Authentication: Overview

Authentication is the process of verifying the identity of a user that is attempting to log on to or access software. In SAS Viya, authentication options vary, based on which interface is being used in your environment:

- In a visual-only deployment, an LDAP provider is needed to verify the user and group information. Users can be authenticated through SAS Logon Manager, using an LDAP provider, Integrated Windows Authentication (IWA), Security Assertion Markup Language (SAML), or OAuth and OpenID Connect.

- In a full deployment, the pluggable authentication module (PAM) validates the user’s credentials when accessing SAS Studio and CAS Server Monitor. In addition, an LDAP provider authenticates the user and verifies the user and group information. Users can be authenticated through SAS Logon Manager, using an LDAP provider, Integrated Windows Authentication (IWA), Security Assertion Markup Language (SAML), or OAuth and OpenID Connect.

- In a programming-only deployment, the supported authentication mechanism is PAM.

Authentication: How To

Authentication Mechanisms

Overview

Authentication mechanisms integrate SAS into your computing environment. External mechanisms include direct LDAP authentication (which is referred to as LDAP in this documentation), host authentication, Integrated Windows Authentication (IWA), Security Assertion Markup Language (SAML), and OAuth and OpenID Connect. Pluggable authentication modules (PAM) extend UNIX host authentication.

The following sections are listed alphabetically. Configure the authentication mechanism that is appropriate for your environment. For more information, see Authentication Mechanisms on page 15.
Configure IWA

Verify IWA Prerequisites

Before configuring Integrated Windows Authentication (IWA), make sure that the following components exist:

Note: These prerequisite components are usually configured by the Active Directory administrator.

1. A service account exists in Active Directory.

2. A service principal name (SPN) is mapped to the service account:
   a. Verify that there is a mapping already configured:

   ```sh
   setspn -F -Q HTTP/hostname.example.com
   ```

   **Sample SPN Query**

   ```
   CN=user-logon-name,OU=Service Accounts,OU=Domain Controllers,OU=Servers,DC=EXAMPLE,DC=com
   HTTP/hostname.example.com
   HTTP/HOSTNAME
   
   Existing SPN found!
   ```

   If an SPN is not found, then contact your information technology support group for assistance with registering the machine.

   b. Verify that the service is linked to the service account:

   ```sh
   setspn -L user-logon-name
   ```

   **Sample Account Query**

   ```
   Registered ServicePrincipalNames for CN=user-logon-name,OU=Service Accounts,OU=Domain Controllers,OU=Servers,DC=EXAMPLE,DC=com:
   HTTP/<hostname>@<example>.com
   HTTP/<hostname>
   
   The value for user-logon-name is the same one identified in the common name (CN) from the previous command output, or as the sAMAccountName on the service account in Active Directory.
   ```

3. A keytab file has been generated by issuing the following command:

   ```sh
   ktutil
   rkt path-to/hostname.keytab
   list -e
   ```

   ```
   slot KVNO Principal
   ---- ---- ------------------------------------------------------------
   1    3    HTTP/<hostname>@<example>.com (arcfour-hmac)
   ```

   For more information about the ktutil command, see the vendor documentation.

How to Configure IWA

Note: This information does not apply to a programming-only deployment.

IWA is a Windows technology that uses Kerberos authentication. To configure IWA, complete the following:
1 Change the permissions, owner, and group on the file by running the following command:

   chmod 600 keytab-filename
   chown sas keytab-filename
   chgrp sas keytab-filename

2 Verify that the SPN is mapped to the principal name.

3 Using the sasboot user, add your user ID or an Active Directory group that contains the environment administrators, as a member of the SAS Administrators group. Then, log off from SAS Environment Manager. For more information, see "Add or Remove Custom Group Members" in SAS Viya Administration: Identity Management.

   **CAUTION! You must specify your personal user ID.** Your user ID must be in your specified LDAP provider. It must match the user ID that you use to log on to your system.

4 Configure the Kerberos authentication properties.
   a Log on to SAS Environment Manager, using your user ID or the sasboot user.
   b Navigate to the SAS Logon Manager configuration definitions. For more information, see Edit Authentication Configuration Instances on page 8.
   c In the Definitions list, select sas.logon.kerberos.
   d In the top right corner of the window, click .
   e In the New sas.logon.kerberos Configuration dialog box, enter the values for the following fields, based on your environment:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug</td>
<td>On</td>
</tr>
<tr>
<td>keyTabLocation</td>
<td>file://location-of-keytab</td>
</tr>
<tr>
<td>servicePrincipal</td>
<td>principal-name-from-keytab</td>
</tr>
<tr>
<td>stripRealmForGss</td>
<td>On</td>
</tr>
</tbody>
</table>

   **Note:** Contact your administrator for the keytab location and the host name of the service principal.
   f Click **Save**.

