

# SAS® 9.4 on Microsoft Windows: Unleashing Kerberos on Apache Hadoop

Stuart J Rogers, SAS Institute Inc., Cary, NC

## ABSTRACT

Do you maintain a Microsoft Windows server providing your organization's SAS® 9.4 environment? Are you struggling to get Kerberos authentication out to Apache Hadoop working correctly for your SAS® Enterprise Guide® or SAS® Studio users? This paper outlines the key steps for enabling Kerberos authentication between your users, your SAS 9.4 deployment on Microsoft Windows server, and your Hadoop data sources. Learn about the configuration changes you need to make, and learn about some effective trouble-shooting techniques to get your environment working and your users happy.

## INTRODUCTION

Kerberos authentication to Hadoop is becoming a common requirement for many organizations. Materials so far (Rogers and Keefer, 2014) have focused heavily on Linux and other UNIX operating systems, with the expectation that Kerberos in Microsoft Windows just works. In this paper, I want to focus solely on Microsoft Windows. We will examine how to enable Kerberos authentication, looking first at the different authentication steps and then discussing at a high level the steps you'll need to complete.

Once we understand the overall process, we will look at some of the specific configuration changes you'll need to complete to get your environment operating correctly. We will also review some specific challenges you'll face when working with a Microsoft Windows environment. This will cover some of the security technology introduced by Microsoft that can limit your ability to easily get the environment working as expected.

Finally, we will examine some troubleshooting techniques. We will look at some different types of logging you can enable to gain a better understanding of what is happening in your environment. You will see some different ways of testing your environment and we will finish with some common issues and the steps you can take to resolve them.

## ENABLING KERBEROS AUTHENTICATION

This section of the paper outlines the key steps for enabling Kerberos authentication, between your users, your SAS 9.4 deployment, and your Secure Hadoop Cluster.

## AUTHENTICATION STEPS

Before you consider how Kerberos authentication is leveraged with the SAS 9.4 environment, you need to remind yourselves of the different authentication steps the environment follows. There are two types of clients, and the authentication steps are different for each type of client. The first type of client is the SAS 9.4 desktop client, such as SAS Enterprise Guide. Figure 1 illustrates the three different steps that are executed.

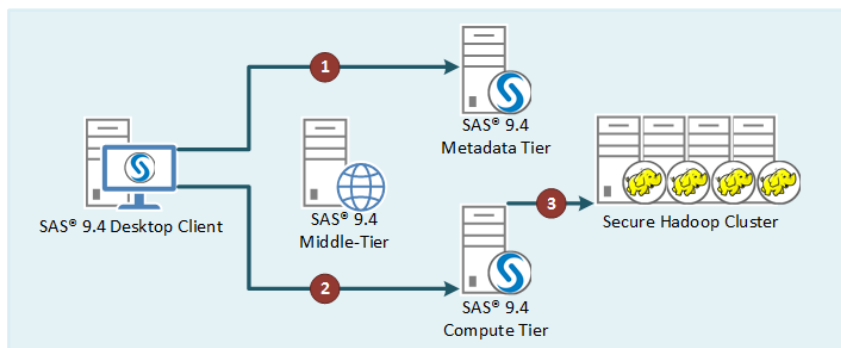


Figure 1. Authentication from Desktop Client

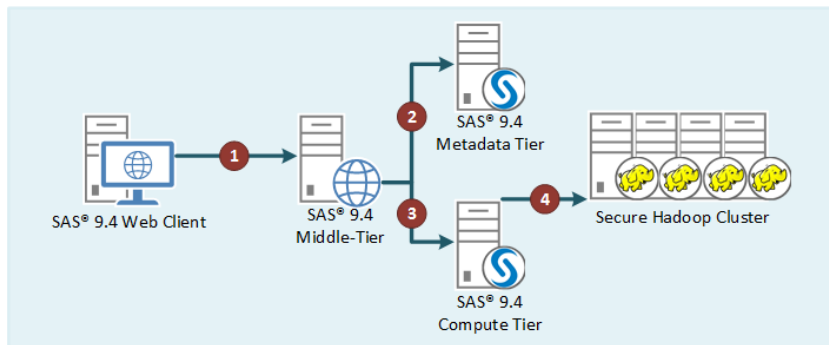
The SAS 9.4 desktop client initially authenticates to the SAS® 9.4 Metadata Server, as shown in step 1. The client reads the metadata defining the connection to the SAS 9.4 compute tier, where the SAS 9.4 Object Spawner is running. In step 2 the SAS 9.4 desktop client authenticates to the SAS Object Spawner and the SAS Workspace Server session is launched. The SAS 9.4 desktop client submits SAS code to the SAS Workspace Server session, and statements in this code include a connection to the Secure Hadoop Cluster, as shown in step 3.

You can see that we have the following three separate points of authentication:

1. The SAS 9.4 Metadata Server,
2. The SAS 9.4 Object Spawner, which launches the SAS Workspace Server,
3. The Secure Hadoop Cluster.

The authentication to the SAS 9.4 Metadata Server is separate from the authentication to the SAS 9.4 Object Spawner even if the two processes are running on the same host. The authentication from the SAS 9.4 Object Spawner flows onto the authentication to the Secure Hadoop Cluster via the SAS Workspace Server.

The second type of client is a SAS 9.4 Web Client, such as SAS Studio. The authentication steps followed by a SAS 9.4 Web Client are shown in Figure 2.



**Figure 2. Authentication from Web Client**

You open SAS Studio in your browser and are authenticated to the SAS 9.4 middle tier, as shown as step 1. The SAS 9.4 middle tier authenticates to the SAS 9.4 Metadata Server shown in step 2. The SAS 9.4 middle tier obtains information about the SAS 9.4 compute tier from the SAS Metadata Server. In step 3 the authentication from the SAS 9.4 middle tier to the SAS Object Spawner is shown and the SAS Workspace Server session is launched. The SAS Object Spawner and the SAS Workspace Server run on the SAS 9.4 compute tier. The SAS 9.4 web client submits SAS code to the SAS Workspace Server session, and statements in this code include a connection to the Secure Hadoop Cluster, as shown in step 4.

In this case you can see we have the following four separate points of authentication:

1. The SAS 9.4 middle tier,
2. The SAS 9.4 Metadata Server,
3. The SAS 9.4 Object Spawner, which launches the SAS Workspace Server,
4. The Secure Hadoop Cluster.

The authentication flows from the SAS 9.4 middle tier to both the SAS 9.4 Metadata Server and SAS 9.4 Object Spawner. From the SAS 9.4 Object Spawner, the authentication flows onto the Secure Hadoop Cluster via the SAS Workspace Server.

## INTEGRATED WINDOWS AUTHENTICATION FROM SAS 9.4 DESKTOP APPLICATIONS

Integrated Windows Authentication (IWA) from the SAS 9.4 desktop applications requires both the SAS 9.4 metadata tier and SAS 9.4 compute tier to be configured to enable IWA. Enabling IWA for the SAS 9.4 desktop applications does not enforce IWA, the clients can still connect using a user name and password. The concept of IWA for the SAS 9.4 desktop applications is covered in the SAS® 9.4 Intelligence Platform: Security Administration Guide. Further details about using Kerberos with SAS 9.4 can be found in the paper “Kerberos and SAS® 9.4: A Three-Headed Solution for Authentication” (Rogers 2013).

