SAS® GLOBAL FORUM 2020 #SASGF

Paper SAS4613-2020

SAS® Cloud Analytic Services Sessions: Understanding Connection Options to Ensure Data Access Security

Philip Hopkins, SAS Institute Inc.

ABSTRACT

In SAS® Viya®, there are a variety of ways to connect to the SAS® Cloud Analytic Services (CAS) analytics engine, or CAS server. The resulting CAS sessions might run under the CAS instance's service account or the end user's identity. This is further complicated by whether the CAS client is using the Kerberos authentication protocol, which results in differences, depending on whether CAS is running on Linux or Windows. The various SAS Viya applications default to certain connection methods, which drive their CAS session's process owner and, finally, the resulting physical security contexts. Understanding the default CAS session behavior for each application and how those defaults can be overridden and engineered to support data access policy is the focus of this paper.

INTRODUCTION

SAS Viya includes multiple run-time engines that offer the ability to do valuable compute processing. The traditional SAS® Programming Runtime Environment (SPRE) lives on in SAS Viya[1] alongside the SAS Cloud Analytic Services (CAS) in-memory analytics engine and the SAS® Micro Analytic Service (MAS). When these runtime engines access sensitive data sources, their sessions should be instantiated in a manner that will ensure data access in compliance with your security policies.

CAS is the primary engine for performing analytics and batch scoring on large in-memory tables at the speed of random-access memory (RAM). CAS sessions are capable of accessing external data sources from a variety of data providers and loading that data into memory creating CAS tables for analysis. While most client web applications connect directly to CAS, some SAS Viya clients use SPRE compute sessions as conduits to CAS; these compute sessions are also capable of accessing sensitive data. Open-source clients might also connect to CAS.

This paper focuses primarily on data accessed through the CAS engine and how its sessions are launched. Variations in session launch, driven by the CAS authentication method, has an effect on session identity and security context. To comply with the requirements of a secure authorization model, it is necessary to engineer CAS session launches appropriately, in accordance with the authentication providers in place for SAS Viya and for relevant data providers. For scenarios where SPRE compute sessions are used as a data access vehicle, it is also necessary to consider how compute sessions are launched, which affects their identity and security contexts.

[1] The resulting SAS session is unchanged. However, the way the session is launched and the communication protocols for distributed access have changed in SAS Viya.

The MAS[2] runtime is a RESTful microservice capable of returning decisions for supplied transaction data based upon published models or decision flows. It is typically used to operationalize analytical models for making decisions in real time. However, MAS sessions are not separate processes the way that CAS and SPRE sessions are. All requests to MAS execute in separate threads of the MAS web service (running under the userID for the SAS Viya installation account). All executing threads inherit the same security context. The transaction data to be processed is supplied by the SAS Viya client or a third-party application at run time. There is no special engineering required in consideration of facilitating data access[3] for MAS, so there will be no discussion of the MAS runtime.

This paper is organized into two parts. The first half discusses how authentication schemes affect the data provider security context of CAS and SPRE compute sessions. The second half discusses how to engineer SAS Viya so that CAS and SPRE compute sessions remain in compliance with data access policies.

IDENTITY AND SECURITY CONTEXTS

Data access privileges are relevant only in the context of some identity. Logging in to any SAS Viya web application establishes your SAS Viya identity. If your browser has yet to establish a session with the SASLogon service, you are redirected to SASLogon for initial authentication[4]. The SASLogon service is an implementation of an OAuth 2.0 authorization server that supports Open ID Connect extensions for authentication. When a user is authenticated, SASLogon builds an OAuth access token (with help from the identities service), containing your username and all of your group memberships to establish an identity profile.

The OAuth access token represents your SAS Viya identity's application security context. The SAS Viya visual web applications use the access token to determine which SAS Viya resources (reports and content folders) you are privileged to access. CAS sessions use the access token to determine which caslibs and loaded tables you have access to. SPRE sessions can also connect traditional SAS libraries to the caslibs that your SAS Viya identity has access to.

CAS SESSIONS AND OUTBOUND SECURITY CONTEXT

A CAS session is a separate collection of processes that replicate the CAS server's processes (controller and workers) for a specific identity. CAS sessions must always have an OAuth access token for the user representing the session's identity. The session processes run under some operating system userID[5] which raises the possibility of an additional and distinct security context.

If the session's application security context is defined by the caslibs and loaded tables its Viya identity can access, then the session's outbound, or data provider, security context is defined by any external data sources it has access to, through the caslibs accessible to its Viya identity.

The data source access will be in the context of the specific data provider. In this paper, data provider is used in the broadest sense, to include:

---

[2] MAS is available only when either of the following SAS Viya offerings are licensed: SAS® Model Manager, SAS® Intelligent Decisioning, or SAS® Event Stream Processing.

[3] For MAS to access a third-party RDBMS, some configuration of drivers and data sources might be required.

[4] By default, SASLogon authenticates using a supplied username and password against a configured LDAP provider. SASLogon also supports Kerberos, SAML, and OpenID Connect authentication.

[5] On Windows, CAS sessions always launch in user identity, due to the way SAS Viya authentication works on Windows. This will be covered further in the section on Authentication.

- Linux operating system for data source files accessed by SPRE or CAS. The process owner for the SPRE or CAS session must have Linux permissions to read, modify, or replace the file as needed.
- Relational database systems, such as Oracle or Teradata.
- Hadoop HDFS or HIVE.

When a CAS session requests access to a caslib or loaded CAS table, authorization is granted by the CAS server, in accordance with its SAS Viya application security context. However, when the session requires access to a data source, the authorization must be granted by both the CAS server and the relevant data provider.

Typically, caslibs for an RDBMS have a security context based on the data provider credentials specified in their connection information. These credentials can be any valid username and password that the caslib creator has access to and may represent an identity that is different from either the Viya or outbound session identities.

However, the data provider security context for O/S file based caslibs (PATH, DNFS) and Hadoop caslibs is a function of the userID under which the session process is launched, which is dependent upon its authentication method. Understanding how CAS establishes outbound session security contexts is the focus of the next few sections.

DETERMINING CAS SESSION SECURITY CONTEXTS

The primary factor in determining the process owner for the CAS session is the authentication method used to connect to CAS. The CAS server supports different types of credentials for authentication, which influence how the user's CAS session is launched, and ultimately the data provider security context. CAS supports the following types of authentication.

