Paper 476-2013

Kerberos and SAS® 9.4: A Three-Headed Solution for Authentication

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

ABSTRACT

Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography. With the release of SAS® 9.4, there are three ways Kerberos can be used with the SAS® Business Analytics Framework. Kerberos provides Integrated Windows authentication from a range of clients to a range of servers. This paper reviews how Kerberos is used with the SAS Business Analytics Framework. It explores the considerations and constraints when using Kerberos and summarizes solutions for some common issues.

INTRODUCTION

Kerberos is an industry standard authentication protocol and is implemented and embedded into many modern operating systems and other software applications. Within this paper we review the Kerberos authentication protocol and provide an understanding of the steps in the authentication process. We discuss the key components used by the Kerberos authentication protocol and show how it has been embedded into the Microsoft Active Directory domain structure.

Once we provide a better understanding of the Kerberos authentication protocol, we examine how the SAS Business Analytics Framework can leverage Kerberos authentication. We look at the different parts of the SAS Business Analytics Framework and see how they fit into the Kerberos authentication protocol. This continues into reviewing how the support of Kerberos authentication has changed over the SAS releases from SAS® 9.2 to SAS 9.4. When examining the SAS Business Analytic Framework, we consider the server tier, middle tier, and client tier. Of specific interest within this section of the paper are those SAS client applications that do not support the use of the Kerberos authentication protocol.

In addition, we review some of the constraints for using Kerberos authentication. We illustrate how the use of Kerberos is tied into the domain trust structure and show some of the prerequisites of using Kerberos. Of these prerequisites, we identify the clear importance of the Service Principal Name and how this is central to the Kerberos authentication protocol.

Finally, we review some of the common issues that arise when using Kerberos and identify solutions to those error messages. Although troubleshooting Kerberos authentication issues is beyond the scope of this paper, it should be made easier by the understanding of the authentication process we present. Understanding the steps in the authentication process is the first and biggest step in being able to troubleshoot issues.

DEFINITION AND BACKGROUND FOR KERBEROS

A network is an insecure collection of devices. Many protocols used in a network do not provide any security, and tools are available to "sniff" passwords sent across networks. This means that applications that send unencrypted passwords over the network are vulnerable to such interception of passwords. Kerberos was created by the Massachusetts Institute of Technology as a solution to common network security problems. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server and vice versa across an insecure network. After a client and server have used Kerberos to provide their identity, they can also encrypt all of their communications to assure privacy and data integrity. The current version of the Kerberos protocol is version 5.

The Microsoft Windows Server operating systems implement the Kerberos Version 5 authentication protocol and extensions for public key authentication, transport of authorization data, and delegation. The Kerberos authentication client is implemented as a Security Support Provider (SSP), and it can be accessed through the Security Support Provider Interface (SSPI). Initial user authentication is integrated with the Winlogon single sign-on architecture.

The Kerberos Key Distribution Center (KDC) is integrated with other Windows Server security services that run on the domain controller. The KDC uses the domain's Active Directory Domain Services database as its security account database. Active Directory Domain Services is required for default Kerberos implementations within the domain or forest.

KEY TERMS AND DEFINITIONS FOR KERBEROS

The Kerberos protocol name is based on the three-headed dog figure from Greek mythology known as Kerberos. The three heads of Kerberos comprise the Key Distribution Center (KDC), the client, and the server. In the same way that Kerberos in Greek mythology guarded the entrance to the underworld, Kerberos here guards the access to network resources. The KDC is the trusted third party used to verify the authenticity of both the client and the server. Most often a client is an end user, and the server is either a computer or a service running on a computer.

The KDC is a single process that provides two services. The KDC provides the Authentication Service (AS) that issues ticket-granting tickets (TGT) for connection to the Ticket-Granting Service (TGS). Before a client can ask for a ticket to another server, it must request a TGT from the AS. The AS verifies the identity of the client using secret key authentication if the key is symmetric (a single key is used for both encryption and decryption). The client proves its knowledge of the key by encrypting a message, and the AS proves its knowledge of the key by decrypting the message.

The TGS is the second service provided by the KDC. When clients want to access a server, they contact the TGS, present their TGT, and ask for a ticket to the server. The KDC issues a service ticket when presented with a valid TGT. The KDC issues the service ticket without directly communicating with the target server. The KDC encrypts the service ticket with a secret key known only to the KDC and the target server. If the server is able to decrypt the message using the secret key it shares with the KDC when the client presents the service ticket to the server, the server is able to verify that the service ticket is valid.

Already within just two paragraphs we have introduced quite a number of terms very specific to the Kerberos protocol. To help you understand the different keys and tickets, and where these are generated and stored, we can represent them graphically. Figure 1 illustrates the three heads of Kerberos with the Workstation, Server, and Domain Controller. The workstation contains our client's credential cache, the server contains the service credentials cache, and the domain controller contains the KDC account database.

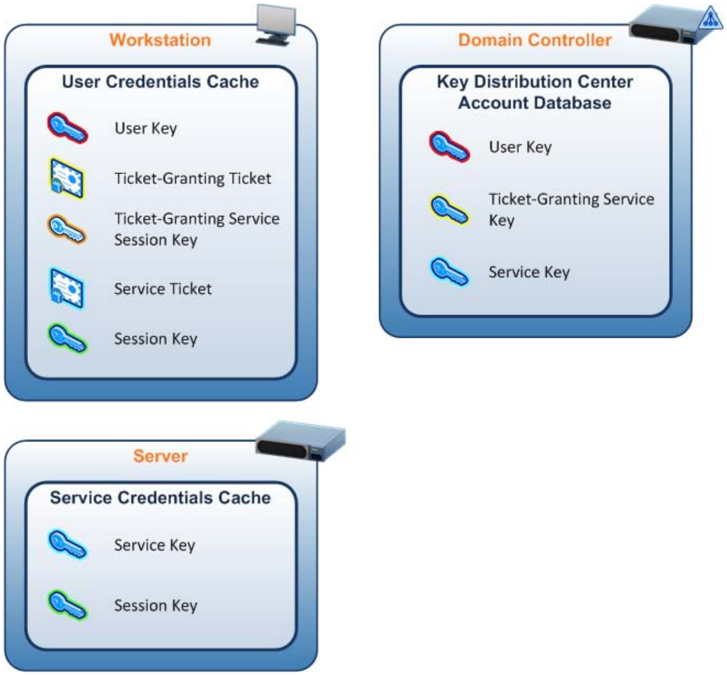


Figure 1: Kerberos Components

User Key: When a user is created, the password is used to create the user key. In Active Directory domains, the user key is stored with the user's object in Active Directory. At the workstation, the user key is created when the user logs on.