As you have seen from the section above, during the authentication steps, you need to have the authentication flow from the SAS 9.4 Object Spawner, which launches the SAS Workspace Server, to the Secure Hadoop Cluster. In Figure 1, authentication flows from step 2 to step 3. In Kerberos terms, you want the authentication to be delegated, or forwarded, from the SAS 9.4 desktop client to the SAS Workspace Server. The delegation enables the SAS Workspace Server to use Kerberos authentication to connect to the Secure Hadoop Cluster. Further details about Kerberos delegation with SAS 9.4 can be found in the paper “Kerberos Delegation with SAS® 9.4” (Rogers 2016).

You know from the paper “Kerberos Delegation with SAS® 9.4” (Rogers 2016), that you can have either a simple topology, with a single instance of the SAS 9.4 Metadata Server and a single SAS Workspace Server, or you could have an advanced topology where the SAS 9.4 Metadata Server is clustered. You know that with the simple topology, both the SAS 9.4 Metadata Server and SAS 9.4 Object Spawner will register their Service Principal Name (SPN) against the local computer account in Active Directory, regardless of whether these components are running on the same host or different hosts.

While with an advanced topology the SAS 9.4 Metadata Server cluster members require you to manually register the SPN, for each cluster member, against the single service account running the instances. If you locate an instance of the SAS 9.4 Object Spawner with one of the SAS 9.4 Metadata Server cluster members, you must remember to run the SAS 9.4 Object Spawner using the same service account. This will ensure that duplicate SPNs are not registered against both the computer account and service account. Further details about the advanced topology are covered in the paper “Kerberos Delegation with SAS® 9.4” (Rogers 2016).

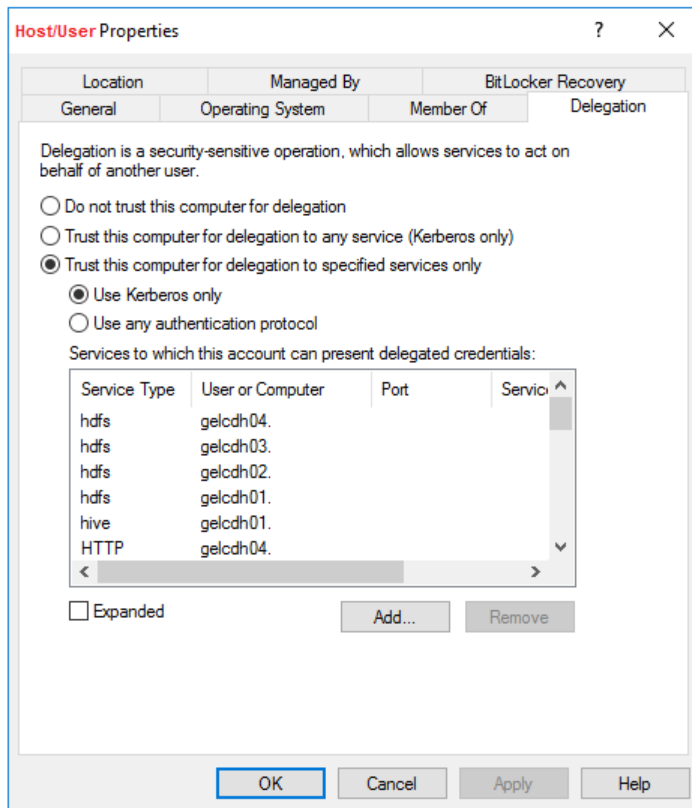
From Figure 1 you can see that only the authentication from the SAS 9.4 Object Spawner needs to be delegated. There is no second authentication from the connection to the SAS 9.4 Metadata Server. The object, either computer account or service account, that has the SPN registered for the SAS 9.4 Object Spawner will require additional settings. These additional settings are that the Active Directory object that has the SPN registered against it must be trusted for delegation.

The SAS 9.4 Object Spawner supports both constrained delegation and unconstrained delegation. Unconstrained delegation means that you are allowing the SAS Workspace Server launched by the SAS 9.4 Object Spawner to make a second connection using Kerberos, as the end user, to any resource. Constrained delegation, however, limits the second connection to specific services. Constrained delegation is an extension added by Microsoft to the Kerberos protocol. When you configure constrained delegation, you must list the services, which are valid for the second connection. Figure 3 shows the delegation settings for a computer account in Active Directory.

From Figure 3, the selection item “**Trust this computer for delegation to any service (Kerberos only)**” is the option you want to select for unconstrained delegation. However, what is illustrated in Figure 3 are the settings for constrained delegation, since these are more complex. The option “**Trust this computer for delegation to specified services only**” is selected and in the table, you can see the list of services that Kerberos delegation can take place with. To add further services, you select the **Add** button and then search for the Active Directory object with the SPN, for the target service, registered against it. This Active Directory object could be a user account or computer account. Once you have selected the object, you are presented with a list of SPNs registered against the object. You can then select the individual services you want. This means that even if you select a computer account you do not have to enable Kerberos delegation to all services registered against that computer account.

In Figure 3, all the registered SPNs for a four node Secure Hadoop Cluster have been selected and hence the HDFS, Apache Hive and HTTP service types are shown. This means that the SAS Workspace

Server will be able to use Kerberos delegation to connect to any of the Secure Hadoop services running on any of the nodes of the Secure Hadoop Cluster.



**Figure 3. Active Directory Delegation Settings**

Finally, as shown in step 3 in Figure 1, you want to use Kerberos authentication from the SAS Workspace Server to the Secure Hadoop Cluster. With the update, you have made to the constrained delegation the Kerberos credentials for the end user are available to the SAS Workspace Server. The only other consideration is for the encryption strength used in the Kerberos tickets. If AES256-bit encryption is used, then the SAS® Private Java Runtime Environment used by the SAS Workspace Server will need to have the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files deployed. This is required since the connection between the SAS Workspace Server and Hadoop uses Java. Further details about deploying the JCE Unlimited Strength Jurisdiction Policy Files is given below.

## IWA FROM SAS 9.4 WEB APPLICATIONS

IWA from the SAS 9.4 web applications requires all the SAS 9.4 tiers to be configured for Kerberos authentication. This means you must complete the configuration detailed above for the SAS 9.4 metadata tier and SAS 9.4 compute tier, as well as configuring the SAS 9.4 middle tier. You want to leverage Kerberos authentication for all four steps shown in Figure 2. You can see from Figure 2 that authentication flows from step 1 to both steps 2 and 3, then authentication flows from step 3 to step 4. This section focuses on step 1 of the IWA process: Kerberos authentication from the web client to the SAS 9.4 middle tier. Steps 2-4 for Kerberos authentication from the web clients is the same as for desktop applications.

As with the SAS 9.4 compute tier, you can configure the SAS 9.4 middle tier to still accept non-Kerberos connections. Details are given in the paper “An Advanced Fallback Authentication Framework for SAS® 9.4 and SAS® Visual Analytics” (Li and Roda 2014). Leveraging the Fallback Authentication Framework will greatly simplify the configuration you need to complete to support IWA, even if your end users only

use Kerberos. However, leveraging the Fallback Authentication Framework is not automatic the way it was for the SAS 9.4 compute tier. You must complete the manual configuration detailed here.

You must manually register the SPN for the SAS 9.4 middle tier against a user object in Active Directory. This user object or service account is used to represent the SAS 9.4 middle tier in Active Directory. This service account will not be used to run the operating system process for the SAS 9.4 Web Application Server. Therefore, another mechanism must be used to provide the long-term Kerberos key to the SAS 9.4 Web Application Server. The long-term Kerberos key is provided in a Kerberos Keytab. More details about generating the Kerberos Keytab can be found in the paper “Tips and Best Practices for Configuring Integrated Windows Authentication” (Roda 2016) and in the paper “Kerberos Delegation with SAS® 9.4” (Rogers 2016).