- SAS Viya OAuth Authentication
- External (Host) Authentication
- Kerberos Authentication

To effectively control the CAS session's security context and achieve the appropriate level of data access for each user, you must be familiar with each of the authentication methods and the scenarios under which they take place.

CAS AUTHENTICATION METHODS

OAuth Authentication

When SAS Viya clients connect to CAS, they send their session's OAuth access token and CAS uses the token to establish its session's SAS Viya identity. Open-source clients can acquire an OAuth token from SASLogon using its open REST API and then supplying the token to CAS for authentication. CAS clients supplying an OAuth token in their connection request can be categorized as OAuth clients.

The default behavior for OAuth based CAS connections is to launch CAS sessions using the configured service ID for the relevant CAS instance, which in typical deployments will be a host account called 'cas'. The CAS session processes are owned by the service userID (for example, 'cas'), which sets a data provider security context for path-based caslibs in the 'cas' host user's identity.

This default behavior can be changed, and these options will be discussed in the Overriding Default CAS Session Launching Behavior section.

External (Host) Authentication

When CAS receives a userID and password, in lieu of an OAuth access token, CAS attempts to acquire a SAS Viya access token from SASLogon, using the supplied username and password to authenticate. If CAS is unable to acquire the OAuth access token the connection request fails. CAS also performs host authentication, i.e. authentication through its underlying host machine[6]. For Linux host authentication, CAS passes the userID and password to the PAM subsystem for authentication. If authentication succeeds, CAS looks up the UID in the Name System Services (NSS) subsystem for identification and other attributes needed for session launch such as the home directory. If identified, the connection is established, and the session is launched in the user's identity.

The CAS session's outbound security context will be derived from the userID supplied. The session will include a SAS Viya security context, to ensure proper access to caslibs and loaded tables, and a data provider security context based on the userID used for authentication. If the underlying PAM subsystem is configured for Kerberos authentication, then the session's data provider security context will include access to a Kerberos credential for the user.

CAS clients supplying a password in their connection request can be categorized as "password clients". Password clients are typically custom programs written in SAS or other supported CAS client languages such as Java or Python, but also include a SAS Workspace Server. Here are some password client scenarios resulting in host authentication:

- Non-SAS clients, such as Python or Java programs, connecting directly to CAS
- SAS9 or SAS Viya Workspace Servers connecting to CAS
- SAS® Studio 4.x in SAS Viya 3.4 and earlier
- SAS® Studio 5.2 (Basic) in SAS Viya 3.5

Kerberos Authentication

In the default configuration, CAS supports OAuth and external authentication as described above. However, CAS can also be configured to support Kerberos authentication and can accept a Kerberos ticket for authentication in a connection request. Clients that supply CAS with a Kerberos ticket in the initial connection request can be categorized as Kerberos clients. This is an uncommon, but possible scenario as Viya does not provide any such clients. The more typical scenario for Viya is to leverage Kerberos in conjunction with OAuth authentication as will be discussed later in the *Overriding Default CAS Session Launching Behavior* section.

Here are some basic direct Kerberos scenarios for CAS:

- accepting a Kerberos connection from a custom Kerberos client
- accepting a Kerberos connection from a SAS 9 Workspace Server where SAS 9 is configured for Integration Windows Authentication (IWA)

SPRE COMPUTE SESSIONS AND OUTBOUND SECURITY CONTEXT

SAS Viya web applications that leverage SPRE are SAS Studio (Enterprise) and SAS® Model Studio. Both of these web applications use compute session runtimes to mediate their connections to CAS. In this context, the SPRE compute sessions act as a CAS OAuth client with data analysis executing within the CAS runtime. However, users can execute data management or analytics code within the SPRE compute sessions, accessing data directly.

[6] The userID and password can either be supplied directly by the connecting client or extracted from a user's `.authinfo` file.

In this context, the SPRE compute session acts a compute server, potentially consuming significant machine resources.

The job execution service in SAS Viya launches SPRE compute server sessions, and SAS batch jobs are SPRE batch sessions. Both can contain code to connect to CAS or access data sources or both.

Each scenario has different implications for complying with data access policy, in accordance with the session's data provider security context for data access. For file-based data sources or those within a data provider secured using Kerberos, this is a function of the identity under which the SPRE sessions are launched. The primary authorization concern for SPRE sessions doing their own computational work is the relevant data provider for the library's data source.

Unlike a caslib, a traditional SPRE library is not a securable object in SAS Viya as it was in SAS®9. However, when a SPRE session is connected to CAS, librefs can be defined using the CAS engine to act as references to user accessible caslibs.

SPRE compute sessions are launched by the Launcher Server process, based upon the username in the OAuth access token requested by the client web application (SAS Studio or SAS Model Studio)[7]. By default, the Launcher Server extracts the userID from the access token and directly launches a server process in that identity, provided the userID is a valid host account and that a home directory exists on the machine. The Launcher behavior is therefore opposite to that of CAS, which defaults to launching CAS processes as the CAS server's service account.

SAS VIYA CLIENT SESSIONS SUMMARY

By default, the majority of SAS Viya visual web applications make a direct OAuth connection to CAS. SAS Model Studio and SAS Studio 5.2 (Enterprise) connect to CAS through their Compute Server sessions, which also make OAuth connections to CAS. SAS Studio (Basic) and SAS Studio 4.x use the traditional SAS Workspace Server which authenticates to CAS with a username and password. Open-source clients have the option of username and password or first acquiring an OAuth token and using it. The default connection and session launch behavior for CAS clients is summarized in Figure 1.

[7] Some steps in the process between the client web application and the Launcher Server have been omitted for simplicity, as the complete chain of service calls is not crucial in this context.

