the JWT.

3. Using the IIS Manager, create an inbound rule for Conditional Access in this new realm by completing the following steps:
a. Start the IIS Manager: Open Run, type inetmgr, and click Enter.
b. In IIS, select the Default Web Site.
c. Under Features View, click URL Rewrite.
d. Set an inbound rewrite rule under the realm folder (for example, SecureAuth3).

![Edit Inbound Rule](image)

The URL rewrite rule, shown in the following image, captures requests and places them on the custom page to decode the JWT that Microsoft sends over VIA POST.
4. Using the IIS Manager, change the query string setting for the SecureAuth realm number (for example, SecureAuth3).
   a. In the IIS Manager, focus on the appropriate realm.
   b. Right-click Request Filtering and select Open Feature.
   c. Select the Query Strings tab.
   d. On the right side of the page, click Edit Feature Settings.
   e. Set Maximum URL length (Bytes) to 6144.
   f. Set Maximum query string (Bytes) to 4096.

For more information about the URL rewrite rule, see the Creating Rewrite Rules for the URL Rewrite Module article, on the Microsoft website.
g. Click OK to save the changes.

Data tab settings

1. Select the Data tab.

2. Create a connection based on the data store type, such as Active Directory or SQL Server.

   a. In the Profile Fields section, set the following auxiliary values:

   - Aux ID 1 – userPrincipalName
   - Aux ID 2 – otherLoginWorkstations
   - Aux ID 5 – otherIpPhone and make it writable. (This field is set from custom pre-authentication page – MSConditionalAccess.aspx.vb)
b. In the **Global Aux Fields** section, designate Global Aux ID 1 as **Validated**.

---

**Workflow tab settings**

Select the **Workflow** tab.

1. In the **Login Screen Options** section, set the following values:
   - Set Default Workflow to **Username | Second Factor**.
   - Set Public/Private Mode to **Public Mode Only**.
2. In the **Customer Identity Consumer** section, set the following values:

- Set **Receive Token** to **Token**.
- Leave other fields set to the default.

**Multi-Factor Methods tab settings**
Select the **Multi-Factor Methods** tab.

1. In the **Phone Settings** section, configure the Multi-Factor Authentication methods that you want enabled. The following example shows how to set the email and text (SMS) methods.
   - Set **Phone Field 1** to One-Time Passcode via Phone Call and SMS.
   - Set **Phone Field 2** to One-Time Passcode via Phone Call and SMS.

2. In the **Email Settings** section, set **Email Field 1** to One-Time Passcode via HTML Email.

---

### Multi-Factor Configuration

#### Phone Settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone Field 1</td>
<td>One-Time Passcode via Phone Call and SMS</td>
</tr>
<tr>
<td>Phone Field 2</td>
<td>One-Time Passcode via Phone Call and SMS</td>
</tr>
<tr>
<td>Phone Field 3</td>
<td>Disabled</td>
</tr>
<tr>
<td>Phone Field 4</td>
<td>Disabled</td>
</tr>
<tr>
<td>Phone/SMS Selected</td>
<td>Voice</td>
</tr>
<tr>
<td>Phone/SMS Visible</td>
<td>True</td>
</tr>
<tr>
<td>Default Phone Country Code</td>
<td></td>
</tr>
<tr>
<td>Phone Mask (RegEx):</td>
<td></td>
</tr>
</tbody>
</table>

#### Email Settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email Field 1</td>
<td>One-Time Passcode via HTML Email</td>
</tr>
<tr>
<td>Email Field 2</td>
<td>Disabled</td>
</tr>
<tr>
<td>Email Field 3</td>
<td>Disabled</td>
</tr>
<tr>
<td>Email Field 4</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

### Post Authentication tab settings

Select the **Post Authentication** tab.

1. In the **Post Authentication** section, set the Authenticated User Redirect dropdown to **OpenID Connect/OAuth2**.

2. In the **User ID Mapping** section, set the following values:
   - Set **User ID Mapping** to **Authenticated User ID**.
   - Set **Name ID Format** to **urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified**.
   - Set **Encode to Base64** to **False**.
3. In the **OpenID Connect/OAuth 2.0 – Settings** section, set the following values:

- Set **Enabled** to **True**.
- Set **Issuer** to the fully qualified domain name (FQDN)/Hostname of the IdP appliance, for example, idp.company.com. This must be publicly facing and have a valid SSL certificate.
- Set **Signing Algorithm** to either **RSA SHA256 (RS256)** or **HMAC SHA256 (HS256)**.
  - **RSA SHA256 (RS256)** is an asymmetric algorithm, which means it uses a public/private key pair. SecureAuth uses the private key for signing and provides you with the public key to use to validate the signature.
  - **HMAC SHA256 (HS256)** is a symmetric algorithm, which means one secret key is shared between SecureAuth and the end-user. The same key is used to create the signature and to validate it. This key must be kept secret at all times.
- Set **Signing Cert** to any certificate that is a private key readable by SecureAuth IdP. Do not use wild cards in a certificate.
- Set **Auto Accept User Consent** to **True** to provide a clean user experience.
- Set **Enable User Consent Storage** to **True** to provide a clean user experience and to enable check session endpoints.
- Set **Consent Storage Attribute** to the **AUX ID 2** value that was mapped to a string attribute, for example, **otherLoginWorkstations**.