When you have created the service account, registered the SPN, and created the Kerberos Keytab you will be ready to configure the SAS 9.4 Web Application Server. After completing the steps detailed in the paper “Kerberos Delegation with SAS® 9.4” (Rogers 2016), you must complete the following steps:

1. Update the SAS Logon Manager application, both uncommenting the error page defined in the `web.xml` and adding the valve and realm definitions.
2. Update the SAS® Web Application Server configuration, adding to the `jaas.config`.
3. (Optional) Provide a Kerberos configuration file.
4. Possibly, deploy the JCE Unlimited Strength Jurisdiction Policy Files.

You will always need to complete the first two items. For the third item, Java will attempt to find the location of the Kerberos Key Distribution Center with DNS queries. However, if you require Kerberos cross-realm authentication or the DNS queries are not successful you can deploy a Kerberos configuration file to provide this information. Finally, if you are using AES256-bit encryption within the Kerberos tickets you will need to deploy the JCE Unlimited Strength Jurisdiction Policy Files to the SAS Private Java Runtime Environment used by the SAS 9.4 middle tier. These policy files enable Java to process this higher level of encryption.

From Figure 2 you can see that authentication needs to flow from the SAS 9.4 middle tier (step 1) to both the SAS 9.4 Metadata Server (step 2) and SAS 9.4 Object Spawner (step 3). As before the flowing of authentication in Kerberos terms is either delegation or forwarding. The SAS 9.4 middle tier does not support constrained delegation. This is a limitation of the version of Java used by the SAS 9.4 Web Application Server. Therefore, you must enable unconstrained delegation for the service account you registered the SPN for the SAS 9.4 Web Application Server.

## **KERBEROS OUTBOUND ONLY**

While not the focus of this paper, one mechanism some people have used to provide access to a Secure Hadoop Cluster is to just focus on the last step in the authentication. This is shown as step 3 in Figure 1 and step 4 in Figure 2. So, Kerberos is only used on the connection to the Secure Hadoop Cluster and other authentication mechanisms are used for all the other steps. This simplifies the setup you need to complete. It also possibly exposes the other authentication steps to attacks that the use of Kerberos would mitigate.

For the connection from the SAS Workspace Server to the Secure Hadoop Cluster you just need to ensure that two items are configured. First you need to ensure the Kerberos credentials of your end user are available to the SAS Workspace Server. Since the SAS Workspace Server is running on Microsoft Windows, as long as the host is joined to a domain this should be true. Care needs to be taken if the domain for the SAS Workspace Server is different from the domain of the end user. In this case the cross-realm trusts will need to already be in place.

Additionally, as discussed above, you might need to deploy the JCE Unlimited Strength Jurisdiction Policy Files to the SAS Private Java Runtime Environment used by the SAS Workspace Server. This is required if the Kerberos tickets use AES256-bit encryption.

## CONFIGURATION CHANGES TO SUPPORT KERBEROS

This section of the paper presents the configuration changes and some Microsoft Windows specific challenges you will face enabling Kerberos authentication for your users.

### SAS 9.4 CONFIGURATION CHANGES

To make it easier for you to follow, I will split the SAS 9.4 configuration changes into the following three sections:

- One for the SAS 9.4 metadata tier and SAS 9.4 compute tier,
- One for the SAS 9.4 middle tier,
- One for the SAS Private Java Runtime Environment.

### SAS 9.4 Metadata and Compute Tier

IWA from SAS 9.4 desktop applications can either be configured during the initial deployment of your SAS 9.4 environment, by selecting the option to enable IWA, or after the deployment is complete. If you configure IWA after the initial deployment you must update the configuration, within metadata, for SAS 9.4 Metadata Server and SAS Workspace Server.

To update the configuration for SAS Workspace Server from within SAS Management Console, find the Logical Workspace Server definition. This is normally **SASApp – Logical Workspace Server** under **Environment Management** ⇒ **Server Manager** ⇒ **SASApp**. You right-click the Logical Workspace Server definition and select **Properties**. On the **Options** tab, you must change the **Security package** to **Negotiate** and ensure that the **Security package list** is “**Kerberos, NTLM**” – which is shown in Figure 4.

By default, the **Service principal name (SPN)** field will be blank, and for most cases you can leave this blank. However, if your SAS 9.4 middle tier is also going to be configured for Kerberos authentication, but is in a different Kerberos realm, then you should specify the full SPN value. In this case it is very important that you include the Kerberos realm in the SPN. This will have the format **SAS/<fully.qualified.hostname>@<Kerberos Realm>**. Most often the Kerberos realm will be the uppercase version of your Windows Domain, but this is not always the case.

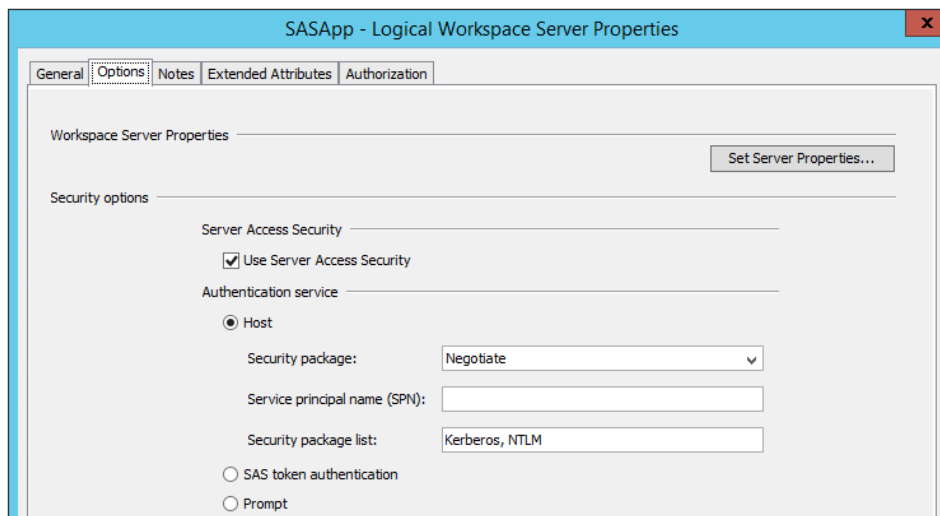


Figure 4. SAS 9.4 Logic Workspace Server Properties

You must make the same changes to the Logical Metadata Server definition. This is normally **SASMeta – Logical Metadata Server** under **Environment Management** ⇒ **Server Manager** ⇒ **SASMeta**.

If you have a complex Kerberos realm structure, especially with non-hierarchical trusts, you might find that connections to the Secure Hadoop Cluster fail. This could be due to issues mapping the network computer name of the Hadoop nodes to the correct Kerberos realm. You can provide the Java process called by SAS Workspace Server or SAS 9.4 Metadata Server with a Kerberos configuration file. The Kerberos configuration can then provide the correct network name to realm mappings and identify the correct trust relationships. To provide the Kerberos configuration file update the `sasv9.cfg` file located in `<SASHOME>\SASFoundation\9.4\nls\<language>\.` Add the following to the `-JREOPTIONS:`

```
-Djava.security.krb5.conf=C:\<Path to file>\krb5.ini
```

After making these changes, you must restart SAS 9.4 Metadata Server and SAS 9.4 Object Spawner.

## SAS 9.4 Middle-Tier

As discussed above, the recommended approach to configure IWA for the SAS 9.4 middle tier is to use the Fallback Authentication Framework. To leverage the Fallback Authentication Framework, you need to make the following changes.