User Client	CAS Client (CAS Client Category)	Default CAS Client Process Owner	Default CAS Session Process Owner
SAS Data Explorer	Microservice (OAuth)	sas	cas
SAS Visual Analytics	Microservice (OAuth)	sas	cas
SAS Data Studio	Microservice (OAuth)	sas	cas
SAS Environment Manager	Microservice (OAuth)	sas	cas
SAS Model Studio	SPRE Compute Session (OAuth)	Viya user	cas
SAS Studio 5.1/5.2 (Enterprise)	SPRE Compute Session (OAuth)	Viya user	cas
SAS Studio 4.x	SPRE Workspace Server (Password)	SAS Studio user	Username
SAS Studio 5.2 (Basic)	SPRE Workspace Server (Password)	SAS Studio user	Username
SAS 9.4 M5+	SPRE Workspace Server (Password)	SAS 9 User	Username
SAS Program	Batch SAS Code (Password)	Batch process owner	Username
Registered OAuth client program	Batch SAS Code (OAuth)	Batch process owner	cas
Open Source Program	Open Source Code (Password)	Open Source user	Username
Registered OAuth client Open Source Program	Open Source Runtime (OAuth)	Open Source user	cas

Figure 1: Default Connection Information for a CAS Session

OVERRIDING DEFAULT CAS SESSION LAUNCHING BEHAVIOR

There are additional configuration options that can be used on Linux deployments to augment the CAS launch process for OAuth clients so that CAS user sessions launch in the SAS Viya user's identity as opposed to the CAS server identity. As discussed earlier in the section *CAS Authentication Methods*, password and kerberos clients get their sessions in the user's identity by default.

There can be many different configuration combinations that result when these options are leveraged, and there is a very specific priority in CAS as to how they are processed. After first understanding the options in this section, the processing logic will be explained in the "*CAS Session Launch Decision Flow"* section.

WINDOWS

On Windows, initial authentication for SAS Viya and CAS authentication are configured for Kerberos by default. As with Linux, CAS still receives an OAuth connection from SAS Viya clients. The SASLogon and CAS Kerberos principals are configured to allow delegation ensuring the users' Kerberos tickets from their PC (browser or other Kerberos client) are forwarded to SASLogon and persisted as SAS Viya credentials. Subsequently, both the SAS Viya Launcher service and CAS have access to the Kerberos credentials for a Viya user's

connection. This ensures that on Windows, both SPRE compute and CAS sessions always run as the client identity[8].

CAS HOST ACCOUNT REQUIRED MEMBERSHIP

For SAS Viya client connections and CAS on Linux, the first requirement is to put users into a special group (a SAS Viya custom group or a group from the LDAP provider), whose groupID[9] (SAS Viya) or group name (LDAP) is exactly named '*CASHostAccountRequired*'. In the remainder of this paper, this group will be referred to as the CHAR group.

The CHAR group is not present by default as a SAS Viya custom group. When defined (either as a custom group or present in the configured LDAP provider), membership in this group is flagged on a user's OAuth token by the Viya identities service. For OAuth client scenarios, membership in this group is a necessary condition for CAS to attempt session launches under the user's OAuth identity, which must map to a valid userID on the underlying CAS host machine. CAS refers to this as a *host launch*. However, the exact identity that CAS will use is a function of other settings, which are discussed next.

Figure 2 is an example CHAR custom group with a single member.

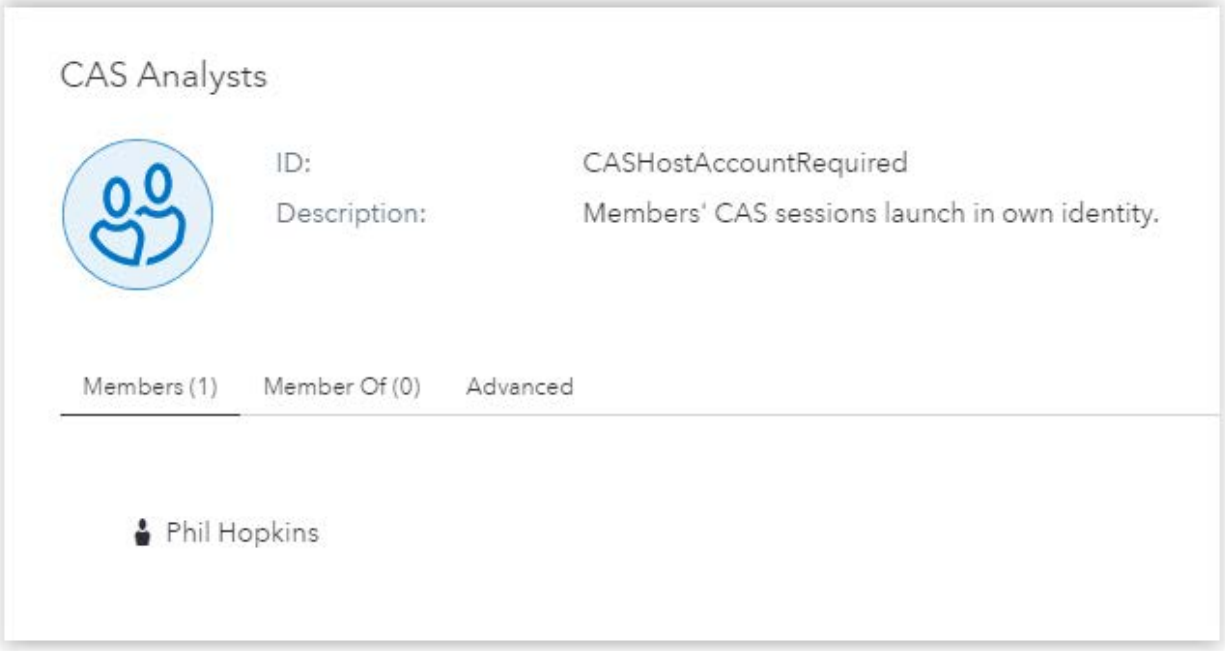

Figure 2: CAS Host Account Required Custom Group

END TO END KERBEROS

Another option for augmenting how CAS launches session identities is to configure SAS Viya for Kerberos authentication during initial logon and propagate the user's Kerberos credentials through to their CAS sessions.

[8] On Windows, if the CAS Kerberos handshake fails for any reason, CAS will launch a session under its configured service account identity for admin users, but the connection will fail for non-admin users.

[9] The **Name** field for the custom group can be changed if desired. Group nesting is supported such that users do not have to be direct members: they can be members through membership in another group which is itself a member of the CHAR group.

As part of CAS Kerberos configuration, CAS has access to a Kerberos keytab for its server's Kerberos principal. For each launched session, CAS initializes a Kerberos credential cache for its Kerberos principal using the key in the CAS keytab. This establishes an outbound security context, for caslibs referencing data source providers that use Kerberos, in the server's identity. For sessions launching in the server's identity, this credential cache could be used to obtain service tickets for downstream resources, such as a Hadoop data provider. The data provider would need to accept a connection for that CAS identity.

