Configure Single Sign-On

This chapter describes how to configure single sign-on for Cisco SD-WAN. Cisco SD-WAN supports single sign-on using Okta or Active Directory Federation Services (ADFS).

Configure Single Sign-On using Okta

Okta provides a secure identity management service that lets you connect any person with any application on any device using Single Sign-On (SSO).

Perform the following steps to configure SSO.

Enable an Identity Provider in vManage

To configure Okta SSO, you must use vManage to enable an identity provider and generate a SAML metadata file:

1. In vManage, click **Administration** > **Settings** > **Identify Provider Settings** > **Edit**.
2. Click **Enabled**.
3. Click **Click here to download the SAML metadata** and save the content in a file. This data will be used for configuring Okta.
4. In the metadata, note the following information that you will use to configure Okta with vManage:
    - Entity ID
    - Signing certificate
    - Encryption certificate
    - Logout URL
    - Login URL

Configure SSO on the Okta Website

To configure SSO on the Okta website:

1. Log on to the Okta website.
2. Create a username using your email address.
3. To add vManage as one SSO application, click on the **Admin** button on the upper right corner to go to the next page. Then check the upper left corner to make sure it shows the **Classic UI** view on Okta. If it shows the **Developer Console**, click on the down triangle to select the **Classic UI**.
4. Click on **Add Application** under **Shortcuts** to the right to go to the next page, and then click on **Create New Application** on the pop-up window. Select **Web** for the platform, and select **SAML 2.0** as the **Sign on Method**. Click **Create**.
5. Give a string as **Application name**.
6. Optional: Upload a logo, and then click **Next**.
7. On **SAML Settings** for **Single sign on URL** section, set the value to the **samlLoginResponse** URL from the downloaded metadata from the vManage UI. Check the box **Use this for Recipient URL and Destination URL**.
8. Copy the **entityID** string and paste it in the **Audience URI (SP Entity ID)** field. The value can be an IP address or the name of the vManage site.
9. For **Default RelayState**, leave empty.
10. For **Name ID format**, select **EmailAddress**.
11. For **Application username**, select **Okta username**.
12. For **Show Advanced Settings**, enter the fields as indicated below.

Table 1:

Component	Value	Configuration
Response	Signed	
Assertion Signature	Signed	
Signature Algorithm	RSA-SHA256	
Digest Algorithm	SHA256	
Assertion Encryption	Encrypted	
Encryption Algorithm	AES256-CBC	
Key Transport Algorithm	RSA-OAEP	

Component	Value	Configuration
Encryption Certificate		**a.** Copy the encryption certificate from the metadata you downloaded. **b.** Go to www.samltool.com and click on **X.509 CERTS**, paste there. Click **Format X.509 Certificate**. **c.** Make sure to remove the last empty line and then save the output (**X.509.cert with header**) into a text file **encryption.cer**. **d.** Upload the file. Mozilla Firefox may not allow you to do the upload. Instead, you can use Google Chrome. You should see the certificate information after uploading to Okta.
Enable Single Logout		Make sure this is checked.
Single Logout URL		Get from the metadata.
SP Issuer		Use the entityID from the metadata.
Signature Certificate		**a.** Obtain from the metadata. Format the signature certificate using www.samltool.com as done above. **b.** Save to a file, for example, **signing.cer** and upload.
Authentication context class	X.509 Certificate	
Honor Force Authentication	Yes	
SAML issuer ID string	SAML issuer ID string	
Attribute Statements (optional)	Field: **Name**	Value: *Username*
	Field: **Name format (optional)**	Value: Unspecified
	Field: **Value**	Value: *user.login*
Group Attribute Statements (optional)	Field: **Name**	Value: Groups
	Field: **Name format (optional)**	Value: Unspecified
	Field: **Matches regex**	Value: **.***

Note It is mandatory to use the two strings, Username and Groups, exactly as shown above. Otherwise, you may be logged in with the default group of Basic.

13. Click **Next**.
14. For **Application Type**, check **This is an internal app that we have created** (optional).
15. Click **Finish**. This brings you to the Okta application page.
16. Click on **View Setup Instructions**.
17. Copy the IDP metadata.
18. In the vManage UI, paste the IDP metadata in vManage using **Identity Provider Settings > Upload Identity Provider Metadata**, and click **Save**.

Assign Users to the Application

To assign users to the application on the Okta website:

1. On the Okta application page, navigate to **Assignments** > **People** > **Assign**.
2. Select **Assign to people** from the drop-down menu.
3. Click on **Assign** next to the user(s) you selected and click **Done**.
4. To add a user, click on **Directory** > **Add Person** > **Save**.

Configure SSO for Active Directory Federation Services (ADFS)

Describes how to use vManage and ADFS to configure Single Sign On (SSO).

The configuration of vManage to use ADFS as IDP involved two steps:

- Step 1 - Import ADFS metadata to vManage
- Step 2- Export vManage metadata to ADFS

Step 2 can be further divided into:

- Edit and then import vManage metadata to ADFS
- Setup ADFS manually using the information from vManage metadata

Import Metadata File into ADFS

Step 1 - Import ADFS metadata to vManage:

1. Download the ADFS Metadata file, typically from the ADFS URL: `https://<your ADFS FQDN or IP>/FederationMetadata/2007-06/FederationMetadata.xml`

2. Save the file as **adfs_metadata.txt**.
3. On the vManage navigate to **Admin > Settings > Identify Provider Settings >Enable**, and then upload **adfs_metadata.txt** to vManage.