First, you must edit the `web.xml` file to uncomment the error page, located near the bottom of the file. You can either edit just the original file and then rebuild and re-deploy the applications. Or you can edit the file in both the installation files and the deployed location. Editing in both places is the recommended practice and will ensure the changes remain if the application is later rebuilt. The two files that need to be changed are as follows:

- `<SASHOME>\SASWebInfrastructurePlatform\9.4\Configurable\wars\sas.svcs.logon\WEB-INF\web.xml.orig`
- `<SASCONFIG>\Lev1\Web\WebAppServer\SASServer1_1\sas_webapps\sas.svcs.logon.war\WEB-INF\web.xml`

Next, you need to define the valve and realm for SAS 9.4 Logon Manager. The valve for SAS 9.4 Logon Manager defines the interceptor that performs the authentication and is given by the following:

```
<Valve
  className="com.sas.vfabricatcsvr.authenticator.SasFallbackAuthenticatorValve
  " authMethod="SPNEGO" />
```

The realm for SAS 9.4 Logon Manager provides immediate authentication for any user who has already established a Kerberos connection and is given by the following:

```
<Realm className="com.sas.vfabricatcsvr.realm.GSSContextEstablishedRealm"
  allRolesMode="authOnly" />
```

You must add these two items, valve and realm, to the context for SAS 9.4 Web Application Server. You will edit the file `<SASHOME>\Lev1\Web\WebAppServer\SASServer1_1\conf\context.xml` and add the items before the closing `</Context>` tag.

You could add the valve and realm just to SAS 9.4 Logon Manager, but this presents issues when trying to obtain additional debug logging information. If you choose to add the valve and realm to SAS 9.4 Logon Manager, the recommended practice is to edit the following two files:

- `<SASHOME>\SASWebInfrastructurePlatform\9.4\Static\wars\sas.svcs.logon\META-INF\context.xml`
- `<SASCONFIG>\Lev1\Web\WebAppServer\SASServer1_1\conf\Catalina\localhost\SASLogon.xml`

Again, the valve and realm definition should be added before the closing `</Context>` tag.

The final change you must complete is to the configuration of SAS Web Application Server. Edit the `jaas.config` file located in the `<SASCONFIG>\Lev1\Web\WebAppServer\SASServer1_1\conf` directory. You need to make two changes to the `jaas.config` file. You must first add the following Kerberos logon module definition:

```
com.sun.security.jgss.krb5.accept {
    com.sun.security.auth.module.Krb5LoginModule required
    doNotPrompt=true
    isInitiator=false
    principal="HTTP/hostname.example.com@EXAMPLE.COM"
    useKeyTab=true
    keyTab="C:/path-to-hostname.keytab"
    storeKey=true;
};
```

The principal specified should match the User logon name of the service account in Active Directory (also the `userPrincipalName` attribute on the account). This might be the same as the service principal name, especially if `ktpass` was used to generate the keytab, but is commonly not the same. The path to the keytab should use forward slashes, even on Windows systems.

Second, you must add two options to the PFS entry at the top of the `jaas.config` file:

```
"idpropagation"="sspi"
"sspisecuritypackagelist"="KERBEROS"
```

Add these options after the `trustedpw` and before the semicolon. The complete PFS entry will look like the following:

```
PFS {
    com.sas.services.security.login.OMILoginModule required
    "port"="8561"
    "host"="hostname.example.com"
    "repository"="Foundation"
    "domain"="DefaultAuth"
    "aliasdomain"="DefaultAuth"
    "debug"="false"
    "trusteduser"="sastrust@saspw"
    "trustedpw"="{sas002}XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
    "idpropagation"="sspi"
    "sspisecuritypackagelist"="KERBEROS"
;
};
```

These options are required for the Kerberos delegation to take place from the SAS 9.4 middle tier to both the SAS 9.4 metadata tier and SAS 9.4 compute tier. You can obtain more details about these steps in the papers “Tips and Best Practices for Configuring Integrated Windows Authentication” (Roda 2016) and “Kerberos Delegation with SAS® 9.4” (Rogers 2016)

You will need to update the `jaas.config` file for each instance of SAS Web Application Server. Also, the changes to the deployed files for SAS 9.4 Logon Manager will need to be repeated for any SAS 9.4 middle-tier cluster members.

As an option, you can provide a Kerberos configuration file. If you name the file `krb5.ini` and place this in SAS Web Application Server configuration directory, `<SASCONFIG>\Lev1\Web\WebAppServer\SASServerN_M\conf`, it will be automatically detected. You might require a Kerberos configuration file if the DNS lookups for the Kerberos Realm and Key Distribution Center



(KDC) does not operate correctly on your network. Also, you might require a Kerberos configuration file if you have a non-hierarchical trust between Kerberos realms. More details about cross-realm authentication can be found in the paper “Kerberos Cross-Realm Authentication: Unraveling the Mysteries” (Rogers 2017).

After making these changes, you must restart the SAS Web Application Server instances. If you have a clustered SAS 9.4 middle tier, all instances of SAS Web Application Server should be restarted since you changed the `jaas.config` file for all instances.

## SAS Private Java Runtime Environment

You should always ensure that SAS Private Java Runtime Environment is up-to-date. SAS provides updates to the Java 7 Runtime Environment. Details about the updates that are available can be found on the SAS Support site here: <https://support.sas.com/en/security-bulletins.html#section2>. You should plan to always deploy the latest version of SAS Private Java Runtime Environment.

As discussed in the previous section, if your Kerberos tickets leverage AES256-bit encryption, you will need to deploy the JCE Unlimited Strength Jurisdiction Policy Files. If you are running the latest release of SAS Private Java Runtime Environment, you should obtain the Zulu® Cryptography Extension Kit from here: <https://www.azul.com/products/zulu-and-zulu-enterprise/zulu-cryptography-extension-kit/>. If you are running a previous release of SAS Private Java Runtime Environment, you should obtain the Oracle® JCE Unlimited Strength Jurisdiction Policy Files from here: <http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html>.

When reviewing and updating your SAS Private Java Runtime Environment (JRE), you should ensure that your use of the stronger encryption types conforms to your local country laws, and you should seek legal counsel to determine your regional laws and limits.

To deploy the Unlimited Strength Jurisdiction Policy Files, you need to extract the `local_policy.jar` and `US_export_policy.jar` files that were obtained when you downloaded the Java Cryptography Extension policy files (as described above). These two files should be placed in the `<SASHOME>/SASPrivateJavaRuntimeEnvironment\9.4\jre\lib\security` folder. The SAS processes should be restarted after deploying these two files.

## BROWSER CONFIGURATION CHANGES

You will need to ensure that the browser that is used by your end users to access the SAS 9.4 web applications is correctly configured to enable both Kerberos authentication and delegation.

### Internet Explorer

Microsoft Internet Explorer will only perform Kerberos authentication to sites listed in “**Trusted sites**” and will only perform Kerberos delegation to sites listed in “**Local intranet**”. You should ensure the URL for the SAS 9.4 middle tier is listed, either directly or by wildcard, under the “Local Intranet” sites. You can perform this via the “**Security**” tab on the “**Internet Options**” within the Control Panel. In addition, there is an advanced setting on the “**Internet Options**” for “**Enable Integrated Windows Authentication**”. You will need to ensure this is also selected.