However, if CAS has access to a Kerberos credential for the SAS Viya user authenticating from an OAuth client, CAS will attempt a host launch using the username contained in the OAuth token. For CAS to have access to the user's Kerberos credential when SAS Viya OAuth clients connect, SASLogon must be configured for Kerberos authentication with delegation privileges.

When SASLogon is configured for Kerberos and its Kerberos principal is trusted for delegation, the Viya web client (e.g. SAS Visual Analytics) forwards the user's Kerberos credential to SASLogon as part of the initial Kerberos authentication. If the SAS Admin CLI connects to SASLogon using Kerberos, it will also forward the user's Kerberos credential to SASLogon. For clients connecting from the SAS® Visual Analytics mobile app with a username and password or from the SAS Admin CLI with a username and password, SASLogon will initialize a Kerberos credential using their password.

All of these logon scenarios result in SASLogon persisting the user's Kerberos credential within SAS Viya. This makes it available for CAS during Kerberos connections from SAS Viya OAuth clients and allows CAS to launch the session in the SAS Viya user's identity.

SAS VIYA STORED CREDENTIALS

For OAuth clients whose connecting user is a member of the CHAR group, when CAS is not configured for Kerberos, CAS first looks to see if the user has access to a stored credential in SAS Viya, either as a personal credential stored for their own identity or as a shared credential available to some other group that the user belongs to. If this credential exists, CAS will attempt to launch the user's session in the identity of the fetched userID.

For OAuth clients whose connecting user is a member of the CHAR group, when CAS is configured for Kerberos, CAS will fall back to fetching the stored credential and attempting to launh the user's session in the identity of the fetched userID.

CAS SESSION LAUNCH DECISION FLOW

The decision process that the CAS controller uses to determine how to launch a user session is depicted below in Figure 3 and outlined in the following:

- For External Password authentication (where CAS is authenticating the user through the PAM subsystem), the user/password is used to request an OAuth token from SASLogon. If the token can be acquired, the session is launched using the host credential represented by the username, providing the PAM authentication succeeds. PAM authentication may result in a Kerberos credential cache being created for the user, which CAS can leverage for caslibs connecting to Kerberos data sources, such as Hadoop.
- For OAuth authentication, if the user is not a member of the CHAR group (i.e. a host launch is not requested), sessions will launch as the CAS service account's userID, whether CAS is configured for Kerberos is not a factor.
- For OAuth authentication where the user is a member of the CHAR group and CAS is not configured for Kerberos:

    - If the user has access to a valid host credential (username and password) stored in SAS Viya, the CAS session is launched using the stored credential.
    - If the user does not have access to a stored credential, CAS sessions will be directly launched[10] using the username found on the OAuth token, provided it is a valid host account.

- For OAuth authentication where the user is a member of the CHAR group and CAS is configured for Kerberos, CAS requests the client to call back again with a Kerberos connection. If the user's Kerberos credential (TGT) is forwarded, CAS uses it to create a local Kerberos credential cache for the user. CAS launches the session in the user's host identity (based on the username in the OAuth token).
    - CAS should be able to retrieve a Kerberos credential for the end user, and use it to do a host launch of CAS based on the OAuth username, provided it is a valid host account and the user's initial logon to SAS Viya is:
        - authenticated to SASLogon using Kerberos, from a browser or the SAS Admin CLI, where SASLogon's Kerberos principal is configured to allow delegation
        - authenticated to a mobile client or the SAS Admin CLI using username and password, while SASLogon is configured for Kerberos
    - If CAS is unable to acquire the user's Kerberos credential, it will fall back to looking for a SAS Viya stored credential, which if found is used to launch the CAS session. If not found, CAS can still directly launch[10] the session based on the OAuth token's username.

Note: If during authentication, host launch is requested but user or password are not valid when CAS attempts to use them for a host launch (for example, if the host authentication provider is not properly integrated with the configured LDAP provider or an incorrect username or password was supplied or fetched by CAS), the CAS session will fail to launch, producing an ERROR in the log similar to:

> ERROR: `OAuth user sgf2020 is not known on the host but requested host launch.`

[10] Direct launched sessions are launched using the root privilege of the CAS Server process and do not result in PAM authentication taking place. Hence, even if PAM were configured for Kerberos, Kerberos credentials cache for the user will be generated only in cases where CAS has access to a password and uses it for authentication.