5 Add Kerberos to the active profile.
   a In the navigation pane, switch to the All services list and select SAS Logon Manager.
   b In the spring instance, click .
   c In the Edit spring Configuration dialog box, add kerberos to the profiles.active field.
      The following value should be specified for the profiles.active field:
      
      ldap,postgresql,kerberos
   d Click **Save**.

6 Restart the SAS Logon Manager service by running the following command:
sudo service sas-viya-saslogon-default restart

Note: It might take several minutes to restart SAS Logon Manager.

Configure the Microsoft Internet Explorer to Use SPNEGO

Configure Security Settings
2. Select Local intranet and then click Sites.
3. Configure the intranet domain settings:
   a. Verify that the check boxes for the following items are selected:
      - Include all local (Intranet) sites not listed in other zones
      - Include all sites that bypass the proxy server
   b. Click Advanced and add your domain name to the Websites list to ensure that Internet Explorer recognizes any site with your domain name as the intranet.
4. Configure intranet authentication:
   a. In the Security level for this zone area, click Custom level.
   b. Scroll to the User Authentication section, select Automatic Logon only in Intranet Zone, and click OK.

Configure Connection Settings
1. Select Tools ⇒ Internet options ⇒ Connections.
2. Click LAN settings.
3. Verify that the proxy server address and port number are correct.
4. Click Advanced.
5. Verify that the correct domain names are entered in the Exceptions field on the Proxy Settings dialog box.

Configure Advanced Settings
1. Select Tools ⇒ Internet options ⇒ Advanced.
2. Scroll to the Security section, and verify that Enable Integrated Windows Authentication is selected.
3. Click OK and restart the browser to activate the changes.

Configure the Mozilla Firefox to Use SPNEGO
1. From a browser window, navigate to about:config.
2. Click I’ll be careful, I promise! to accept the security warning.
3. In the Search field, enter network.negotiate.
4. Double-click the network.negotiate-auth.trusted-uris Preference Name, enter http://hostname.example.com, in the Enter string value field, and then click OK.
   Note: The values in the Enter string value field are comma-separated.
Configure the Google Chrome to Use SPNEGO

**Configure Security Settings**

1. Click the Chrome menu key on the browser toolbar, and then select **Settings**.
2. Select **Show advanced settings**.
3. Scroll to the **Network** section, and click **Change proxy settings**.
4. In the Internet Properties dialog box, select **Security**.
5. Select **Local intranet**, and then click **Sites**.
6. Configure the intranet domain settings:
   a. Verify that the check boxes for the following items are selected:
      - Include all local (Intranet) sites not listed in other zones
      - Include all sites that bypass the proxy server
   b. Click **Advanced** and add your domain name to the **Websites** list to ensure that Internet Explorer recognizes any site with your domain name as the intranet.
7. Configure intranet authentication:
   a. In the **Security level for this zone** area, click **Custom level**.
   b. Scroll to the **User Authentication** section, select **Automatic Logon only in Intranet Zone**, and click **OK**.

**Configure Connection Settings**

1. In the Internet Properties dialog box, select **Connections**.
2. Click **LAN settings**.
3. Verify that the proxy server address and port number are correct.
4. Click **Advanced**.
5. Verify that the correct domain names are entered in the **Exceptions** field on the Proxy Settings dialog box.

**Configure Advanced Settings**

1. In the Internet Properties dialog box, select **Advanced**.
2. Scroll to the **Security** section, and verify that **Enable Integrated Windows Authentication** is selected.
3. Click **OK** and restart the browser to activate the changes.

**Configure SAML**

Overview

**Note:** This information does not apply to a programming-only deployment.

To configure the Security Assertion Markup Language (SAML), complete the following:

1. Configure the SAML Service Provider on page 6
Configure the SAML Service Provider

1. Log on to SAS Environment Manager.
2. Navigate to the SAS Logon Manager configuration definitions. For more information, see "Edit Authentication Configuration Instances on page 8."
3. In the Definitions list, select sas.logon.saml.
   