### Google Chrome

The Google Chrome browser follows most of the settings for Microsoft Internet Explorer. However, by default it does not implement Kerberos delegation. This means you will be able to complete the authentication step 1 in Figure 2 (browser to SAS 9.4 middle tier), but not the follow-on steps 2 and 3 (delegated authentication to SAS 9.4 metadata tier and SAS 9.4 compute tier). You will need to add a Windows registry entry to enable Kerberos delegation with Google Chrome. You will need to add **Software\Policies\Google\Chrome\AuthNegotiateDelegateWhitelist** to either the **HKEY\_LOCAL\_MACHINE** or **HKEY\_CURRENT\_USER** registry keys. The value is a string representing the URL of sites Kerberos delegation should be enabled for. You can find more details about this Windows registry entry here: <http://dev.chromium.org/administrators/policy-list-3#AuthNegotiateDelegateWhitelist>.

## Mozilla Firefox

The Mozilla Firefox browser does not implement the same settings as Microsoft Internet Explorer or Google Chrome. You will need to configure Mozilla Firefox separately. Within Mozilla Firefox, enter the URL “**about:config**” and make the changes. There are hundreds of configuration options, so it is faster to search for the options to change. You can search for “**nego**” to return all the options you need to change. You need to set the following options:

- **network.negotiate-auth.trusted-uris**
- **network.negotiate-auth.delegation-uris**

These settings are comma separated strings. The first option lists sites you want to perform Kerberos authentication to and the second option lists sites you want to perform Kerberos delegation to. If you want to include everything under part of a network domain you just use a dot, so the entry “.**example.com**” will include any site that has example.com in the host name of the server.

## MICROSOFT WINDOWS CHALLENGES

Microsoft has introduced several security features in newer releases of Microsoft Windows that present challenges for the configuration of Kerberos delegation through the SAS 9.4 environment and on to the Secure Hadoop Cluster. In this section, we will detail some of these features and the challenges they present.

### User Security Restrictions

Starting with Microsoft Windows 2003, Microsoft introduced the “**Account is sensitive and cannot be delegated**” flag. When this flag is set, the security context of the user is not delegated to a service even if the service account is set as trusted for Kerberos delegation. Therefore, if your end users have this flag set then they would be able to make the initial authentication connection, but would not be able to make the follow-on authentication step. For example, with authentication from the SAS 9.4 web application, shown in Figure 2, your end users will be able to authenticate to the SAS 9.4 middle tier. But they will not be able to authenticate to the SAS 9.4 metadata tier or SAS 9.4 compute tier. The only resolution is to clear this flag from your end users.

### Access to Ticket-Granting Ticket Session Key

Also starting with Microsoft Windows 2003, Microsoft limited access to the session key associated with the Kerberos Ticket-Granting Ticket. This means that only processes that use the Microsoft Window Security Support Provider Interface (SSPI) can correctly use the Ticket-Granting Ticket. This especially impacts Java processes that rely on the GSS-API for Kerberos implementation.

This restriction on the GSS-API only impacts the connection to the Secure Hadoop Cluster from SAS Workspace Server. This is due to the Java connection to the Secure Hadoop Cluster. The browsers, SAS Management Console, and SAS Enterprise Guide all leverage the SSPI for Kerberos authentication.

To work around this limitation Microsoft has a Windows registry key that can be set to re-enable access to the TGT session key. The key is `AllowTgtSessionKey` in the following path:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters
```

This entry does not exist in the Windows registry by default and needs to be set to true. More details about the Windows registry key can be found here: <https://support.microsoft.com/kb/837361/>.

### User Account Control

Starting with Windows 7 and Windows 2008 R2, Microsoft has extended the limited access to the Ticket-Granting Ticket session key. The Windows registry key (`AllowTgtSessionKey`) detailed above does not apply to your users who are a local administrator with User Access Control (UAC) enabled. This means that you could find that some of your end users are able to access the Secure Hadoop Cluster while other end users cannot. Essentially, there are two simple resolutions for this issue. Either remove

local administrator rights from the user, or turn off UAC on the SAS 9.4 compute tier. More details about this issue can be found here: <http://support.microsoft.com/kb/2627903>.

## Resource-based Constrained Delegation

In Microsoft Windows 2012, Microsoft introduced a new resource-based Kerberos constrained delegation. Resource-based constrained delegation can be used when the front-end services and the resource services are not in the same domain. The traditional model of constrained delegation, that we examined above, is limited to a single domain and requires a domain administrator since you configure the options on the front-end service's account.

In Windows 2012, service administrators can specify which service identities can impersonate users to their services. Configure constrained delegation on the resource permits services such as SAS Workspace Server or the Secure Hadoop services to control which service accounts can delegate Kerberos authentication to them. You can explore further details about the concepts of resource-based constrained delegation here: <https://technet.microsoft.com/library/hh831747.aspx>. You can see examples of configuring resource-based constrained delegation here: <https://blog.kloud.com.au/2013/07/11/kerberos-constrained-delegation>.

You can control these configuration settings from within Microsoft PowerShell on any domain connected host and you do not need to have domain administration privileges. You use either **ADComputer** or **ADUser** to interact with the **PrincipalsAllowedToDelegateToAccount** setting for either computer or user accounts. For example, to list the settings for a service account used for the HIVE service, use the following:

```
Get-ADUser <HIVE Service Account> -Properties  
PrincipalsAllowedToDelegateToAccount
```

An example when there is nothing set is given in **Error! Reference source not found.**

```
DistinguishedName           : CN=<HIVESrv>,OU=SrvcAccnts,  
DC=example,DC=com  
Enabled                     : True  
GivenName                  : <HIVESrv>  
Name                       : <HIVESrv>  
ObjectClass                : user  
ObjectGUID                 : 4fc2d239-8450-4542-bd6d-bdcef9022b69  
PrincipalsAllowedToDelegateToAccount : {}  
SamAccountName             : <HIVESrv>  
SID                        : S-1-5-21-2186632733-3503596390-  
3468275802-21740  
Surname                    :  
UserPrincipalName          : HIVE/hostname@EXAMPLE.COM
```

### Output 1. Output from Get-ADUser PowerShell Cmdlet

An example where a computer account can delegate authentication to HIVE is given in Output 2.

```
DistinguishedName           : CN=<HIVESrv>,OU=SrvcAccnts,  
DC=example,DC=com  
Enabled                     : True  
GivenName                  : <HIVESrv>  
Name                       : <HIVESrv>  
ObjectClass                : user  
ObjectGUID                 : d31b9fb5-8d2c-4e1a-a1c8-3a65c0a59200  
PrincipalsAllowedToDelegateToAccount : {CN=ComputerName,CN=Computers,  
DC=example,DC=com}  
SamAccountName             : <HIVESrv>
```

SID	: S-1-5-21-2186632733-3503596390-
3468275802-18531	
Surname	:
UserPrincipalName	: HIVE/hostname@EXAMPLE.COM

## Output 2. Output from Get-ADUser PowerShell Cmdlet

Resource-based constrained delegation can present challenges for you. First, as with standard constrained delegation, this is not supported for the SAS 9.4 middle tier. If you have attempted to configure resource-based constrained delegation for the connection to SAS 9.4 Object Spawner or SAS 9.4 Metadata Server this will not work. You must revert to unconstrained delegation for the SAS 9.4 middle tier.

Second, if you are attempting to use resource-based constrained delegation for the Secure Hadoop Cluster, you might run into issues with setting the correct principals. Resource-based constrained delegation only supports either user or computer objects defined in the **PrincipalsAllowedToDelegateToAccount** field. You cannot mix users and computers in this field. Therefore, if you need to enable some services running as local system on certain hosts and other services running as a service account, you must revert to standard constrained delegation.

Finally, resource-based constrained delegation requires Windows 2012. If you have Windows 2008R2 servers in the authentication referral path, the process will fail. You can apply the Microsoft Hotfix 2665790 (<http://support.microsoft.com/kb/2665790>) to any Windows 2008 R2 domain controllers. Also, the two servers hosting the front-end services and resources services must be running Windows 2012.

To remove the resource-based constrained delegation setting you just need to set the **PrincipalsAllowedToDelegateToAccount** field to null. For example:

```
Set-ADUser <HIVE Service Account> -PrincipalsAllowedToDelegateToAccount $null
```

## Windows Defender Credential Guard

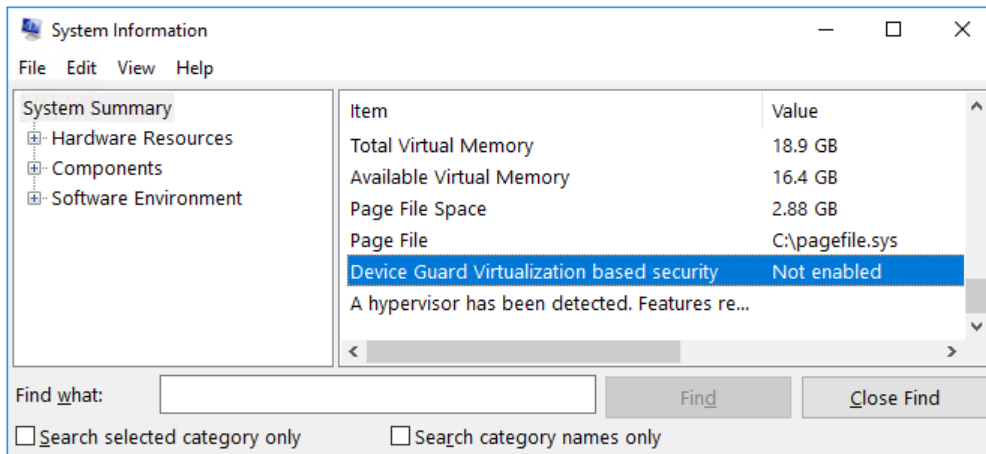
Microsoft has further enhanced security options in Windows 10 and Windows Server 2016. In both operating systems, Microsoft has leveraged virtualization technology to isolate the Local Security Authority (LSA). Data stored in the isolated LSA process is not accessible to the rest of the operating system. LSA uses remote procedure calls to communicate with the isolated LSA process. You can obtain details about how Windows Defender Credential Guard operates here: <https://docs.microsoft.com/en-gb/windows/access-protection/credential-guard/credential-guard-how-it-works>.

You will find that Windows Defender Credential Guard presents challenges since as Microsoft states; applications will break if they require either of the following:

- Kerberos unconstrained delegation
- Extracting the Kerberos TGT

You know from the discussion above this will negatively impact both the SAS 9.4 middle tier and the connection from SAS Workspace Server to the Secure Hadoop Cluster. There is no mitigation in the way that Windows Defender Credential Guard operates. So, you will need to ensure that this technology is not enabled.

You can confirm the status of Windows Defender Credential Guard by launching `MSINFO32.EXE` and viewing the information. The first field to look for is “**Device Guard Virtualisation based security**”. This field shows if the virtualization technology required by Windows Defender Credential Guard is running. Then the field “**Device Guard Security Services Running**” will list if Credential Guard is running on the host. Figure 5 illustrates the situation where the virtualization technology is not running, so Windows Defender Credential Guard cannot be running.



**Figure 5. Checking Windows Defender Credential Guard Status**

## TROUBLESHOOTING TECHNIQUES

In this section of the paper, I will present information about how to enable additional logging, to help with troubleshooting, and then some various methods of testing the environment to provide troubleshooting information.

### ADDITIONAL LOGGING

The first step in effective troubleshooting is for you to obtain as much information as possible. We obtain the information by enabling additional logging from the different parts of the environment. With logging enabled, when you carry out further troubleshooting steps, you will be able to collect the correct information.

### SAS 9.4 Middle-Tier

You can collect a myriad of information from the SAS 9.4 middle tier. The specific information we are interested is the following:

1. Access/content of the Kerberos keytab file,
2. Issues with authenticating to the SAS 9.4 middle tier,
3. Issues with Kerberos delegation.

You can address the first two by enabling debug logging for the Java Kerberos and GSS-API implementations. You need to update the `wrapper.conf` for the SAS Web Application Server instance running SAS 9.4 Logon Manager. This will normally be `SASServer1_1` and the `wrapper.conf` file is located in the `<SASCONFIG>\Lev1\Web\WebAppServer\SASServer1_1\conf` directory. Add two additional arguments as follows:

```
wrapper.java.additional.62=-Dsun.security.krb5.debug=true
wrapper.java.additional.63=-Dsun.security.jgss.debug=true
```

Ensure that the numbering sequence continues from the existing entries. Then, you must enable debug logging output for the process. Also in the `wrapper.conf` file, locate the property `wrapper logfile.loglevel` and set the value to `DEBUG`, so that it looks like the following example:

```
wrapper logfile.loglevel=DEBUG
```

You need to restart the SAS Web Application Server instance for your changes to take effect. When you connect to the SAS 9.4 middle tier, you will obtain debug logging information in the file <SASCONFIG>\Lev1\Web\WebAppServer\SASServer1\_1\logs\wrapper.log. This will include information about reading the contents of the Kerberos keytab and the types of encryption used as well as information about the authentication of the end users. Output 3, provides an example output where the authentication is operating correctly.

```

...| Search Subject for SPNEGO ACCEPT cred (<<DEF>>,
sun.security.jgss.spnego.SpNegoCredElement)
...| Search Subject for Kerberos V5 ACCEPT cred (<<DEF>>,
sun.security.jgss.krb5.Krb5AcceptCredential)
...| Found KeyTab
...| Entered Krb5Context.acceptSecContext with state=STATE_NEW
...| >>> KeyTabInputStream, readName(): EXAMPLE.COM
...| >>> KeyTabInputStream, readName(): HTTP
...| >>> KeyTabInputStream, readName(): <sasmiddletier>.example.com
...| >>> KeyTab: load() entry length: 95; type: 23
...| >>> KeyTabInputStream, readName(): EXAMPLE.COM
...| >>> KeyTabInputStream, readName(): HTTP
...| >>> KeyTabInputStream, readName(): <sasmiddletier>.example.com
...| >>> KeyTab: load() entry length: 95; type: 17
...| >>> KeyTabInputStream, readName(): EXAMPLE.COM
...| >>> KeyTabInputStream, readName(): HTTP
...| >>> KeyTabInputStream, readName(): <sasmiddletier>.example.com
...| >>> KeyTab: load() entry length: 111; type: 18
...| Java config name:
C:\SAS9.4\Config\Lev1\Web\WebAppServer\SASServer1_1\conf\krb5.ini
...| Loaded from Java config
...| Added key: 18version: 3
...| Added key: 17version: 3
...| Added key: 23version: 3
...| Ordering keys wrt default_tkt_etypes list
...| Using builtin default etypes for default_tkt_etypes
...| default etypes for default_tkt_etypes: 18 17 16 23 1 3.
...| >>> EType: sun.security.krb5.internal.crypto.ArcFourHmacEType
...| Using builtin default etypes for permitted_etypes
...| default etypes for permitted_etypes: 18 17 16 23 1 3.
...| >>> EType: sun.security.krb5.internal.crypto.ArcFourHmacEType
...| replay cache for UserName@EXAMPLE.COM is null.
...| object 0: 1515581964001/1082
...| >>> KrbApReq: authenticate succeed.
...| Krb5Context setting peerSeqNumber to: 324395701
...| >>> EType: sun.security.krb5.internal.crypto.ArcFourHmacEType
...| Krb5Context setting mySeqNumber to: 89647732

```

### Output 3. Example wrapper.log Output

You can confirm that Kerberos delegation has taken place by updating the logging configuration for the SAS 9.4 Logon Manager application itself. Edit the SASLogon-log4j.xml file and add the following definition before the <root> definition:

```

<category additivity="false"
name="com.sas.svcs.security.authentication.gss">
  <priority value="DEBUG"/>
  <appender-ref ref="SAS_CONSOLE"/>

```

```
<appender-ref ref="SAS_FILE"/>
</category>
```

The SASLogon-log4j.xml file is in the <SASCONFIG>\Lev1\Web\Common\LogConfig directory. This will provide information about the user name and the Kerberos principal name of your end user that has delegated credentials. Output 4 illustrates a successful connection where delegated credentials are provided for the end user.