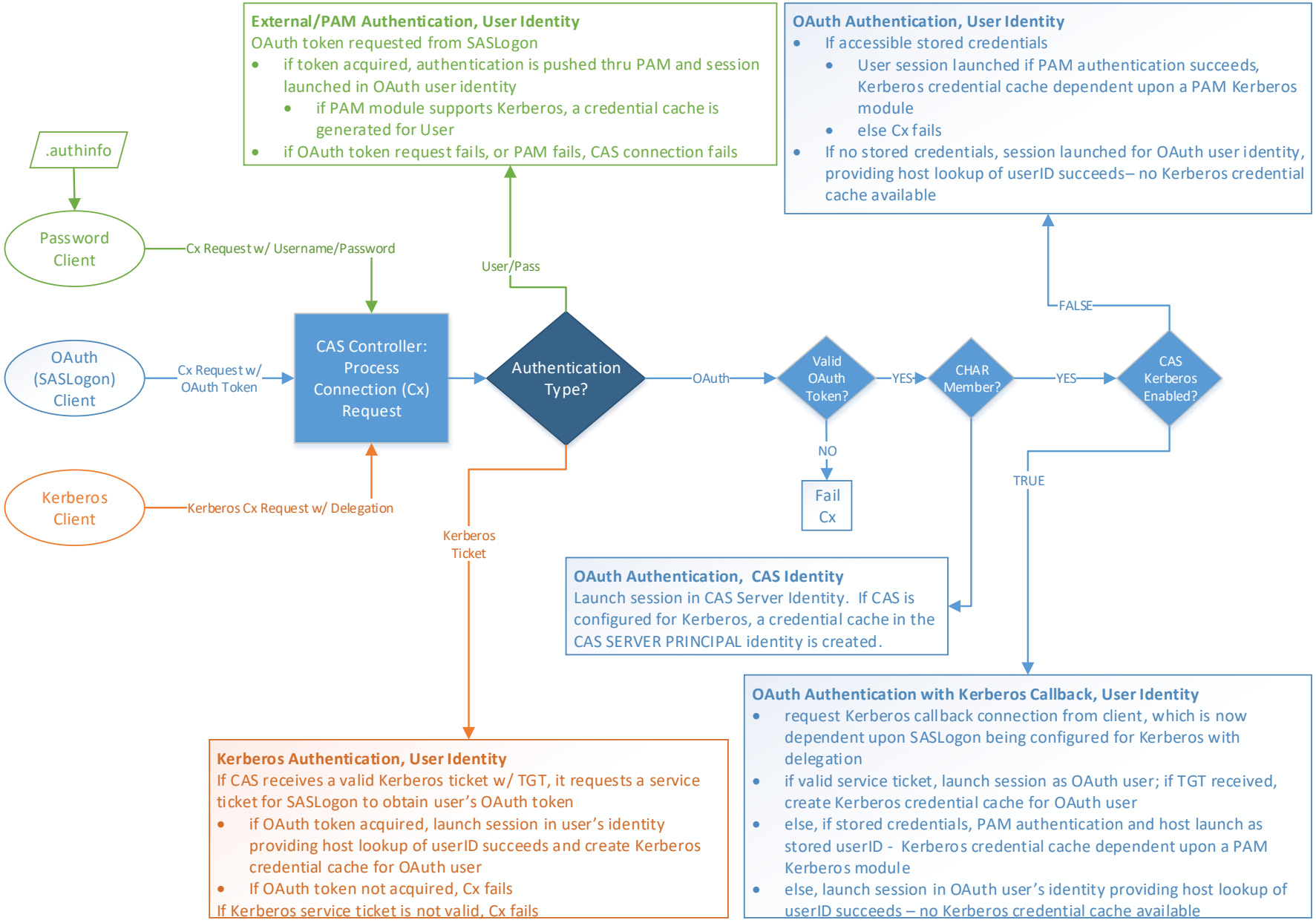

Figure 3: CAS Authentication and Session Launching on Linux

VERIFYING THE CAS SESSION LAUNCH TYPE

The CAS controller log file is a great way to confirm what CAS is doing regarding session launches. CAS writes useful log messages at the default INFO level that vary depending upon the type of client (OAuth, Password, Kerberos) and credentials supplied.

OAUTH CLIENTS

Connecting from SAS Studio (Enterprise) launched Compute Server

INFO [00001850] MAIN NoUser MAIN [tkident.c:1323] - User sgf2020 successfully authenticated using the OAuth authentication provider.

Connecting from SAS Studio (Enterprise) launched Compute Server requesting Host launch

INFO [00002757] MAIN NoUser MAIN [tkidentoauth.c:737] - OAuth user sgf2020 requested host launch.

INFO [00002757] MAIN NoUser MAIN [tkident.c:1323] - User sgf2020 successfully authenticated using the OAuth authentication provider.

INFO [00002757] MAIN NoUser MAIN [tkcsesinst.c:748] - Successfully created session 8b87a764-916c-8a4e-b44d-907efe1fef42.

INFO [00002757] MAIN sgf2020 329 [casgeneral.c:4824] - Launched session controller. Process ID is 28619.

```
sgf2020   28619  61615  0 02:37 ?        00:00:00   /opt/sas/viya/home/SASFoundation/utilities/bin/cas
```

Connecting from an OAuthClient requesting Host launch, Kerberos.Enabled=True

INFO [00000106] MAIN NoUser MAIN [tkidentoauth.c:737] - OAuth user sgf2020 requested host launch.

INFO [00000106] MAIN NoUser MAIN [tkident.c:1323] - User sgf2020 successfully authenticated using the OAuth authentication provider.

INFO [00000106] MAIN NoUser MAIN [tkidentgss.c:976] - User sgf2020@TSAD.UNX.SAS.COM has authenticated using kerberos.

INFO [00000106] MAIN NoUser MAIN [tkident.c:1323] - User sgf2020 successfully authenticated using the Kerberos authentication provider.

INFO [00000106] MAIN NoUser MAIN [tkcsesinst.c:748] - Successfully created session 38644842-d16e-014a-a089-bb0241e4a38f.

INFO [00000106] MAIN sgf2020 9 [casgeneral.c:4824] - Launched session controller. Process ID is 58227.

```
sgf2020   58227  63911  0 04:35 ?        00:00:00   /opt/sas/viya/home/SASFoundation/utilities/bin/cas
```

PASSWORD CLIENTS

Connecting from a Workspace Server: SAS Studio 4.x or SAS Studio 5.2 (Basic)

INFO [00002032] MAIN NoUser MAIN [tkident.c:1323] - User sgf2020 successfully authenticated using the OAuth authentication provider.

INFO [00002032] MAIN NoUser MAIN [tkident.c:1323] - User sgf2020 successfully authenticated using the External PAM authentication provider.

OVERRIDING DEFAULT SPRE SESSION LAUNCHING BEHAVIOR

As discussed earlier, the default launch behavior for SPRE compute sessions on Linux is the opposite to CAS with respect to process owner. The SAS Viya Launcher, responsible for spawning SPE compute sessions, accepts OAuth tokens for authentication from clients such as SAS Studio, and by default launches a compute server process under that host ID. While the objective is still to ensure that the resulting compute session has the requisite access to data, the options for changing the default behavior are different than for CAS. It is important to keep in mind that although the session is launched in user identity by default, the Launcher does not have the user's password when it receives an OAuth token for authentication, and it must leverage its root privilege to impersonate the Viya user. There is no PAM authentication taking place by default.

For Kerberos, the Launcher service drives off the same property as the CAS controller (`sas.compute.kerberos`). Since the OAuth token can't trigger PAM, this is not an option for the compute server to acquire a Kerberos ticket in the user's identity should a PAM Kerberos module be in place. Enabling Kerberos authentication provides the opportunity for the compute session to acquire a Kerberos credential for the end user, if SASLogon is also configured for Kerberos with delegation. SASLogon persists the user's Kerberos credential, making it available to web applications to fetch it and provide it to the Launcher. This Kerberos credential is persisted on the compute server machine, where it can be leveraged by a compute session to access downstream data providers secured with Kerberos, such as Hadoop.

SPRE compute session launches can be augmented in two additional ways. First, the persisting of a SAS Viya credential works much in the same way here as it does for CAS sessions, except that the Launcher uses a stored SAS Viya credential without any need for the user to belong to a special group.