   Note: If you change any of the sas.logon.saml properties, the new metadata must be provided to the Relying Party in ADFS. If it is not, the SAML connections might fail.
4. In the top right corner of the window, click .
5. In the New sas.logon.saml Configuration dialog box, enter values for the required fields, based on your environment. The following table provides guidance on what information needs to be provided for the listed fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>entityBaseURL</td>
<td>The external URL that will be called by the browser (for example, <a href="https://hostname.example.com/SASLogon">https://hostname.example.com/SASLogon</a>).</td>
</tr>
<tr>
<td>entityID</td>
<td>The unique ID that represents the service provider that is included in protocol messages between relying parties. Change from the default value that is pre-populated.</td>
</tr>
<tr>
<td>serviceProviderCertificate</td>
<td>Paste a copy of the PEM-encoded (base64) certificate that will be used by the service provider.</td>
</tr>
<tr>
<td>serviceProviderKey</td>
<td>Paste a copy of the PEM-encoded (base64) key that will be used by the service provider.</td>
</tr>
<tr>
<td>serviceProviderKeyPassword</td>
<td>Provide the password for the service provider, or leave blank if there is no password.</td>
</tr>
</tbody>
</table>
6. Click Save.

Configure the SAML Identity Provider – SAS Environment Manager Configuration

1. Complete the following steps in SAS Environment Manager:
   a. In the Definitions list, select sas.logon.saml.providers.default.
   b. In the top right corner of the window, click .
   c. In the New sas.logon.saml.providers.default Configuration dialog box, enter values for the required fields, based on your environment. The following table provides guidance on what information needs to be provided for the listed fields:
Configure the SAML Identity Provider – Relying Party Configuration

You can either configure the relying party trust or supply the required information to your information technology support group, in order for them to add the relying party trust. Here is an overview of the steps to perform, if you configure the relying party trust. The steps might vary, depending on which tool you use for configuration.

1. If the identity provider requires it, configure Transport Layer Security (TLS), if it has not already been configured. For more information, see “Secure Apache HTTP Server” in Encryption in SAS Viya: Data in Motion.

2. Download the application metadata.xml file, which contains information about the service provider, or provide the https://hostname/SASlogon/saml/metadata link to your information technology support group.

3. Request that your information technology support group configure a relying party in the identity provider.

Configure PAM

Note: This information does not apply to a visual-only deployment.

Default pluggable authentication module (PAM) configuration files are installed for both the CAS server and SAS Studio.

1. As a user with root authority, edit the /etc/pam.d/service file. For the CAS server, service is cas. For SAS Studio, service is sasauth.
The following information is displayed for the CAS server:

```bash
$ vi /etc/pam.d/cas
#%PAM-1.0
auth  include   password-auth
account include   password-auth
password include   password-auth
session include   password-auth
```

The following information is displayed for SAS Studio:

```bash
$ vi /etc/pam.d/sasauth
#%PAM-1.0
auth  include   password-auth
account include   password-auth
```

2 Make any modifications to the file that are necessary for your environment.

3 Save the file and exit.

Session Management

Overview

Note: This information does not apply to a programming-only deployment.

The following sections provide information about customizing SAS Logon Manager and the user's session experience.

Edit Authentication Configuration Instances

1 From the side menu ☰, under SAS Environment Manager, click Resources ☰ Configuration.

2 In the top left corner of the window, select Definitions from the drop-down box.

See Also

- SAS Viya Administration: Configuration Properties

Customize Sign-in, Sign-out, and Session Time-out Content

You can configure customized content that is displayed when users of SAS web applications sign in, sign out, or the session reaches the time-out interval. To enable the display of customize content, follow these steps:

1 In the Definitions list, select sas.logon.custom.

2 In the top right corner of the window, click ✉.

3 In the New sas.logon.custom Configuration dialog box, specify the URI that contains the custom content that you want to display. Here are the available fields:

- login
- logout
- timedout

For a description of the properties, see "sas.logon.custom" in SAS Viya Administration: Configuration Properties.
4 Click **Save**.

**Customize Concurrent Sign-in Sessions**

1 In the **Definitions** list, select **sas.logon.sessions**.

2 In the top right corner of the window, click ![icon](image).

3 In the New **sas.logon.sessions** Configuration dialog box, you can set the following properties:
   - `maxConcurrentSessions`
     Set this property to limit a user to a certain number of concurrent sessions.
   - `rejectNewSessionsIfMaxExceeded`
     When sessions are limited, the default behavior is to cause an existing session to expire and grant a new session to the user attempting to authenticate. To override this behavior and prevent a new session from being granted, set this property to `true`.

4 Click **Save**.

**Configure the HTTP Session Time-out Interval**

1 In the **Definitions** list, select **server**.

2 In the top right corner of the window, click ![icon](image).

3 In the New **server** Configuration dialog box, complete the following:
   a Select **SAS Logon Manager** from the **Services** drop-down list.

   b Click ![icon](image).

   c In the **Name** field, specify **session.timeout**.

   d In the **Value** field, specify the amount of time a session has to be idle before it times out, in seconds.

   e Click **Save**.