Ticket-Granting Service Key: All KDCs in the same realm use the same service key. This is based on the password assigned to the krbtgt account. Every Active Directory domain has the krbtgt built-in account.

Ticket-Granting Service Session Key: The TGS Session Key is generated randomly by the KDC and used only as long as the TGT is valid. The TGS Session Key is used to encrypt authentication messages sent to the TGS by the client.

Service Key: Services use a key based on the account password they use to log on.

Session Key: The session key is generated randomly by the KDC. The session key is used for tickets and is short-lived and only used as long as that session or service ticket is valid. The session key is used to encrypt authentication messages sent to the service by the client.

Ticket-Granting Ticket: A ticket-granting ticket is used to authenticate with the TGS. TGTs are encrypted with a key used by the KDCs. The client cannot read the TGT. The TGT is used to avoid the performance penalties of looking up a user's long-term key every time the user requests a service.

Service Ticket: A service ticket is used to authenticate with services other than the TGS and is meant only for the target service. A service ticket is encrypted with a service key

Service Principal Name: A Service Principal Name (SPN) is a unique identifier for a service running on a server. Every service that will use Kerberos authentication needs to have an SPN set for it so that clients can identify the service on the network. The format of the SPN is *service*/*hostname*:*port*, where *service* is the service class of the SPN, *hostname* is the server to which the SPN belongs, and *port* is the port on which the service is registered to run.

User Principal Name: A User Principal Name (UPN) is the name of a user in the format of *username@domain*.

PROCESS FLOW OF KERBEROS AUTHENTICATION

With the different components of Kerberos authentication defined, it is now possible to present the process flow of authentication with Kerberos. There are two main steps to authentication with Kerberos. The first is the user's initial login when a number of the key Kerberos components are set up. The second step is to actually authenticate the user to a specified service.

As part of the user login process on a Microsoft Windows domain, the client sends an Authentication Service Request (KRB_AS_REQ), as shown in Figure 2. This message includes the user principal name, the name of the account domain, and pre-authentication data. The pre-authentication data is the current timestamp, which is encrypted with the user's key derived from the user's password. The KDC obtains its copy of the user key from the user's record in Active Directory. The KDC decrypts the pre-authentication data and evaluates the timestamp inside. If the timestamp matches the KDC's current time to within a margin, the KDC can be assured that the pre-authentication data was encrypted with the user's key and can thus verify that the user is genuine.

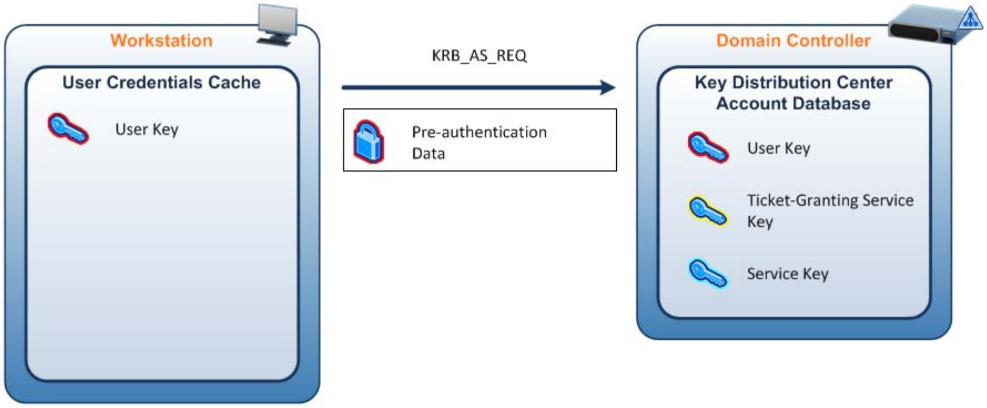


Figure 2: Kerberos Authentication Service Request

The KDC replies with an Authentication Service Reply (KRB_AS_REP), as shown in Figure 3. This message includes the TGT and a copy of the session key the user can use in communicating with the KDC. The TGT is encrypted using the KDC's long-term key, and the user's copy of the session key is encrypted using the user's long-term key. This means that the user cannot decrypt the TGT; this can be decrypted only by the KDC.

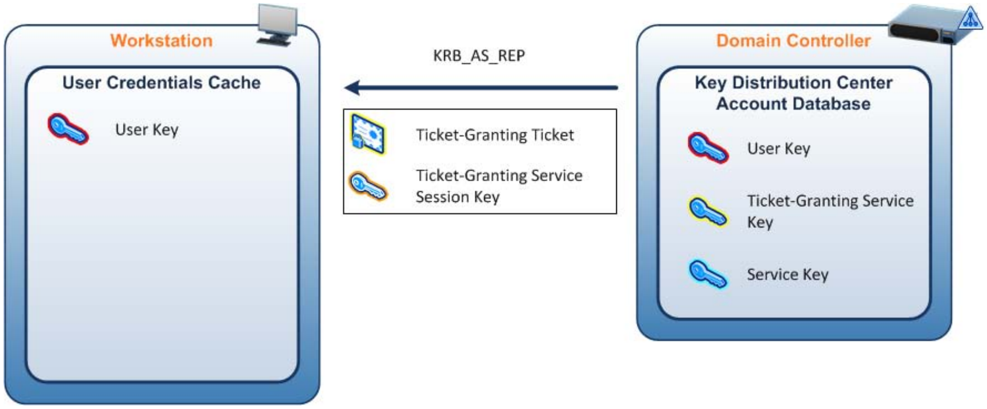


Figure 3: Kerberos Authentication Service Reply

This completes the initial Kerberos exchange. Once the user has a valid TGT, requests can be made to the KDC for additional service tickets. For example, the rest of the Microsoft Windows login process requests service tickets for the computer that the user is logging in to and other network resources such as mapped drives or connections to Microsoft Exchange.