In addition, as of SAS Viya 3.5, SPRE sessions can be configured to launch under a designated service account. However, the option is limited in that the configured service account is associated with a compute server context, and applications such as SAS Studio (Enterprise) support only a single context, which means only one service account can be configured for all SAS Studio users. More information on compute server service accounts can be found in the SAS communities blog post. "SAS Viya 3.5 Compute Server Service Accounts".

Both stored credentials and a configured service account make the password available to the Launcher processes and trigger PAM authentication in the host layer. If PAM is subsequently configured for Kerberos, a Kerberos credential cache becomes available to the launched compute sessions, and they can access downstream Kerberos data providers such as Hadoop in that identity.

Rather than changing the behavior to have SPRE sessions run in the same outbound security context as the SAS Viya user, the previous augmentation scenarios can change the SPRE session to run under some mediated Linux userID.

END-TO-END KERBEROS FOR DATA PROVIDER ACCESS AS END USER

A good example of a practical configuration showing how both CAS and SPRE can be used to connect to a Kerberos data provider, such as Hadoop, is depicted in Figure 4. SAS Visual Analytics is representative of any web application that connects directly to CAS, and SAS Studio (Enterprise) is representative of applications that connect to CAS thru a SPRE session.

The following numbered items explain the numbered reference points on the diagram:

1. SAS Viya Logon using Integrated Windows Authentication (IWA) from the PC's browser
    a. Initial authentication via Kerberos with traditional delegation of User identity
    b. Identification of user and OAuth token creation
    c. Persistence of forward User Kerberos credential
2. OAuth Authentication (not shown) happens first from SAS Visual Analytics to CAS and SAS Studio to SPRE Compute via the Launcher services. If CAS finds the user in the CHAR group, it requests clients to call back with a Kerberos connection. Clients retrieve the user's Kerberos credential from the Credentials service and forward it to CAS and Launcher as part of Kerberos authentication.
3. SAS Studio user initiates a CAS connection from the compute session and OAuth Authentication occurs from the SPRE compute session to CAS, forwarding Kerberos credentials.
4. CAS and SPRE sessions are launched.
    a. SPRE sessions have access to Kerberos credential caches for the SAS Viya user (assuming that Launcher defaults are in place: no stored credential or configured service ID).
    b. CAS sessions have access to a Kerberos credential cache for the SAS Viya user, if and only if the SAS Viya user belongs to the CHAR group.

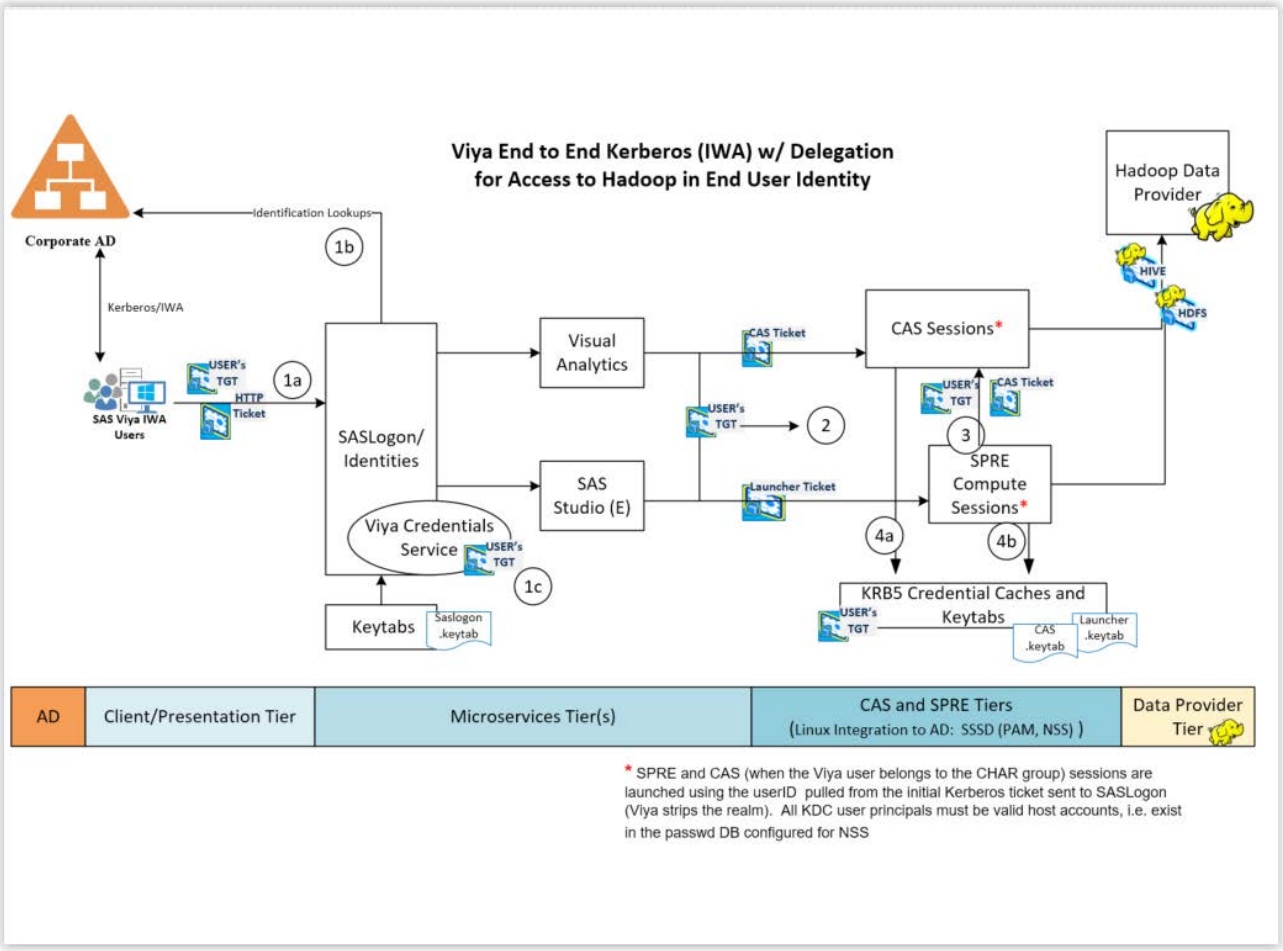

Figure 4: End-to-End Kerberos Authentication Flow

SESSION IDENTITY AND IMPACTS FOR AUTHORIZATION TO DATA

To ensure compliance with your data access policy requires a strategy for CAS and SPRE session launching, for all users. This ensures all users are restricted to their privileged data.