4 Click **Save**.

5 Restart all services to reflect the new time-out interval. For more information, see “All Servers and Services” in SAS Viya Administration: General Servers and Services.

**Disable Logins**

As a SAS administrator, you can disable logins through operating system fire wall rules or using LDAP. This will disable new sessions, end current sessions, and prevent others from using the deployment. For more information, see the appropriate documentation for your operating system.

**Additional Authentication Topics**

**Authenticate the REST APIs**

1 Register a new client ID and secret by completing the following steps:
   
   **Note:** You have to register a client ID once.
Obtain a token to register a new client ID and secret. For more information, see “Obtain an ID Token to Register a New Client ID” in Encryption in SAS Viya: Data in Motion.

Use the token to register the new client ID and secret by running the following curl command:

```bash
-d '{
    "client_id": "client-id",
    "client_secret": "client-secret",
    "scope": ["openid", "group-name-1", "group-name-2", "group-name-n"],
    "resource_ids": "none",
    "authorities": ["uaa.none"],
    "authorized_grant_types": ["password"],
    "access_token_validity": "time-in-seconds"
}
```

**Note:** The initial line of the curl command must be entered on one line. It is shown on more than one line for display purposes only.

**Note:** The value for the `scope` parameter should be a list of scopes and groups that you might request when obtaining a token. Ensure that you specify the list correctly. SAS Viya treats group memberships as scopes. Therefore, the list of scopes is the list of group memberships that you might request when obtaining a token. The "openid" is a special scope that represents authentication only. For certain grant types, you might be prompted to approve or deny each of the scopes, so the list should be limited to the minimum necessary.

A token can be used until it expires. By default, this is 12 hours. To acquire a token, run the following curl command:

```bash
curl http://localhost/SASLogon/oauth/token
-H "Accept: application/json"
-H "Content-Type: application/x-www-form-urlencoded"
-d "grant_type=password&username=username&password=password"
-u "client-id:client-secret"
```

**Note:** The values for `client-id` and `client-secret` should be the same as the values that were specified in Step 1b.

Call a SAS Viya API with the token by passing the HTTP Authorization header as a Bearer token:

**Authorization: Bearer token.**

Create an Authinfo File

The Authinfo file supplies a user name and password that is sent to CAS for authentication. For information about how to create a .authinfo file, see “Authinfo File Requirement” in SAS Viya: System Programming Guide.

Authentication: Concepts

Authentication Architecture

In a visual-only and full deployment, authentication services are provided by SAS Logon Manager. SAS Logon Manager is based on the Cloud Foundry User Account and Authentication (UAA) server. The security architecture is built around Open Authorization (OAuth) and OpenID Connect. By default, authentication is performed via a Lightweight Directory Access Protocol (LDAP) provider. Authentication support is also available for Kerberos, OAuth, and Security Assertion Markup Language (SAML).
In a programming-only and full deployment, host authentication is supported. You can configure the host to use only pluggable authentication modules (PAM).

**Authentication Options**

**Authentication for Visual Interfaces**
With visual interfaces, users are authenticated through SAS Logon Manager. SAS Logon Manager is a web application that handles all authentication requests for SAS web applications and is accessed via the Apache HTTP Server.

The following figure shows how a user authenticates to SAS Logon Manager and the supported authentication mechanisms.
The following four protocols are available for you to configure for authentication:

- The first option is a Lightweight Directory Access Protocol (LDAP) provider. This is the default configuration. In this configuration, SAS Logon Manager displays a logon form and submits the entered credentials to LDAP. The identity service verifies users in LDAP. For more information, see LDAP Authentication on page 15.
The second option is Kerberos (Integrated Windows Authentication). In this configuration, SAS Logon Manager uses SPNEGO to authenticate users against the Kerberos Key Distribution Center (KDC). The identity service verifies users in LDAP. For more information, see Kerberos Authentication on page 15.

The third option is OAuth. In this configuration, SAS Logon Manager uses OAuth and OpenID Connect to authenticate users. The identity service verifies users in LDAP. For more information, see OAuth Authentication on page 21.

The fourth option is Security Assertion Markup Language (SAML). In this configuration, SAS Logon Manager uses a SAML provider to authenticate users. The identity service verifies users in LDAP. For more information, see SAML Authentication on page 21.

With all four options, the connection to the SAS Cloud Analytic Services (CAS) environment is performed using internal OAuth tokens that are generated by SAS Logon Manager. In most cases, the session that is started by the CAS controller runs on the operating system as the same user who launched the CAS operating system service. This defaults to the cas account.