We now examine this second process in which a user authenticates to gain access to a specified service. The Kerberos client requests credentials for the service by sending the KDC a Kerberos Ticket-Granting Service Request (KRB_TGS_REQ), as shown in Figure 4. This message includes the user's name, an authenticator encrypted with the user's key, the TGT, and the name of the service for which the user wants a ticket. The KRB_TGS_REQ message will typically be sent to the same domain controller that issued the TGT. However, TGS requests can be made to any domain controller. Therefore, if the original KDC becomes unavailable, the client can discover a new KDC through a DNS query and then send a KRB_TGS_REQ there.

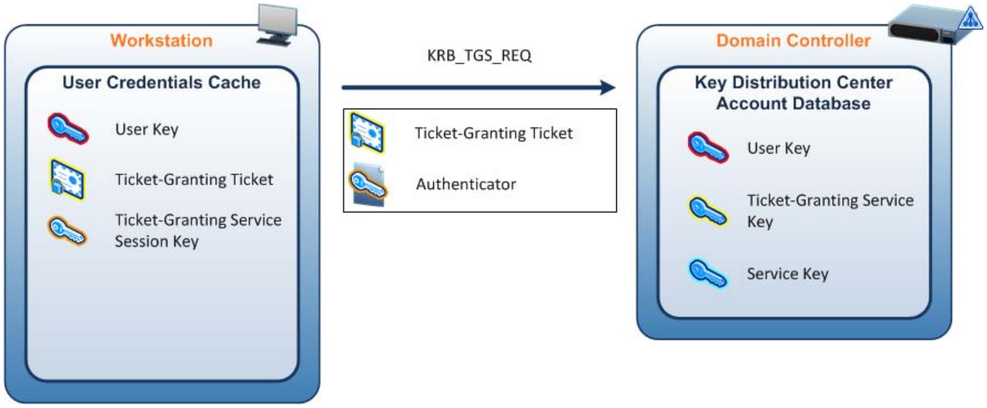


Figure 4: Kerberos Ticket-Granting Service Request

When the KDC receives the KRB_TGS_REQ, it decrypts the TGT with its own secret key, extracting the user's TGS session key. It uses the session key to decrypt the authenticator and evaluates that. If the authenticator passes the test, the KDC extracts the user's authorization data from the TGT and creates another session key for the client to use with the service. The KDC encrypts one copy of this new session key with the user's TGS session key. It embeds another copy of the session key in a ticket, along with the user's authorization data, and encrypts the ticket with the service's key. The KDC then sends these credentials back to the client in a Kerberos Ticket-Granting Service Reply (KRB_TGS_REP), as shown in Figure 5.

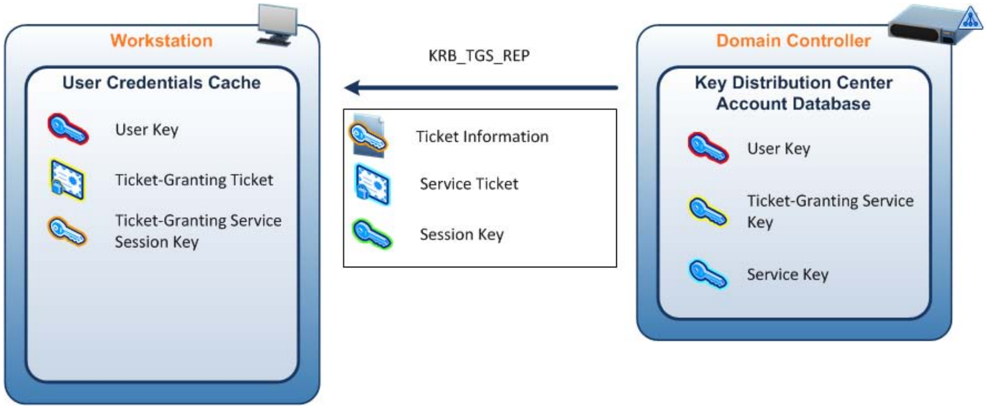


Figure 5: Kerberos Ticket-Granting Service Reply

When the Kerberos client receives the reply, it uses the user's TGS session key to decrypt the session key to use with the service, and stores the key in its credentials cache. Then it extracts the ticket to the service and stores that in its cache. The Kerberos client then requests access to the service by sending the service a Kerberos Application Request (KRB_AP_REQ), as shown in Figure 6. This message contains the following four pieces of information:

- an application option flag indicating whether to use the session key
- an application option flag indicating whether the client wants mutual authentication
- the service ticket obtained in the TGS exchange
- an authenticator encrypted with the session key for the service