Leave the following fields set to the default:

- **Authorization Code Lifetime**
- **Access Token Lifetime**
- **Refresh Token Lifetime**
4. In the OpenID Connect/OAuth 2.0 – Scopes section, set the Discoverable check box for the openid scope.
5. In the **OpenID Connect/OAuth 2.0 – Clients** section, click the **Add Client** button and set the following values:

- Set **Name** to **ConditionalAccess** or another appropriate name.
- Set **Client ID** to the appropriate client ID for this client.
- Set the **Enabled/Disabled** check box.

6. In the **OpenID Connect/OAuth 2.0 - Client Details** section, set the following values:

- Set **Enabled** to **True**.
- Set **Name** to **ConditionalAccess** or another appropriate name.
- Set **JSON Web Encryption** to **Disabled**.
- Set **JSON Web Key URI** to **Blank**.
7. In the **Allowed Flows** section, set the following values:
   - Set **Authorization Code** to True.
   - Set **Implicit** to True.
   - Set **Hybrid** to False.
   - Set **Client Credentials** to False.
   - Set **Resource Owner** to False.
   - Set **Refresh Token** to True.
   - Set **Introspection** to True.
   - Set **Revocation** to True.

8. In the **OpenID Connect/OAuth 2.0 - Client Redirect URIs** section, click the **Add Redirect URI** button and set the Client Redirect URI to https://login.microsoftonline.com/common/federation/OAuth2ClaimsProvider
9. In the OpenID Connect/OAuth 2.0 – Claims section, set the following values:
   - Set **Sub** to the AUX ID field assigned the **userPrincipalName** value that was set on the **Data** tab, where **Aux ID 5**, where AuxID5 is set to **otherIPPhone**.
   - Select the **Discoverable** check box.

10. In the OpenID Connect/OAuth 2.0 – Custom Claims section, click the **Add Custom Claim** button and set the following values:
   - Set **Claim** to **SecureAuthMFA**.
   - Set **Profile Property** to **Global Aux ID 1**.
   - Set the **Discoverable** check box.
System Info tab settings

Select the System Info tab.

1. In the Links section at the bottom of the screen, click Click to edit Web Config file to edit the web.config file.

2. Add the following key under the <appSettings> section:

   <add key="MSConditionalAccess-ProfileField" value="AuxID5" />

   For information about editing the web.config file, see the System Info Tab Configuration document.

Save all changes made to this configuration and exit.

Configure Microsoft Custom Control

Create and configure a new custom control for Microsoft Conditional Access.

1. Log in to Microsoft Azure.
2. Click Azure Active Directory in the left pane.
3. In the Security section, click Conditional access.
4. In the Manage section, click Custom controls.
5. Click New custom control.
6. Enter the JSON for customized controls in the fill-in field.
Enter the JSON provided by SecureAuth Support, then click **Save**. (Contact SecureAuth Support per the Prerequisites steps, if you did not already request this information.)
Configure the JSON file as follows, using the above image as a guide:

a. Set `AppId` to the data application referenced by Microsoft.
b. Set `ClientId` by retrieving the designated realm located under the Post Authentication tab, in the OpenID Connect/OAuth 2.0 - Clients section.
c. Set `DiscoveryUrl` to the OpenID configuration for the designated realm.

For your convenience, copy the following code snippet into the JSON file and change values appropriately:

```json
{
    "Name": "Name for SecureAuth MFA",
    "AppId": "Microsoft data App ID",
    "ClientId": "SecureAuth ClientID",
    "DiscoveryUrl": "https://SecureAuthURL/secureauthXX/.well-known/openid-configuration",
    "Controls":
    [
        {
            "Id": "SecureAuthIdP",
            "Name": "SecureAuthIdP",
            "ClaimsRequested":
            [
                {
                    "Type": "SecureAuthMFA",
                    "Value": "Validated",
                    "Values": null
                }
            ]
        }
    ]
}
```

Create a Policy

Create a Microsoft Conditional Access policy.

1. Log into Microsoft Azure.
2. Click Azure Active Directory in left pane.
4. Specify the users, apps, and controls that you want to assign the policy to.
Test Microsoft Conditional Access with SecureAuth IdP

Test that Microsoft Conditional Access works with SecureAuth IdP. In this scenario, you will test with Microsoft Teams, but you could also test with Outlook or Skype for Business.

1. Log in to Microsoft Teams: https://teams.microsoft.com

5. Save your changes.
2. Enter your email address in the following screen:

![Microsoft Sign in page](image)

3. Enter your password in the following screen:

![Microsoft Sign in page](image)
4. Select the kind of two-factor authentication method to use to log into Microsoft Teams. The following example shows the text message (SMS) method.

5. In the following screen, enter the one-time passcode that was sent to you:

6. The following Microsoft Teams screen is displayed if the configuration between Microsoft Conditional Access and SecureAuth IdP is successful.

If you do not see this screen or if you receive an error message, contact SecureAuth Support.