**Authentication for Programming Interfaces**

The following figure shows how a user is authenticated while using programming interfaces.
In a deployment with programming interfaces, the user’s credentials are entered into SAS Studio via the Apache HTTP Server. SAS Object Spawner then uses pluggable authentication module (PAM) configuration files on the host to validate the user ID and password. The user ID and password can be a local account on the host or, depending on the PAM configuration, an account in the LDAP provider. Once the user is authenticated, the SAS Workspace Server is started. The PAM configuration file for SAS Studio is sasauth and includes the password module.

SAS Workspace Server connects to the CAS environment using the user ID and password that were used to start the SAS Workspace Server. The CAS controller uses its own PAM configuration to validate the user’s credentials and launch the session process as the user. The PAM configuration file for CAS is cas and includes the password module.

The CAS controller uses the user ID and password to obtain an internal OAuth token from SAS Logon Manager. This requires the user ID and password to be valid in the LDAP server that is configured for SAS Logon Manager. Otherwise, CAS cannot obtain an OAuth token, and the session will fail. Therefore, PAM for SAS Studio (sasauth), PAM for CAS (cas), and SAS Logon Manager should all use the same or equivalent LDAP
Authentication Mechanisms

LDAP Authentication

Overview of LDAP

Note: This information does not apply to a programming-only deployment.

In SAS Viya, LDAP is used for identifying and authenticating users. Third-party LDAP server implementations are supported, including Microsoft Active Directory and OpenLDAP.

How It Works in SAS Viya

LDAP is the default authentication mechanism. The Identities service always makes a direct connection to LDAP to obtain user and group information. By default, SAS Logon Manager authenticates users using a direct connection to the configured LDAP provider. To ensure that network connections are secure, the connection between the browser and the Apache HTTP Server can be secured with HTTPS. In addition, the connection between SAS Logon Manager and the LDAP provider can be secured with LDAPS.

For information about configuring LDAP, see “Configure the Connection to Your Identity Provider” in SAS Viya: Deployment Guide.

Kerberos Authentication

Overview of Kerberos

Note: This information does not apply to a programming-only deployment.

Kerberos is a network authentication protocol that is used to verify user or host identity. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a service (and vice versa) across an insecure network connection. During Kerberos authentication, a user’s credentials (user ID and password) are not sent over the network. Instead, both the client and the service use the credentials that they have been supplied as a key in an encryption algorithm to encrypt the message that is sent between the client and the service. If the client sends an encrypted message, when the service uses the same key to decrypt the message, it is proven that the credential is known without having to transmit the credentials.

Key Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>An application that is attempting to connect to and access a resource, on behalf of a user. Resources include reports that are viewed, services that are accessed, and databases that are queried. In SAS Viya, the client is the web browser.</td>
</tr>
<tr>
<td>Service</td>
<td>A service, or server, that hosts a resource the user wants to connect to. The service must be able to validate the service tickets presented by the client.</td>
</tr>
<tr>
<td>Key Distribution Center</td>
<td>A trusted third party within Kerberos that verifies the authenticity of the client and service. Both the client and service must trust the KDC. In addition, end users and services must register with the KDC.</td>
</tr>
</tbody>
</table>
Service Principal Name
A unique name that is used to identify a web service that is running on a server. Before a service principal name (SPN) can be used, it must be registered. Every web service that uses Kerberos authentication needs to have an SPN set for it so that clients can identify the server on the network. An SPN usually matches the pattern of HTTP/hostname.example.com.

Keytab File
A file containing pairs of Kerberos principals and encrypted keys. The keys are associated with a password for the principal. The principals are SPNs. Keys can use different encryption algorithms. For a single principal, you might have several entries that correspond to each encryption type.

Ticket-granting ticket
An encrypted identification file that is valid for a limited amount of time. After a user is authenticated, this file is granted to a user for data traffic protection by the KDC. The TGT file contains the session key, its expiration date, and the user's IP address.

How It Works in SAS Viya
In addition to using the LDAP provider to obtain user and group information, you can configure SAS Logon Manager for Kerberos authentication. This option replaces the option to use the default LDAP provider for authentication to SAS Logon Manager. Kerberos, which is also referred to as Integrated Windows Authentication (IWA), provides the user with single sign-on capabilities from the browser on their desktop. Single sign-on allows the user to access the SAS Viya visual interfaces without being prompted to enter their credentials.

IWA
IWA uses Kerberos authentication and is a Microsoft technology that is used in an environment where users have Windows domain accounts. With IWA, the credentials are hashed before being sent across the network. The client browser proves its knowledge of the password through a cryptographic exchange with the web application server. When IWA is used in conjunction with Kerberos, IWA enables the delegation of security credentials. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection.

For configuration information, see How to Configure IWA on page 2.