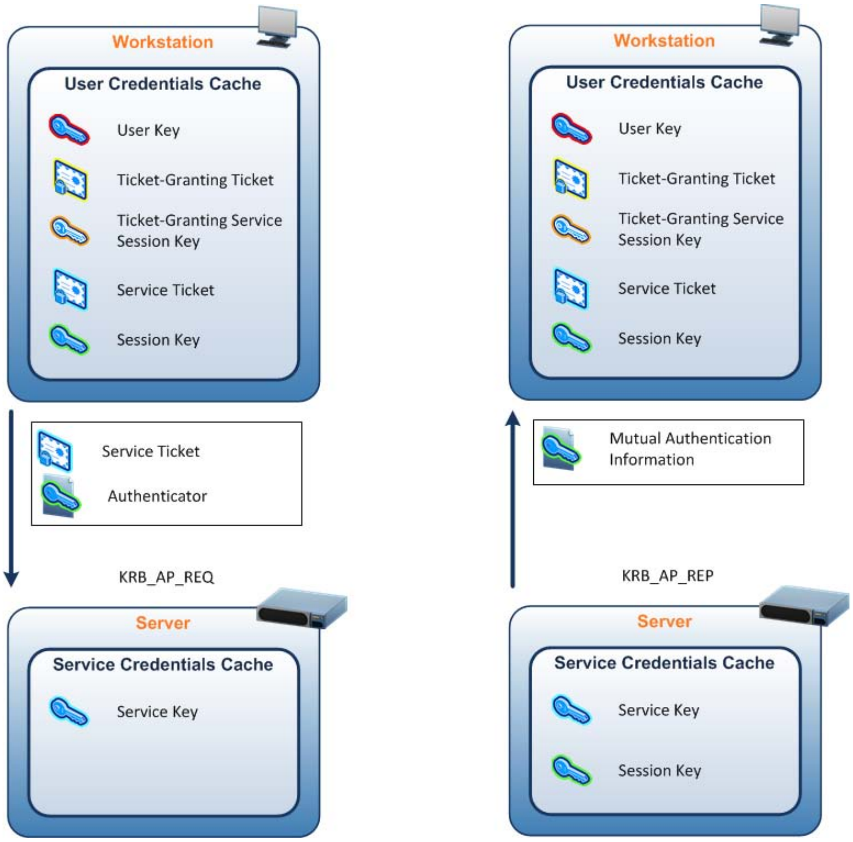


Figure 6: Kerberos Application Request and Reply

The service receives the KRB_AP_REQ, decrypts the ticket, and extracts the user's authorization data and the session key. The service uses the session key to decrypt the user's authenticator and then evaluates the timestamp inside. If the authenticator passes the test, the service looks for a mutual authentication flag in the client's request. If the flag is set, the service uses the session key to encrypt the time from the user's authenticator and returns the results in a Kerberos Application Reply (KRB_AP_REP), as shown in Figure 6. If the flag is not set, then no response is needed.

When the client receives the KRB_AP_REP, it decrypts the service's authenticator with the session key it shares with the service and compares the time returned by the service with the time in the client's original authenticator. If the times match, the client knows the service is genuine.

This completes the review of Kerberos authentication. The next section of the paper moves on to consider how SAS authentication can leverage Kerberos for secure authentication across the network.

SAS AUTHENTICATION WITH KERBEROS

From the review of Kerberos in the preceding section, you now understand the three heads of Kerberos authentication. For SAS authentication with Kerberos, we have the SAS server process, the KDC, and the SAS client application. We will break down the SAS server process into three different types. The first type consists of those SAS server processes running on a Microsoft Windows operating system, the next are SAS server processes running on UNIX operating systems, and finally the SAS server processes running in the middle tier. We split the type of SAS server process because there are different levels of support, configuration options, and considerations, all based on the type of server process.

For each of the types of SAS server process, we consider how each one interoperates with the Kerberos authentication protocol. For the SAS servers discussed in the first two types, we include the SAS® Metadata Server, SAS® OLAP Server, and SAS Integrated Object Model (IOM) servers processing data. These IOM servers include the SAS Workspace Server, SAS® Stored Process Server, and the SAS Pooled Workspace Server, all launched via the SAS object spawner.

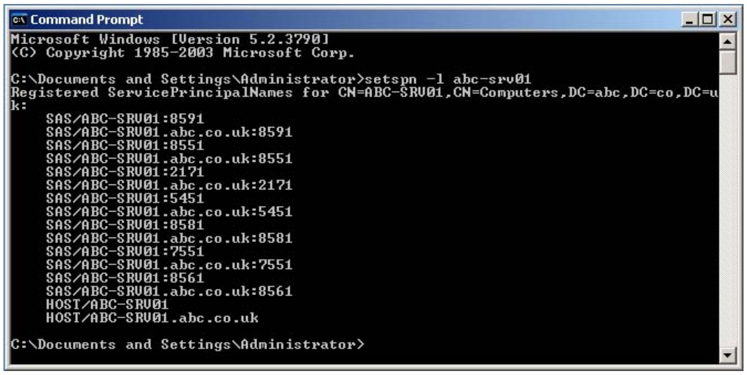
WINDOWS SERVERS

Since the release of SAS® 9.2, the SAS server processes running on hosts with the Microsoft Windows operating system have been able to use Kerberos for authentication. For the KDC to act as the trusted third party, Kerberos must be made aware of the SAS server process. Kerberos is made aware of the SAS server process by registering the Service Principal Name (SPN) for the SAS server process. For the SAS server process running on Microsoft Windows, the process of registering the SPN is straightforward. The computer account within Active Directory has the ability to register SPNs against itself. Therefore, when the SAS server processes start, so long as they are running as the local system account, they are able to register their own SPNs.

The format of the SPN automatically registered by the SAS server processes has changed through the SAS releases. With SAS 9.2, the format is SAS/*hostname*: *port* and SAS/*fully.qualified.hostname*:*port*. For example, for a SAS Metadata Server process, we would have SAS/sasmeta:8561 and SAS/sasmeta.mydomain.com:8561. Each of the SAS server processes registers both the short name and fully qualified domain name for the host the process is running on. Because the SPN includes the network port the process is listening on, there will be multiple SPNs registered for a single host. For example, if we have a single host running all the SAS server processes, we would have all the following SPNs registered:

- Metadata Server – SAS/*hostname*:8561, SAS/*fully.qualified.hostname*:8561
- Object Spawner – SAS/*hostname*:8581, SAS/*fully.qualified.hostname*:8581
- Workspace Server – SAS/*hostname*:8591, SAS/*fully.qualified.hostname*:8591
- SHARE Server – SAS/*hostname*:8551, SAS/*fully.qualified.hostname*:8551
- Connect Spawner – SAS/*hostname*:7551, SAS/*fully.qualified.hostname*:7551
- OLAP Server – SAS/*hostname*:5451, SAS/*fully.qualified.hostname*:5451
- Table Server – SAS/*hostname*:2171, SAS/*fully.qualified.hostname*:2171

This is illustrated in Figure 7.