The strategy must integrate any general authentication practices or policy at your organization, with the authentication options in Viya and how they impact the way that CAS and SPRE sessions access data sources. In the second half of this paper we will consider key questions in the decision process and some guiding principles that should be helpful toward developing this strategy.

DECISION PROCESS TO MEET DATA ACCESS OBJECTIVES

GENERAL DECISIONS

Here are some general questions that should be asked when creating the authentication architecture to support secure data access:

1. Is this an ad-hoc environment or a read-only consumer reporting environment?
2. Is the environment comprised of both a SAS 9.x deployment alongside a SAS Viya deployment? Are SAS 9.4M5 (or later) workspace servers usable instead of SAS Viya compute servers for data management activities. This will depend upon the specifics of the SAS®9 deployment. When SAS® Grid Manager is present, it might be a more appropriate choice for heavy workloads.
3. Will resources requiring Kerberos authentication be accessed from SPRE or CAS?
    a. When the answer to this question is yes, the SAS Viya environment should be configured for end-to-end Kerberos as depicted in the "End-to-End Kerberos for Data Provider Access as End User" section.
    b. In a SAS 9.x deployment, full end-to-end Kerberos configuration is not always necessary because the user's password is leveraged to authenticate to compute tier sessions. If PAM is configured for Kerberos, a Kerberos credential cache for the end user will be generated[11].

DECISIONS FOR SPRE COMPUTE SESSIONS

For SPRE compute sessions, the fundamental strategy choice is whether it is necessary to augment the default behavior of having SPRE compute sessions launch in the SAS Viya user's identity. This decision process should consider the following questions:

1. Are the SAS Viya users' identities also valid host accounts?
    a. If they are not, is it acceptable to have many users share compute sessions in a mediated identity?
        i. If a mediated identity is acceptable, should it be a shared credential that is stored in SAS Viya or a configured service account?
        ii. Otherwise, it is necessary to configure SAS Viya to launch SPRE sessions using a personal stored credential.
2. Are the SPRE compute sessions going to require access to data (files, database tables, or Hadoop) in the end-user identity?
    a. If yes, leave the defaults and allow the SPRE sessions to launch using the end-user identity. If Kerberos resources will be required, ensure that SASLogon is also configured for Kerberos.
    b. If no, consider configuring SAS Viya so that all SPRE sessions launch under a mediated identity and ensure that it has no access to any sensitive locations on the SPRE server.

Another important SPRE compute scenario to consider is whether modelers will take advantage of the open-source node available through SAS Model Studio. This node allows a SAS model to leverage open-source code, such as Python. What are the access requirements for any inputs to the Python process? Also, in these cases the open-source

[11] In SAS®9, the presence of Data Loader for Hadoop requires the configuration of IWA or end-to-end Kerberos.

runtime will execute on the SPRE nodes and might require access to resources in a particular identity.

DECISIONS FOR CAS SESSIONS

For CAS sessions, the fundamental strategy decision is whether it is necessary to augment the default behavior of having CAS compute sessions launch in the CAS server's service account identity. This decision process should consider the following questions:

1. Do some users have the ability to access the Linux file systems directly?
2. Will CAS need secure access to path or Kerberos based CAS data sources and for which users or processes?
3. Which users need access to the data provider layer in their SAS Viya identity?

GENERAL DESIGN PRINCIPLES

Due to the high variability in SAS Viya usage patterns and data access policy needs, it is difficult to prescribe a one size fits all solution. However, some general authorization design principles can be established.

KERBEROS DATA SOURCES

SPRE and CAS sessions needing access to Kerberos data sources in the user's identity must have their launcher process (Launcher service or CAS, respectively) trusted for delegation and launched in the end user's identity. For CAS, this means that ensuring the users are members of the CHAR custom group. Even a host with PAM configured for Kerberos is not likely to be helpful because under default SAS Viya OAuth authentication the lack of a password known to CAS results in a direct launch of the session by the CAS controller leveraging its Linux root privilege.

ALTERNATE SAS VIYA AUTHENTICATION METHODS AND HOST INTEGRATION

When SAS Viya has been configured to use an alternate authentication mechanism, such as SAML or Open ID Connect, it is sometimes the case that the host Linux systems are not integrated with the LDAP registry that SAS Viya is configured to use. As a result, the username from the SAS Viya logon may not have a corresponding host account. In this scenario, SPRE compute sessions may need to be configured to launch in the host account of a mediated identity and CAS sessions in this scenario might need to remain limited to launching as the CAS service account's userID.

JOB SCHEDULING IN A KERBEROS ENVIRONMENT

Job scheduling using SAS Viya in a Kerberos environment is challenged by the possibility of persisted Kerberos credentials expiring before the job is run. If the CAS session cannot access a valid Kerberos credential, but requires access to a Kerberos data provider, it will be unable to access the data source. One possible solution to these challenges is to dedicate a job execution user and persist a credential specifically for executing scheduled jobs, add the user to the CHAR group, and ensure the host system's PAM framework is configured for Kerberos. With all of these conditions in place, the job (when launched) will be able to make outbound connections to Kerberos data providers.

SPRE LOCKDOWN AND CAS WHITELISTING

When users are launching sessions in their own identity, consider using the LOCKDOWN option in SPRE and white listing in CAS for caslibs. Both of these features are designed to limit the user's access to the file system through their sessions. This results in fewer

locations to manage permissions.

A common concern with data policy compliance is to ensure that users do not place sensitive data they may be privileged to in locations that are open publicly, such as the /tmp file system, where other non-privileged users could see it. These two features limit the librefs and caslibs that developers can assign, and thus limits where data can be saved.

PATH CASLIB SUBDIRECTORY INCLUSION

When global caslibs of type PATH or DNFS are assigned, the CAS security administrator should consider marking them to include all subdirectories to be considered as part of the data source locations for the caslib. CAS does not allow two or more caslibs to reference the same data source paths. When subdirectories are included in a caslib definition, CAS will consider all subdirectory locations within the caslib's defined path, as being the same as the location in its definition.