Kerberos in Visual Interfaces
There are three possible scenarios for accessing a Hadoop environment that is secured by Kerberos (IWA). The following figure illustrates the first scenario:
In this scenario, there is a full deployment and a user that relies on Kerberos or Integrated Windows Authentication to log on to SAS Logon Manager as part of his or her access to the visual interfaces. The user accesses the visual interfaces and connects to CAS and a secured Hadoop environment. The connection between the visual interfaces and CAS uses the internal OAuth to authenticate the user. Users obtain an OAuth token from the SAS Logon Manager as part of their initial authentication, and this token is used to authenticate to CAS. The OAuth token provides user and group information that enables CAS to provide integration with the authorization information that is stored in the SAS services.

Since the user used an OAuth token to connect to CAS, there are no operating system credentials provided to CAS. Therefore, no credentials are forwarded to the CAS Server Controller, and it cannot authenticate the user.
and launch a CAS Session Controller running as the user. The CAS Session Controller and CAS Session Workers all run as the service account that launched the CAS operating system service. By default, this is the cas account.

If the user needs to access a secured Hadoop cluster, where Kerberos Service Tickets are required to access the data, then the CAS Server Controller Kerberos credentials need to be provided. The credentials are provided in the form of a Kerberos keytab file. This keytab file enables the CAS Server Controller to initialize a set of Kerberos credentials at start-up. This set of Kerberos credentials does not have to be for the cas user. The credentials in the keytab can be for any user, but all access from CAS to Hadoop uses this single, shared credential. In the diagram, the shared credentials are listed as “CAS Hadoop Account”.

Kerberos in Programming Interfaces

The following figure illustrates the second scenario:
In this scenario, there is a full deployment and a user that provides his or her user ID and password to CAS. CAS uses its own pluggable authentication modules (PAM) configuration to validate the user’s credentials and launch the CAS Session Controller process running as the user. In addition, the CAS Server Controller also uses the user ID and password to obtain an OAuth token from SAS Logon Manager and then to obtain access control information from the SAS Viya services. Obtaining the OAuth token from the SAS Logon Manager ensures any restrictions or global caslibs defined in the visual interfaces are observed in the programming interfaces.

With the CAS Session Controller running as the user and any access controls validated, the CAS session can access the secured Hadoop cluster. Since the CAS session is launched using the PAM configuration, the Kerberos credentials used to access Hadoop are those of the user. The PAM configuration on the machines hosting CAS must be linked to Kerberos. The PAM configuration then ensures that the Kerberos Ticket-Granting Ticket is available to the CAS Session as it is launched.

Users accessing both the programming interfaces and the visual interfaces see different permissions in the secured Hadoop environment, depending on which interface they are using. With programming interfaces, they access Hadoop with their own account. Conversely, with visual interfaces, they access Hadoop with a shared credential.

Kerberos in Visual Interfaces Using a Custom Group

In the third scenario, there is a full deployment in which users attempt to access Hadoop using the custom group, CASHostAccountRequired. For more information about the CASHostAccountRequired custom group, see “Initial Rules for Other Predefined Custom Groups” in SAS Viya Administration: Identity Management.

**CAUTION! You might experience unexpected behavior.** The “no entry” signs in the following figure show that the user is not allowed in access Hadoop because Hadoop is secured with Kerberos, and the user does not have the proper credentials. The shared credentials are not passed on because the connection is being run by a different user.
In this scenario, the user of the visual interfaces has an experience closer to the programming users when launching the CAS session. As an administrator, if you can create the CASHostAccountRequired custom group and add users to this group, CAS processes the start-up of their sessions differently. If a user is a member of the CASHostAccountRequired group, CAS looks for sufficient information in the OAuth token to launch a valid operating system process as the user.

The CAS Server Controller directly launches the CAS Session Controller using the user ID from the OAuth token rather than passing any credentials to the PAM stack on the host. Therefore, the user ID from the OAuth token must be known to the operating system where the CAS Session Controller and CAS Session Workers are running. The process runs as the user on the host, even though the user was not authenticated on the host.
Since the PAM stack was not used to authenticate the user and launch his or her CAS session, the user's Kerberos credentials were never initialized.

In addition, since the CAS session is now running as the user, it is unable to access the Kerberos credentials cache initialized by the CAS Server Controller on start-up from the Kerberos keytab. Therefore, the user cannot access the secured Hadoop cluster.

OAuth Authentication

Overview of OAuth

Note: This information does not apply to a programming-only deployment.

Open Authorization (OAuth) is a token-based authentication standard on the Internet. OAuth allows a user's account information to be used by third-party services, such as Google, without providing the user's password. OAuth acts as an intermediary on behalf of the user, giving the third-party service an access token that authorizes specific account information.