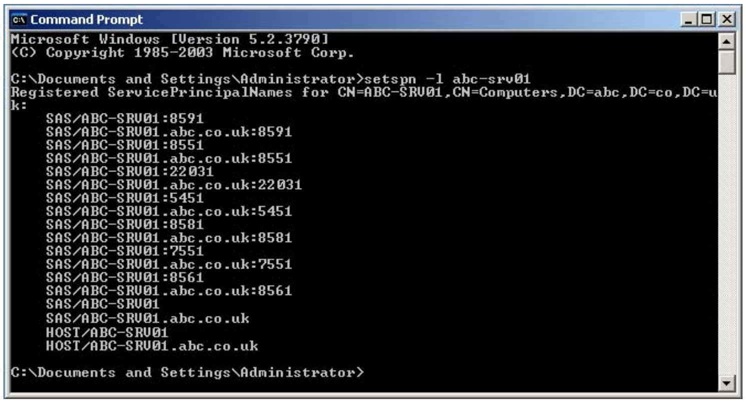
```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>setspn -l abc-srv01
Registered ServicePrincipalNames for CN=ABC-SRV01,CN=Computers,DC=abc,DC=co,DC=u
k:
        SAS/ABC-SRV01:8591
        SAS/ABC-SRV01.abc.co.uk:8591
        SAS/ABC-SRV01:8551
        SAS/ABC-SRV01.abc.co.uk:8551
        SAS/ABC-SRV01:2171
        SAS/ABC-SRV01.abc.co.uk:2171
        SAS/ABC-SRV01:5451
        SAS/ABC-SRV01.abc.co.uk:5451
        SAS/ABC-SRV01:8581
        SAS/ABC-SRV01.abc.co.uk:8581
        SAS/ABC-SRV01:7551
        SAS/ABC-SRV01.abc.co.uk:7551
        SAS/ABC-SRV01:8561
        SAS/ABC-SRV01.abc.co.uk:8561
        HOST/ABC-SRV01
        HOST/ABC-SRV01.abc.co.uk

C:\Documents and Settings\Administrator>
```

Figure 7: SAS 9.2 Service Principal Names

The release of SAS® 9.3 added another two SPNs to those automatically registered. With SAS 9.3, the SPNs SAS/*hostname* and SAS/*fully.qualified.hostname* are registered. By registering just the service type against the host name and removing the port number, we have a more generic service registered. With this type of SPN, the client needs to request only a single Service Ticket to authenticate to any SAS server process running on the host. The list of SPNs for a single machine SAS 9.3 server is shown in Figure 8. Finally, with the release of SAS® 9.4, none of the SPNs with port numbers are registered. Therefore, with SAS 9.4, a single machine with all the SAS server processes has only two SAS SPNs registered, as shown in Figure 9.

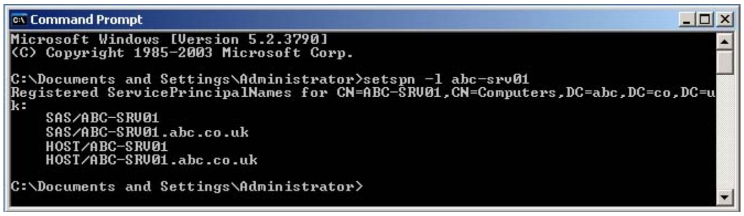


```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>setspn -l abc-srv01
Registered ServicePrincipalNames for CN=ABC-SRV01,CN=Computers,DC=abc,DC=co,DC=u
k:
        SAS/ABC-SRV01:8591
        SAS/ABC-SRV01.abc.co.uk:8591
        SAS/ABC-SRV01:8551
        SAS/ABC-SRV01.abc.co.uk:8551
        SAS/ABC-SRV01:22031
        SAS/ABC-SRV01.abc.co.uk:22031
        SAS/ABC-SRV01:5451
        SAS/ABC-SRV01.abc.co.uk:5451
        SAS/ABC-SRV01:8581
        SAS/ABC-SRV01.abc.co.uk:8581
        SAS/ABC-SRV01:7551
        SAS/ABC-SRV01.abc.co.uk:7551
        SAS/ABC-SRV01:8561
        SAS/ABC-SRV01.abc.co.uk:8561
        SAS/ABC-SRV01
        SAS/ABC-SRV01.abc.co.uk
        HOST/ABC-SRV01
        HOST/ABC-SRV01.abc.co.uk

C:\Documents and Settings\Administrator>
```

Figure 8: SAS 9.3 Service Principal Names



```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>setspn -l abc-srv01
Registered ServicePrincipalNames for CN=ABC-SRV01,CN=Computers,DC=abc,DC=co,DC=u
k:
        SAS/ABC-SRV01
        SAS/ABC-SRV01.abc.co.uk
        HOST/ABC-SRV01
        HOST/ABC-SRV01.abc.co.uk

C:\Documents and Settings\Administrator>
```

Figure 9: SAS 9.4 Service Principal Names

Each of the client applications like SAS® Enterprise Guide or SAS® Management Console know how to take the information provided in the connection profile and build the correct Service Principal Name. When you set up the profile to connect to ABC-SRV01, the client builds the SPN SAS/ABC-SRV01, and this is the SPN the client requests the Service Ticket for. We discuss the SAS client applications in more detail in a later section of this paper.

With the required Service Principal Name registered, Kerberos is aware of the SAS server processes. In addition to registering the SPNs, the SAS server processes also need to be able to process the Service Ticket, as previously outlined. Because the SAS server processes are running on Microsoft Windows operating systems, SAS is able to leverage the standard features built into Microsoft Windows to process the Service Ticket and authenticate the user.

The configuration of the SAS server processes to use Kerberos authentication on Microsoft Windows operating systems is simple. This configuration can be automatically completed by the SAS® Deployment Wizard when creating the environment. The SAS Deployment Wizard presents a single check box that, when selected, ensures that all components are configured for Kerberos authentication. The SAS server processes then offer Kerberos as one of the possible methods of authentication. This does not prevent users from still using user names and passwords.

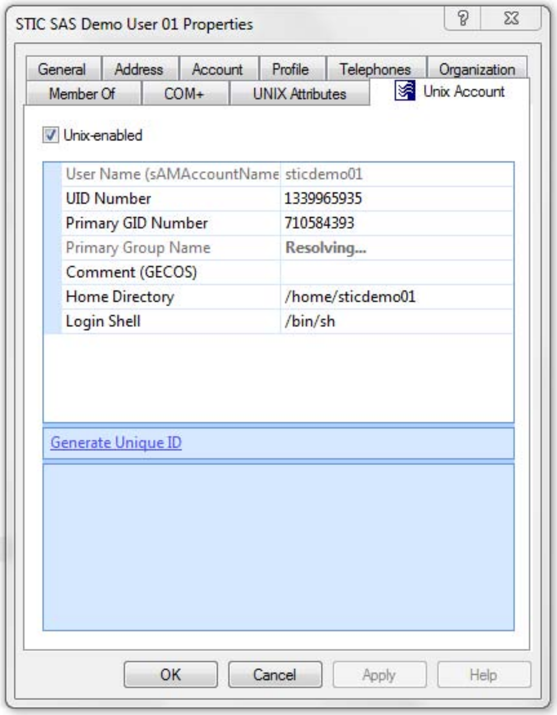
Alternatively, the SAS server processes can be configured manually to use Kerberos authentication after the initial deployment using the SAS Deployment Wizard. Server participation in Kerberos is affected by invocation commands; the start-up script for the SAS server process requires the **-sspi** option. This affects the SAS Metadata Server, object spawner, and SAS OLAP Server. For metadata-aware server processes, such as the SAS Workspace Server, there are settings in the metadata definition for the server. Additional details are available in the *SAS Intelligence Platform: Security Administration Guide* for the given release of SAS software.