Having subdirectories included reduces the scope of pathways to ensure and reduces the burden on the security modeler. A CAS developer cannot assign a path based caslib in their session, at a subdirectory location, in order to bypass an access control preventing his or her access to the parent location. CAS would prevent the allocation of the new caslib if the security admin has created a global caslib at a parent location.

While the file system permissions are also a factor controlling access to the subdirectory, and should be set correctly, the subdirectory inclusion feature can act as insurance should the file system paths be incorrect.

JUST- IN-TIME CAS TABLE LOADING

When considering CAS session identity and security contexts, one must keep in mind that SAS Visual Analytics includes a feature to load CAS tables if a report requires data from a table that is not loaded. If CAS sessions have been engineered to launch in the Viya user's identity, and just-in-time loading is required, then it is necessary to ensure the user identity can access both the report and the data sources for any caslibs used in the report.

GENERAL SOLUTIONS

There are two SAS Viya usage patterns that turn up repeatedly when working with customers. First, the need for ad hoc usage of SAS Viya where users have freedom to explore and analyze data looking for valuable insights. The second is the need for more stable environments that can surface high value reporting metrics to key stakeholders in the organization. Generalized solutions for an ad hoc environment and a reporting environment are useful starting points.

PRACTICES FOR CONSUMER REPORTING ENVIRONMENTS

In a production reporting environment, a simplistic best practice might be to disallow any sessions from launching in user identity. The premise here is that CAS is fully loaded with all the tables published reports will require, so no "just-in-time" loads are needed. However, loading CAS tables might require CAS to launch in user identity to access certain data sources (unless the CAS service ID could be ensured access to all data providers).

Additionally, the compute servers may be restricted to launch as a mediated service ID whose outbound access can be properly controlled.

Limiting all CAS sessions to launch as the CAS service ID identity can be accomplished by making two configuration choices:

1. Do not allow a CHAR group to exist, or ensure it has no report consumers as members.

2. Modify the CAS property that enables its Authentication providers[12], dropping the element that allows for External Authentication. This will prevent connections from SAS®9 or SAS Viya clients that supply only a password as well as open-source clients using a password.

If the CAS data loading jobs require data provider access in a special identity other than the CAS service account ID, it might be necessary to have the ETL process authenticate using OAuth or Kerberos and be the sole member of the CHAR group.

Real world scenarios rarely line up to simple best practices. Nonetheless, it can be a useful design technique to understand and envision a straight forward solution, identify exceptions that are absolutely required, and adjust the simpler solution as little as possible to meet the mandatory exceptions while keeping it manageable.

PRACTICES FOR AD HOC ENVIRONMENTS

Ad hoc environments present different challenges, and typically have scenarios where developers, data scientists, or analysts need access to data source providers using their own identity. As we have seen, there are several options to engineer both SPRE compute and CAS sessions to launch in the user's own identity, provided Viya is authenticating using simple LDAP authentication or Kerberos authentication. When Viya is authenticating using SAML and the user's identity is not valid on the host, it is possible to persist an identity in Viya that might be suitable for providing the data scientists the type of data access they require.

In an ad hoc environment, it is usually advisable to limit the CHAR membership to only the users who require it.

THOUGHTS FOR WINDOWS ENVIRONMENTS

The consistency in Windows environments (for example, all CAS sessions launching in the user's identity) eliminates many of the concerns and choices available in Linux environments. This consistency also allows the authorization modeler to focus on ensuring that directories for SPRE and CAS file-based data sources are structured appropriately with the correct access for all end users.

For data providers requiring Kerberos access, the modeler can operate on the following premises:

- If the end user's Kerberos credentials exist for CAS sessions, the session will launch under the user's SAS Viya identity.
- If CAS fails to acquire the end user's Kerberos credentials, it is due to a temporary configuration problem. The CAS session launch will fail for regular users. For members of the SAS Viya SAS Administrators group, CAS sessions will launch as the CAS service ID.

[12] In the casconfig.lua configuration file, the cas.provlist property (cas.provlist = 'oauth.ext.kerb') specifies the authentication providers that CAS supports.

CONCLUSION

SAS Viya is a complex application that empowers its users with many ways to derive insight from data and simultaneously opens up many pathways to both external data sources and data that is resident within the CAS analytics engine.

Data access policy compliance is also a complex endeavor as well as an ongoing concern for organizations in today's technological world.

Engineering your SAS Viya environment to both empower users for insight while meeting the strict demands of data policy compliance requires a thorough understanding of how SAS Viya enables access to data and offers features to control this access. In this paper, we have thoroughly discussed how SAS Viya enables access to data sources in an "out-of-the-box" deployment. This paper also looked at all the ways a default deployment can be augmented to better align and integrate into a variety of technology contexts.
There is no simple solution, but hopefully the ideas presented will be useful in guiding the design activities necessary to achieve a solution customized for your organization's needs.

REFERENCES

Rogers, Stuart. "SAS Viya 3.4 Kerberos with CAS". 2018. December 22, 2018. https://communities.sas.com/t5/SAS-Communities-Library/SAS-Viya-3-4-Kerberos-with-CAS/ta-p/523246

Rogers, Stuart. 2020. "SAS Viya 3.5 Compute Server Service Accounts". https://communities.sas.com/t5/SAS-Communities-Library/SAS-Viya-3-5-Compute-Server-Service-Accounts/ta-p/620992. Last modified January 29, 2020.

SAS Institute Inc. 2019. *SAS Micro Analytic Service 5.4: Programming and Administration Guide*. Cary, NC: SAS Institute Inc. https://documentation.sas.com/?docsetId=masag&docsetTarget=p0gehkxmerovitn1w5nv6yvgpws5.htm&docsetVersion=5.4

SAS Institute Inc. 2019. *SAS Viya 3.5 Administration*. Cary, NC: SAS Institute Inc. https://documentation.sas.com/?cdcId=calcdc&cdcVersion=3.5&docsetId=calwlcm&docsetTarget=home.htm

ACKNOWLEDGMENTS

I would like to thank several colleagues who provided valuable review of the technical subject matter: Angie Hedberg, Larry Noe, Andrew Weida, and Eric Davis.

I would also like to acknowledge the value received from the many timely blog postings by Stuart Rogers, which have informed me on several key areas with respect to SAS Viya authentication.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Philip Hopkins
SAS Institute Inc.
philip.hopkins@sas.com