Key Terms

<table>
<thead>
<tr>
<th>Access token</th>
<th>Specifies identifying information for a user, including the user's credentials, groups, and privileges.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenID Connect</td>
<td>An authentication layer built on top of OAuth 2.0.</td>
</tr>
<tr>
<td>Flow</td>
<td>The process for obtaining an OAuth token.</td>
</tr>
</tbody>
</table>

How It Works in SAS Viya

An OAuth and OpenID Connect provider can be internal to the customer's environment or it can be an external provider, such as Google Authenticator or Facebook. When the OAuth option is configured, this does not completely replace the default LDAP provider. Instead, when users access the SAS Logon Manager, they are presented with a link to authenticate using OAuth and the standard logon form using the LDAP provider. Users can then select which to use. OAuth can provide single sign-on from the OAuth provider. For example, when a user signs in to his or her Google account, the user can then access the visual interfaces of SAS Viya without being prompted any further for credentials.

When you access CAS from a web application, such as SAS Visual Analytics or SAS Environment Manager, your OAuth token is validated. A CAS session is launched under a shared service account, unless you are in the custom group, CASHostAccountRequired, whose members always access CAS using their personal host account.

The CAS server accepts OAuth tokens from SAS Visual Analytics and SAS Visual Investigator. It accepts user ID and password from SAS Studio and batch connections. When the user ID and password are specified, both PAM and SAS Logon Manager authentication are performed. The SAS Logon Manager authentication produces an OAuth token that is validated.

SAML Authentication

Overview of SAML

Note: This information does not apply to a programming-only deployment.

The Security Assertion Markup Language (SAML) standard defines a framework for exchanging security information about users between an identity provider and service provider. This security information is packaged
in the form of portable XML assertions that applications working across security domain boundaries can trust. SAML allows for single sign-on to web browser applications.

Key Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Federation</td>
<td>Allows multiple identity management systems to work together and establish trust.</td>
</tr>
<tr>
<td>Assertion</td>
<td>A package of information, in the form of an XML document, that is created and sent during a federated access request.</td>
</tr>
<tr>
<td>Claims</td>
<td>Information that a federation member is asserting to be true.</td>
</tr>
<tr>
<td>Identity provider</td>
<td>A federation member that authenticates users and keeps track of their information. Creates assertions for the users, and sends them to service providers.</td>
</tr>
<tr>
<td>Service provider</td>
<td>A federation member that consumes assertions to make access control decisions for its applications.</td>
</tr>
<tr>
<td>Metadata</td>
<td>An XML document that is produced by a SAML provider to describe its service endpoint URLs, x.509 certificate, and other information in a standard way for consumption by partners in the federation.</td>
</tr>
<tr>
<td>Relying party</td>
<td>A server providing access to secure software.</td>
</tr>
</tbody>
</table>

How It Works in SAS Viya

SAML supports configuring the SAS Logon Manager to be integrated with an external SAML identity provider. This identity provider can be internal or external to the customer’s environment. If it is internal, a tool similar to Oracle Access Manager can be used. If it is external, something like salesforce.com can be used. SAML does not completely replace the default LDAP provider. End-users accessing the SAS Logon Manager can choose SAML authentication or the default LDAP provider. This option also provides single sign-on with the third-party SAML provider.

When a user attempts to access a service URL, the service provider, which is SAS Logon Manager, initiates the exchange with an authentication request, and the identity provider sends a response that contains the assertion. The SAML protocol defines the structure and content of these request and response messages. When the user logs in to a service or system, instead of providing credentials to the service provider, the service provider trusts the identity provider to validate the credentials. Therefore, users do not have to provide their credentials directly to anyone but the identity provider.

For configuration information, see Configure SAML on page 5.

PAM Authentication

Overview of PAM

Note: This information does not apply to a visual-only deployment.

Pluggable authentication modules (PAM) enable you to determine how applications use authentication to verify the identity of a user. It is an industry-standard technology that extends UNIX host authentication to recognize additional authentication providers. PAM uses “modules” or libraries to access multiple authentication methodologies. SAS Viya supports host authentication.
How It Works in SAS Viya

Default PAM configuration files, `/etc/pam.d/service`, are installed as a part of the SAS Viya deployment process.

**Note:** For the SAS Cloud Analytic Services (CAS) server, `service` is `cas`. For SAS Studio, `service` is `sasauth`.

For sasauth to perform authentication, entries must be made in the PAM configuration files that are provided by SAS, that describe what authentication services are used when sasauth performs an authentication. This includes the account and auth module types. The session and password modules are not supported.

In a multi-machine deployment, configure PAM on the host with SAS Object Spawner and the host with CAS controller.