UNIX SERVERS

We have examined what happens when the SAS server processes are running on Microsoft Windows operating systems, but what about UNIX operating systems? SAS 9.3 introduced support for Kerberos authentication with UNIX servers. Again, as with Microsoft Windows, the SAS server processes for UNIX operating systems must know how to interact with Kerberos. Because each UNIX distribution implements Kerberos in slightly different ways, it was decided to rely on a third-party package to provide the link between SAS and Kerberos. Quest Authentication Services was selected as the third-party package to provide this integration.

Quest Authentication Services is a software bundle licensed separately from SAS. It is licensed directly from Dell. Purchasing Quest Authentication Services provides a separate support agreement with Quest for assistance with authentication issues. SAS has a specific minimum requirement on Quest Authentication Services; SAS requires at least version 4.0.1.23. Quest Authentication Services is supported on Red Hat Linux, SuSE Enterprise Server, Solaris (SPARC and x64), HP-UX, and IBM AIX. Quest Authentication Services has two components: the first extends the Active Directory schema and provides management tools, and the second component is an agent that runs on the UNIX host. The agent then allows for the integration into Active Directory and communicates with Kerberos.

The extension of the Active Directory schema provided by the Quest tools enables the storing of UNIX user attributes in Active Directory, as shown in Figure 10. This provides a central point to manage the UID, GID, home directory, and login shell for end-user accounts. The end-user accounts are not considered UNIX enabled until these properties have been defined. Care must be taken to ensure that the UID is unique.



STIC SAS Demo User 01 Properties

☑ Unix-enabled

User Name (sAMAccountName	sticdemo01
UID Number	1339965935
Primary GID Number	710584393
Primary Group Name	Resolving...
Comment (GECOS)	
Home Directory	/home/sticdemo01
Login Shell	/bin/sh

Figure 10: Quest Active Directory Schema Extensions

For the SAS server processes, after Quest Authentication Services are set up and configured, the steps for Kerberos authentication are straightforward. The Quest tools enable the creation of a service account in Active Directory and also register Service Principal Names in the format SAS/*hostname* and SAS/*fully.qualifed.hostname*. The service account represents all the SAS processes running on the UNIX host. With this in place, the SAS servers on UNIX use the Quest libraries to make their connections to Kerberos. Additional information about completing the required steps is given in the *Configuration Guide for SAS® Foundation for UNIX Environments*.

The configuration of the SAS server processes running on UNIX operating systems requires some additional steps that are not required on Microsoft Windows operating systems. In the SAS Deployment Wizard, both the options to configure Pluggable Authentication Modules (PAM) and Integrated Windows Authentication should be selected. (PAM enables the SAS server processes to leverage the full options provided by the Quest tools. This will enable the use of the extended Active Directory schema values in creating valid user session on the UNIX host.

Both of the options within the SAS Deployment Wizard require additional manual steps. For the PAM configuration, a sasauth configuration file must be created on the operating system. The Quest tools auto-configure SSH for PAM, and these settings can be used as a basis for the sasauth configuration file. In addition, for the Kerberos configuration, the SAS server processes require the location of a keytab file for the service account. This is provided in an environment variable (KRB5_KTNAME) added to the <config_dir>/Lev1/level_env.sh script file. With these two steps complete, the SAS server processes will be able to use Kerberos for authentication.

If the configuration options are not selected when running the SAS Deployment Wizard, the same steps can be completed manually after deployment. The *SAS Intelligence Platform: Security Administration Guide* for the given release of SAS software includes instructions for completing the manual steps.

MIDDLE TIER

In the preceding two sections, we have considered the cases in which the server process within the Kerberos protocol is a SAS server process. For the case of the middle tier, the server process forming one of the three heads of Kerberos is the Java Application Server. That is, for the scope of authentication from a browser to the SAS middle tier, the server process running the SAS Logon Manager is our point of concern. To enable Kerberos authentication from a browser to the SAS middle tier, we configure the SAS Logon Manager to pass the authentication processing to the application server hosting the SAS Logon Manager.

SAS 9.2 and SAS 9.3

With SAS 9.2 and SAS 9.3, there are three supported application servers: JBoss, WebSphere, and WebLogic. The overall setup for these application servers is very similar to the configuration for UNIX servers. We require a service account in Active Directory to represent the application server. The Service Principal Names (in the middle-tier case these are HTTP(S)/*hostname* and HTTP(S)/*fully.qualified.hostname*) are registered against the service account. A keytab file containing the credentials for the service account is made available to the application server so that it can connect to the KDC. In addition, a Kerberos configuration file is created that provides the default encryption types and domain to Kerberos Realm mappings. The application server is then configured for Kerberos authentication, which is often referred to as Integrated Windows Authentication or Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO).

The methods of configuring the application server for SPNEGO are different for each of the application servers. SAS provides specific documentation for both SAS 9.2 and SAS 9.3 for each application server. With JBoss, SPNEGO is configured as a Java Authentication and Authorization Service (JAAS) module. For the 4.2.x versions of JBoss, this requires an additional third-party module. For JBoss 5.x, the SPNEGO module is part of the distribution. WebLogic implements SPNEGO in a similar fashion to JBoss as a JAAS module. WebLogic includes the Identity Assertion module as part of the standard deployment. WebSphere implements SPNEGO differently from, the other application servers and implements it as a security filter.