```
... INFO
com.sas.svcs.security.authentication.gss.GSSCredentialCachingFilter -
Received credentials for 'UserName'
... DEBUG
com.sas.svcs.security.authentication.gss.GSSCredentialCachingFilter - Using
OMIServerPrincipal 'Principal Name'
... DEBUG
com.sas.svcs.security.authentication.gss.GSSCredentialCachingFilter -
Cached credential for 'UserName' with key 'Principal Name'
... DEBUG
com.sas.svcs.security.authentication.gss.GSSCredentialCachingFilter -
Credentials from 'UserName' have remaining lifetime of 35952 secs
... DEBUG
com.sas.svcs.security.authentication.gss.GSSCredentialCachingFilter - Put
username 'Principal Name' into cache with URL
http://sasmiddletier.example.com:8080/SASLogon/v1/gss/?username=Principal
Name
```

#### Output 4. Example SAS 9.4 Logon Manager Output for Kerberos Delegation

You can find more details about these logging configurations in the paper “Tips and Best Practices for Configuring Integrated Windows Authentication” (Roda 2016).

### SAS 9.4 Workspace Server

You know how to enable standard SAS Workspace Server logging, or you can review the SAS 9.4 Intelligence Platform documentation. Here I propose additional logging that will provide more insight into the Kerberos connection from SAS Workspace Server to the Secure Hadoop Cluster. Just as we discussed above for the SAS 9.4 middle tier, we can use the Java debug options to obtain debug logging for the connection to Secure Hadoop.

Update the `sasv9.cfg` file located in <SASHOME>\SASFoundation\9.4\nls\<language>\ directory to add the Java debug options. Add the following to the `-JREOPTIONS`:

```
-Dsun.security.krb5.debug=true
-Dsun.security.jgss.debug=true
```

After making these changes, you should restart SAS 9.4 Object Spawner.

Enabling this level of debugging for the Java process used by SAS Workspace Server will, as with the SAS 9.4 middle tier, provide much more detail on what is happening. The log file that is produced on the host with SAS Workspace Server will not be in the normal location. The file will be in the `C:\Users\<End-Username>\AppData\Roaming\SAS\LOGS` directory and will be called `sasjava.<random characters>.log`, for example, `sasjava.16a0.42ec4fcd.log`. This log file will be automatically deleted as soon as SAS Workspace Server terminates, so you will need to ensure it is copied somewhere else before it is deleted.

To illustrate the useful information contained in this log, look at the error message shown in Output 5. This is the error shown in SAS Studio when trying to connect to a Secure Hadoop Cluster. The message

“GSS initiate failed” tells us something has gone wrong in establishing the Kerberos connection but does not give much more detail.

```
ERROR: Error trying to establish connection: Could not open client
transport with JDBC Uri:
jdbc:hive2://hivehost.example.com:10000/default;principal=hive/_HOST@HADOOP
.EXAMPLE.COM: GSS initiate failed
ERROR: Error in the LIBNAME statement
```

#### Output 5. Example Error in Hadoop LIBNAME Statement

If you have the additional Java debug options enabled, you can see the actual Kerberos error that has caused the “GSS initiate failed” failure. Output 6 shows part of the `sasjava` log with the real problem. Here we are attempting a cross-realm authentication from the end-user Kerberos realm of `USER.EXAMPLE.COM` to the Kerberos realm for Secure Hadoop of `HADOOP.EXAMPLE.COM`. But the Kerberos implementation on SAS Workspace Server is unable to find the second realm. Therefore, you can see the error message “**Server not found in Kerberos database**”.