For configuration information, see Configure PAM on page 7.

**Authinfo File**

Authentication is used to control access to the CAS server and its resources. Your identity must be successfully authenticated before your session is created. SAS Studio authenticates the connection to CAS by using your user credentials. When password information is not available, an attempt is made to find an Authinfo file (.authinfo is the default filename on Linux). The Authinfo file provides a user name and password to CAS for PAM authentication. It is an alternative to including passwords in programs.

You can also force the use of the Authinfo file by specifying the CAS_AUTH_METHOD environment variable. For more information, see `CAS_AUTH_METHOD`.

The Authinfo file is required when you are using the command line to submit commands for the following tasks:

- Run programs in batch mode. The USER= option in the CAS statement or SAS system option `CASUSER=` can be specified.
- Perform limited server administration using the `casadmin` command.
- Run commands in line mode.

**Note:** SAS Studio user credentials are used to authenticate your connection to CAS. SAS Studio does not use the `.authinfo` file for authentication.

Typically, the `.authinfo` file resides in the `$HOME` directory.

The Authinfo file format is based on the `.netrc` file specification. The `.netrc` file format is an older format. You can see the file specification at Netrc Format. In addition to the standard `.netrc` file standards, the Authinfo specification allows for putting commands in the file as well as using quoted strings for passwords. The quoted strings allow for spaces within passwords.

If the Authinfo file contains values that match the host, port, or user name, the information contained in the Authinfo file is used to connect to CAS.

The following system options and environment variables can be used to override the `.authinfo` file. These options point to Authinfo files that are located in a different directory or that are named differently. Here are the ways that the AUTHINFO system option, environment variable, and the statement option can be used to override the Authinfo file.

- Environment variable `AUTHINFO` takes precedence over the `.authinfo` file.
- SAS system option `AUTHINFO=` (alias `CASAUTHINFO=`) overrides the `AUTHINFO` environment variable as well as the `.authinfo` file.
- `AUTHINFO=` option in the CAS statement overrides the `AUTHINFO=` system option, the `AUTHINFO` environment variable, and the `.authinfo` file.

For more information, see the following documents:

- “AUTHINFO= System Option” in SAS Cloud Analytic Services: Language Reference
Additional Authentication Topics

SAS/CONNECT Authentication

Note: This information does not apply to a visual-only deployment.

As an administrator, you might want to enable SAS Viya to accept connections for existing SAS 9 environments. SAS/CONNECT enables that connection, and passes credentials that can be used in the SAS Viya environment.

With SAS Viya, your credentials are used to authenticate to CAS when you are using SAS/CONNECT. When additional SAS/CONNECT servers are spawned, SAS/CONNECT forwards your credentials to the spawned SAS/CONNECT server session.

Here are the ways that SAS/CONNECT and CAS authenticate your user credentials.

- When the user is using any environment that is not a SAS Viya environment and is connecting to SAS Viya via the SAS/CONNECT spawner, the spawner passes the SIGNON credentials to the SAS/CONNECT server where the credentials can be used to connect to CAS.
- When the user is in the SAS Viya environment using SAS Studio and starting SAS/CONNECT server sessions (using SASCMD SIGNON or the CONNECT Spawner), the CAS credentials (if they exist) are passed to the SAS/CONNECT server in SAS Viya.
- When running SAS Viya in batch or line mode, the Authinfo file is used to authenticate to CAS. If you specified the USER= option in the CAS statement, CASUSER= system option, or if you specified the CAS_AUTH_METHOD environment variable, Authinfo file authentication is used.


Single Sign-On

Note: This information does not apply to a programming-only deployment.

Single sign-on (SSO) is an authentication model that enables users to access a variety of computing resources without being repeatedly prompted for their user IDs and passwords. For example, SSO can enable a user to access SAS servers that run on different platforms without interactively providing the user's ID and password for each platform. SSO can also enable someone who is using one application to launch other applications based on the authentication that was performed when the user initially logged on.

The SAS Logon Manager is the central point for handling changes to authentication mechanisms, such as the addition of third-party SSO products. SAS Viya supports the following SSO products:

- Kerberos on page 15
- SAML on page 21
- OAuth on page 21

Dual Authentication

In a dual authentication environment, users are validated against the LDAP server and the host authentication mechanism. The following conditions exist:
- If PAM is configured to use local accounts and those users also log on to the visual components, then those local accounts must match the LDAP server used for SAS Logon Manager.

- If PAM is configured to use an LDAP server, SAS Logon Manager should be configured to use the same LDAP server.

- When directly connecting to the CAS server using SAS Studio or a batch job, the user ID and password that are supplied are authenticated against both the LDAP server and PAM.