The SAS 9.2 and SAS 9.3 configuration of the SAS Logon Manager and SAS Remote Services are largely independent of the application server used. The SAS Logon Manager is updated to include a security filter within the web application. The security filter directs the authentication to the application server. SAS Remote Services is updated to include application server-specific JAR files in the Java class path for the application. Also, the JAAS configuration for SAS Remote Services is updated to use a different SAS authentication domain. Additional details of the configuration and troubleshooting for SPNEGO configurations can be found in the SAS Global Forum 2011 paper “Single Sign-On Configuration and Troubleshooting for SAS® 9.2 Enterprise BI Web Applications.”

SAS 9.4

Beginning with the release of SAS 9.4, SAS will include an embedded middle-tier server called SAS Web Application Server and will no longer require nor support external third-party application servers. SAS has licensed and included a commercial software suite of middle-tier technologies beginning with SAS 9.4. The SAS 9.4 middle-tier components have been integrated and optimized for SAS middle-tier deployment, configuration, and execution. These components are being branded and delivered in the SAS offerings and referred to as the SAS Web Application Server, SAS Web Server, and SAS® Environment Manager.

The SAS Web Application Server is an enterprise-class web application server optimized to deliver the robust level of capabilities required to optimally support the SAS 9.4 web applications deployed through traditional on-premises physical hardware, virtualized environments, or in the cloud. By focusing on a single embedded middle tier, SAS has been able to perform extensive integration to deliver a more highly available, resilient, and manageable deployment. Additionally, SAS provides first- and second-line support for the complete set of integrated components.

For the configuration of Kerberos authentication with the SAS 9.4 middle tier, it is the SAS Web Application Server that must be configured. As with previous releases of SAS using other application servers, for Kerberos authentication with the SAS Web Application Server, we require a service account within Active Directory, registered Service Principal Names, and a Kerberos keytab file representing the credentials of the service account.

The container within the SAS Web Application Server providing the serving of the web applications is Tomcat. The release of Tomcat used by the SAS Web Application Server includes a built-in SPNEGO Valve for providing Kerberos authentication. As such, to configure the SAS Web Application Server for SPNEGO, two files are updated. The first is server.xml; within this file, a Tomcat Realm is defined to support the container-managed security and defines the process for obtaining user roles. The second file is the jaas.config; this file contains the options specific to Kerberos for the Tomcat SPNGEO Valve, including the location of the Kerberos Keytab file. As with SAS 9.2 and SAS 9.3, a Kerberos configuration file is required, defining the default encryption types and domain to Kerberos Realm mappings.

Another architectural change for SAS 9.4 is the removal of SAS Foundation Services, or SAS Remote Services as it is sometimes called. With SAS 9.4, a new set of services within the SAS Web Application Server provide the functionality previously provided by SAS Foundation Services. This includes a new Central Authentication Service (CAS), which provides the link from the web applications back to the SAS Metadata Server and other IOM servers. As such, the only other item that needs to be configured for SPENGO authentication with SAS 9.4 is the SAS Logon Manager.

The SAS 9.4 Logon Manager, as with previous releases, must be updated to include a security constraint within the web.xml of the application. In addition, the three files controlling how the SAS Logon Manager interacts with the Central Authentication Service must also be updated. Figure 11 illustrates the files that must be updated for the configuration of SPNEGO with the SAS Web Application Server.