```
>>>KRBError:
  sTime is Thu Jan 11 14:40:02 EST 2018 1515699602000
  suSec is 403051
  error code is 7
  error Message is Server not found in Kerberos database
  realm is USER.EXAMPLE.COM
  sname is hive/hivehost.example.com
  msgType is 30
KrbException: Server not found in Kerberos database (7)
```

#### Output 6. SAS 9.4 Workspace Server Java Debug Logging

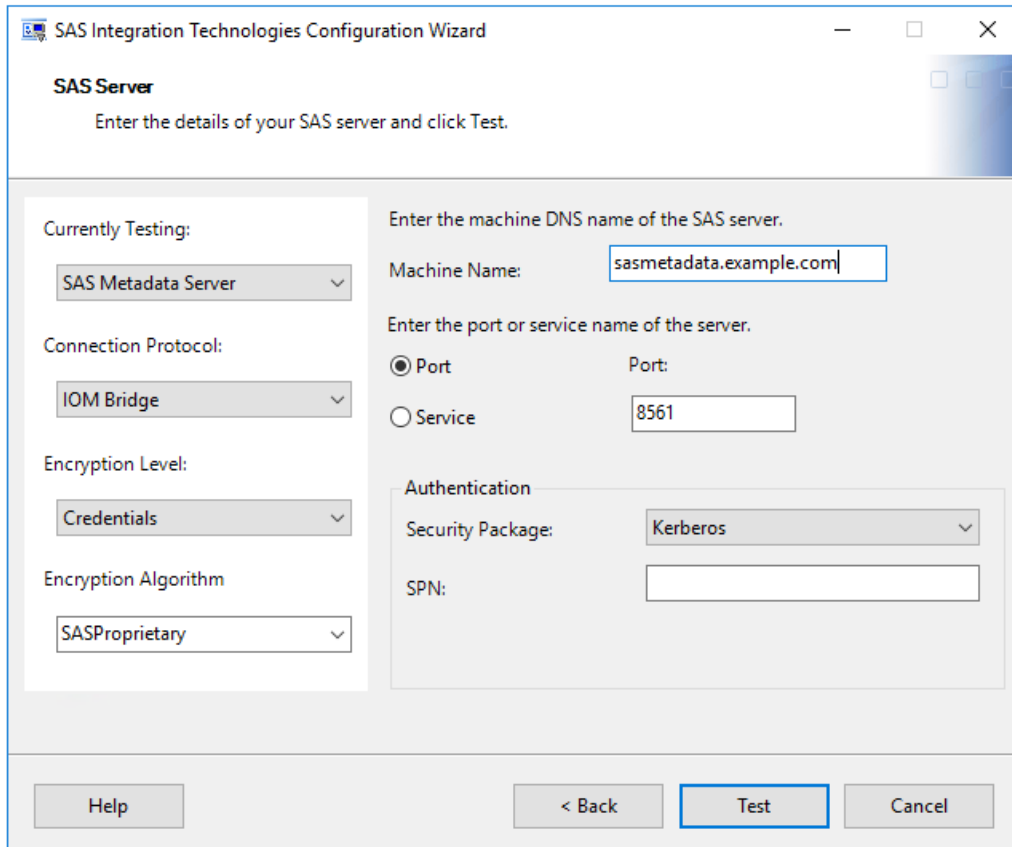
You know, from the discussion above, the resolution to such an error is to provide the Java process launched by SAS Workspace Server with a Kerberos configuration file. So, you update the `sasv9.cfg` and add the option to define a Kerberos configuration file. Where the Kerberos configuration file provides the information about the different Kerberos realms and cross-realm trusts.

### TESTING WITH SAS® INTEGRATION TECHNOLOGIES CONFIGURATION WIZARD

You can start testing with the tools provided by SAS. To test the connection to SAS 9.4 Metadata Server or SAS Workspace Server, you can use the SAS Integration Technologies Configuration Wizard. When you launch the wizard, the first option is to test SAS servers. Once in the wizard, you can manually define the connection to the server to test, the default options will test SAS Workspace Server. However, you can update the options as shown in Figure 6, to test SAS 9.4 Metadata Server as well.

You will not be able to test the connection to the SAS 9.4 middle tier or the Secure Hadoop Cluster using the SAS Integration Technologies Configuration Wizard. However, it does enable you to validate the Kerberos connection to SAS 9.4 Metadata Server and SAS Workspace Server. This quick and simple test will allow you to quickly resolve issues with these authentication steps.





**Figure 6. SAS Integration Technologies Configuration Wizard**

## OTHER TESTING

You can use SAS Studio to test the connection from the SAS 9.4 web applications into the SAS 9.4 environment. Launching SAS Studio will validate steps 1-3 in Figure 2. If you can launch SAS Studio, you will know that the SAS 9.4 middle tier, SAS 9.4 metadata tier, and SAS 9.4 compute tier are all correctly processing the Kerberos delegated authentication. If you have issues with SAS Studio, you can easily see where your problems are occurring at one of the following:

- Issues logging in to SAS Studio mean there are problems with the SAS 9.4 middle tier configuration
- Issues launching SAS Workspace Server mean there are problems delegating the authentication from the SAS 9.4 middle tier

If you experience problems launching SAS Workspace Server, and you are using Google Chrome, don't forget you need the registry entry to enable Kerberos delegation with this browser.

You can then test the connection to the Secure Hadoop Cluster by issuing a simple Hadoop `LIBNAME` statement. With the Hadoop `LIBNAME` statement, if you have the latest configuration files from the Secure Hadoop Cluster, you will not need to specify the `HIVE_PRINCIPAL` or `HDFS_PRINCIPAL` options. As we have shown above the best way of debugging issues with the Hadoop `LIBNAME` statement is to update the `sasv9.cfg` and add the Java debugging options. You will then be able to gain a better understanding of the underlying issue.

Finally, in Table 1 below, I present some common issues and steps you can take to resolve the issues.

Common Issues	Resolution
Failures to delegate from SAS 9.4 middle tier to SAS 9.4 compute tier	<ul style="list-style-type: none"> <li>• Confirm “<b>Trust this user for delegation to any service (Kerberos only)</b>” for the SAS 9.4 middle-tier service account</li> <li>• Confirm browser settings</li> </ul>
Failure to use Kerberos to launch SAS Workspace Server from SAS Enterprise Guide	<ul style="list-style-type: none"> <li>• Confirm SAS Enterprise Guide is running on a different host than SAS Workspace Server</li> <li>• Confirm cross-realm settings, in the SAS Enterprise Guide profile. It might be necessary to enter the full SPN and include REALM.</li> </ul>
Kerberos error “ <b>Failed to find any Kerberos tgt</b> ” connecting to Secure Hadoop Cluster	<ul style="list-style-type: none"> <li>• Confirm the Windows registry setting AllowTgtSessionKey</li> <li>• Confirm membership in Local Administrators and UAC settings</li> <li>• Confirm presence of Windows Defender Credential Guard</li> </ul>
Kerberos error “ <b>Server not found in Kerberos database</b> ” connecting to Secure Hadoop Cluster	<ul style="list-style-type: none"> <li>• Confirm cross-realm settings and add a Kerberos configuration file to SAS Workspace Server</li> </ul>
Kerberos error “ <b>Client not found in Kerberos database</b> ” connecting to Secure Hadoop Cluster	<ul style="list-style-type: none"> <li>• Confirm cross-realm settings and add a Kerberos configuration file to SAS Workspace Server</li> </ul>

**Table 1. Common Issues and Resolutions**

## CONCLUSION

Kerberos authentication to Hadoop is becoming a common requirement for many organizations. Materials so far have focused heavily on Linux and other UNIX operating systems, with the expectation that Kerberos in Microsoft Windows just works. In this paper, we have focused solely on Microsoft Windows. We examined how to enable Kerberos authentication, looking first at the different authentication steps, and then discussing at a high level the steps you’ll need to complete.

Once we understood the overall process, we looked at some of the specific configuration changes you’ll need to complete to get your environment operating correctly. We examined some specific challenges you’ll face when working with a Microsoft Windows environment. This included some of the security technology introduced by Microsoft that can limit your ability to get the environment working as expected.

Finally, we reviewed some troubleshooting techniques. We looked at some different types of logging you can enable, to gain a better understanding of what is happening in your environment. You saw some different ways of testing your environment and we finished with some common issues and the steps you can take to resolve them.

## REFERENCES

Rogers, Stuart J. 2013. “Kerberos and SAS® 9.4: A Three-Headed Solution for Authentication.” *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings13/476-2013.pdf>.

Rogers, Stuart J., and Keefer, Tom. 2014. "Hadoop with Kerberos: Architecture Considerations." SAS Technical Papers. Cary, NC: SAS Institute Inc. Available [https://support.sas.com/resources/papers/Hadoop\\_Architecture.pdf](https://support.sas.com/resources/papers/Hadoop_Architecture.pdf).

Rogers, Stuart J., and Keefer, Tom. 2014. "Hadoop with Kerberos: Deployment Considerations." SAS Technical Papers. Cary, NC: SAS Institute Inc. Available [https://support.sas.com/resources/papers/Hadoop\\_Deployment.pdf](https://support.sas.com/resources/papers/Hadoop_Deployment.pdf).

Li, Zhiyong, and Roda, Mike. 2014. "An Advanced Fallback Authentication Framework for SAS® 9.4 and SAS® Visual Analytics". *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings14/SAS102-2014.pdf>.

Rogers, Stuart J. 2016. "Kerberos Delegation with SAS® 9.4." *Proceedings of the SAS Global Forum 2016 Conference*. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings16/SAS3443-2016.pdf>.

Roda, Mike. 2016. "Tips and Best Practices for Configuring Integrated Windows Authentication." *Proceedings of the SAS Global Forum 2016 Conference*. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings16/SAS3720-2016.pdf>.

Rogers, Stuart J. 2017. "Kerberos Cross-Realm Authentication: Unraveling the Mysteries." *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings17/SAS0623-2017.pdf>.

## ACKNOWLEDGMENTS

I would like to thank the following people for taking the time to review and contribute to this paper:

- Chuck Hunley
- Mike Roda
- Donna Bennett

## RECOMMENDED READING

- SAS® 9.4 Intelligence Platform: Installation and Configuration Guide
- SAS® 9.4 Intelligence Platform: Security Administration Guide
- SAS® 9.4 Intelligence Platform: Middle-Tier Administration Guide

## CONTACT INFORMATION

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

Stuart J Rogers  
SAS Institute Inc.  
[stuart.rogers@sas.com](mailto:stuart.rogers@sas.com)  
<http://www.sas.com>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.