   Step 2 - Export vManage metadata to ADFS:

4. With **Identify Provider Settings**enabled, **Click here** to download SAML metadata and save into a file, which is typically `192.168.1.15_saml_metadata.xml`.
5. Edit vManage Metadata file by deleting everything from **<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">** to **</ds:Signature>**.
6. Edit vManage Metadata file by deleting everything from **<md:KeyDescriptor use="encryption">** to **</md:KeyDescriptor>**.
7. Import the new modified vManage Metadata file into ADFS, and enter the **entityID** as **`Display Name`**.
8. Click **Next** until the end.
9. Open **Edit Claim Rule**, and add the following four new custom rules in the exact sequence:

   ```
   @RuleName = "sAMAccountName as Username" c:[Type ==
   "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
   Issuer == "AD AUTHORITY"]=> issue(store = "Active Directory", types
   = ("Username"), query = ";sAMAccountName;{0}", param = c.Value);
   ```

   ```
   @RuleName = "sAMAccountName as NameID" c:[Type ==
   "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
   Issuer == "AD AUTHORITY"] => issue(store = "Active Directory", types
   =
   ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier"),
   query = ";sAMAccountName;{0}", param = c.Value);
   ```

   ```
   @RuleName = "Get User Groups and save in temp/variable" c:[Type ==
   "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
   Issuer == "AD AUTHORITY"] => add(store = "Active Directory", types =
   ("http://temp/variable1"), query = ";tokenGroups;{0}", param =
   c.Value);
   ```

   ```
   @RuleName = "Parse temp/variable1 and Send Groups Membership" c:[Type
   == "http://temp/variable1", Value =~ "(?i)^SSO-"] => issue(Type =
   "Groups", Value = RegExReplace(c.Value, "SSO-", ""));
   ```

10. Verify the final result.
11. In the Active Directory, create the following two security groups: **`SSO-Netadmin`** and **`SSO-Operator`**.

Note If you are using different naming convention for the two security groups, then you have to modify the Regular expression value **`"(?i)^SSO-"`** in the step above.

Any active directory users who are not members of the two groups will only have **Basic** access to vManage.

Add ADFS Relying Party Trust

Before you begin

To add ADFS relying party trust using vManage:

1. Navigate to **Admin > Settings > Identify Provider Settings > Enable**.
2. Download the ADFS Metadata file, and upload it into **vManage**. An example of a URL, **`https://<your ADFS FQDN or IP>/FederationMetadata/2007-06/FederationMetadata.xml`**.
3. **Click here** to download SAML metadata, and save into a file. An example of a saved file, 192.168.1.15_saml_metadata.xml.
4. Open the file with an XML editor, and check that the following information is available:
   - Entity ID
   - Signing certificate
   - Login URL
   - Logout URL
5. Navigate to **`https://www.samltool.com/format_x509cert.php`**.
6. For **Signing certificate**, copy Signing certificate from "metadata" [everything between <ds:X509Certificate> and </ds:X509Certificate>].
7. Navigate to **`www.samltool.com`** page, click **X.509 CERTS > Format X.509 Certificate**, and paste the copied content
8. Save the output ("X.509 cert with header") into a text file "Signing.cer". Remember to remove the last empty line.

Add ADFS Relying Party Trust Manually

To add ADFS relying party trust manually:

1. Launch **AD FS 2.0 Management**.
2. Navigate to **Trust Relationships > Relying Party Trusts**.
3. Click **Action > Add Relying Party Trust**.
4. Click **Start**.
5. Select **Enter data about the relying party manually**, and click **Next**.
6. Enter **Display name** and **Notes**, and then click **Next**.
7. Select **AD FS 2.0 profile**, and click **Next**.
8. Click **Next** to skip **Configure Certificate** page.
9. Click **Enable support for the SAML 2.0 Webs So protocol**.
10. Open a text editor, and open **`10.10.10.15_saml_metadata.xml`** file.

11. Copy the vale of the **Location** attribute for **AssertionConsumerService**, and paste it into the **Relying party SAML 2.0 SSO service URL** text box.
12. Click **Next**.
13. Copy the value of **entityID** attribute, and paste it into the **Relying party trust identifiers** text box.
14. Click **Add**, and click **Next**.
15. Click **Next** to skip **Configure Multi-factor Authentication Now** section.
16. Select **Permit all users to access this relying party**, and click **Next**.
17. Click **Next** to skip **Ready to Add Trust** section.
18. Click **Close**.
19. Open **Edit Claim Rules** window, and add the following four new custom rules in this order:
    - `@RuleName = "sAMAccountName as Username" c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"]=> issue(store = "Active Directory", types = ("Username"), query = ";sAMAccountName;{0}", param = c.Value);`
    - `@RuleName = "sAMAccountName as NameID" c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"] => issue(store = "Active Directory", types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier"), query = ";sAMAccountName;{0}", param = c.Value);`
    - `@RuleName = "Get User Groups and save in temp/variable" c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"] => add(store = "Active Directory", types = ("http://temp/variable1"), query = ";tokenGroups;{0}", param = c.Value);`
    - `@RuleName = "Parse temp/variable1 and Send Groups Membership" c:[Type == "http://temp/variable1", Value =~ "(?i)^SSO-"]=> issue(Type = "Groups", Value = RegExReplace(c.Value, "SSO-", ""));`
20. Open the **Edit Claim Rules** window, and verify that the rules display in the **Assurance Transform Rules** tab.
21. Click **Finish**.
22. Open **Properties** window of the newly created **Relying Party Trust**, and click the **Signature** tab.
23. Click **Add**, and add the **Signing.cer** created in **Step 6**.
24. In the **Active Directory**, click the **General** tab, and enter the following two security groups in the **Group name** text box:

    `SSO-Netadmin`

    `SSO-Operator`

Note If you use different naming convention for the two security groups, then you have to modify the **Regular** expression value for `(?i)^SSO-` mentioned in **Step 19**.

Note Any active directory user who is NOT a member of these two groups, will only have **Basic** access to vManage.