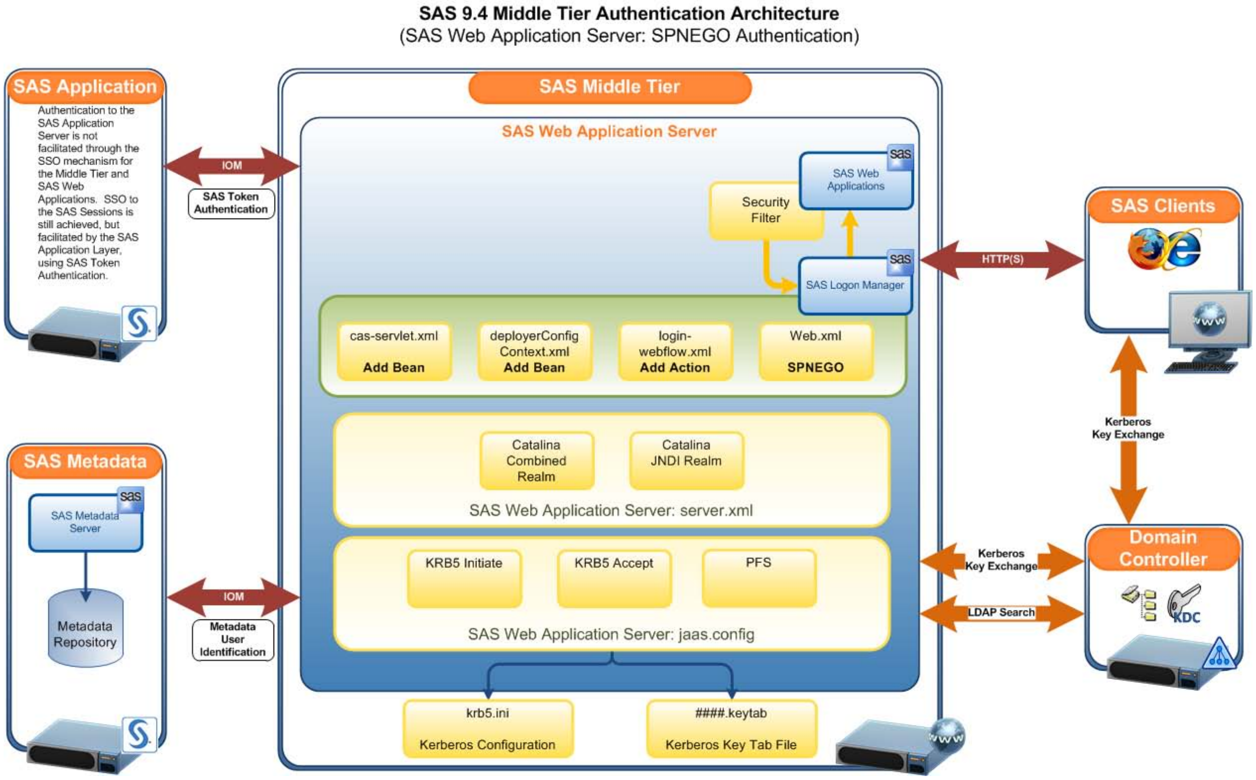

SAS 9.4 Middle Tier Authentication Architecture
(SAS Web Application Server: SPNEGO Authentication)
SAS Application
Authentication to the SAS Application Server is not facilitated through the SSO mechanism for the Middle Tier and SAS Web Applications. SSO to the SAS Sessions is still achieved, but facilitated by the SAS Application Layer, using SAS Token Authentication.
IOM
SAS Token Authentication
SAS Middle Tier
SAS Web Application Server
SAS Web Applications
Security Filter
SAS Logon Manager
cas-servlet.xml
Add Bean
deployerConfig Context.xml
Add Bean
login-webflow.xml
Add Action
Web.xml
SPNEGO
Catalina Combined Realm
Catalina JNDI Realm
SAS Web Application Server: server.xml
KRB5 Initiate
KRB5 Accept
PFS
SAS Web Application Server: jaas.config
krb5.ini
Kerberos Configuration
####.keytab
Kerberos Key Tab File
SAS Metadata
SAS Metadata Server
Metadata Repository
IOM
Metadata User Identification
SAS Clients
HTTP(S)
Kerberos Key Exchange
Domain Controller
Kerberos Key Exchange
LDAP Search
KDC

Figure 11: SAS 9.4 Middle Tier SPNEGO Configuration

An alternative scope for middle-tier authentication is from a stand-alone Java client to the middle tier. Within this scope, a Java client connects to the middle-tier services, is authenticated, and is then provided access to other SAS server processes. This scope of Java client to middle tier authentication is valid only for SAS 9.3 and SAS 9.4; this type of processing did not exist in SAS 9.2. For SAS 9.3, the stand-alone Java clients authenticating via the middle tier do not support authentication in the middle tier; these clients can use only authentication based on the SAS Metadata Server. SAS 9.4 introduces support within the middle-tier services for authentication in the middle tier and for using forwarded/delegated credentials to access other SAS server processes. More details are given in the next section where we examine the client tier.

CLIENT TIER

So far in our consideration of Kerberos authentication, we have not covered the third head of Kerberos: the clients. For SAS clients, we have three different types of clients. We have stand-alone clients that make direct connections to the IOM servers, we have stand-alone clients that make connections via the middle tier, and we have clients that run in a browser. From the first section of this paper where we reviewed the process of Kerberos authentication, it is clear that the client application must take an active part in the Kerberos authentication process. The client application must be able to request a service ticket from the KDC and process the service ticket once it has been created.

The stand-alone clients that make a direct connection to the IOM servers have all the necessary pieces to participate in Kerberos authentication. These clients include SAS Management Console, SAS® Data Integration Studio, and SAS Enterprise Guide. For these clients, the connection details entered in the connection profile are sufficient for the client to request the correct service ticket and authenticate the user.

For the clients who run in a browser, the browser does the processing of the ticket request. When a web application configured for SPNEGO is requested, the application server responds with a request for the browser to negotiate the authentication. As long as the browser has been configured to allow negotiated authentication for that site, the browser then requests a service ticket based on the URL of the site. Therefore, the web application does not require any code to handle the ticket requests because the browser handles everything.

Finally, our third type of SAS clients are those stand-alone clients that connect via the middle tier. Examples of these types of clients are SAS® Enterprise Miner, SAS® Forecast Studio, and SAS® Mobile BI. For all these clients, the authentication process works through services within the middle tier rather than through direct connections to the

SAS Metadata Server. These clients present more of a challenge for the configuration of Kerberos authentication. Although work is underway with SAS Enterprise Miner and SAS Forecast Studio to provide pass-through Kerberos authentication this is not complete. For the SAS Mobile BI client running on Android or iOS, the limitations of the operating system prevent the implementation of Kerberos authentication.

CONSIDERATIONS AND CONSTRAINTS FOR KERBEROS

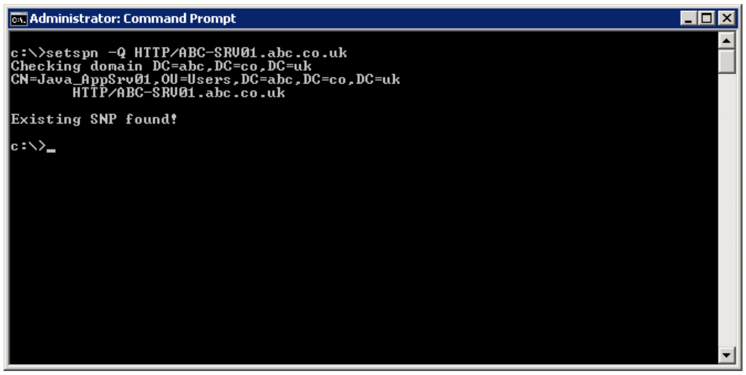
Given the information we now have about the process of Kerberos authentication and the way SAS implements Kerberos, we are now able to review the considerations and constraints for implementing Kerberos. The first constraint is an obvious one: The three heads of the Kerberos configuration must be within the same trusted domain structure. This does not mean that the server, KDC, and client must be in the same domain. The components can be in different subdomains, but a trust must exist between the domains for Kerberos authentication to work. The Microsoft TechNet article “How the Kerberos Version 5 Authentication Protocol Works” covers this in more detail, along with examples.

The KDC also needs to be able to find the service within Active Directory. The client will specify the Service Principal Name (SPN) as part of the request for the service ticket. The SPN must be unique within Active Directory, and only one result is returned when the KDC searches Active Directory. If there are multiple versions of the same SPN registered in Active Directory, the KDC does not issue a service ticket and an error is returned to the client. Equally, if no SPN is found registered in Active Directory, the KDC is unable to issue a service ticket and an error is returned to the client.

Therefore, it is important to ensure that the correct SPN is registered. If the server will be accessed via a DNS alias, then the SPN for the alias needs to be registered. Remember for the SAS Servers, the automated process on Windows, or the Quest Tools on UNIX, will only register SPNs for the host name of the server. So it requires additional manual steps to register the SPN for any DNS aliases. These additional SPNs can still be registered against the same object in Active Directory; the computer object for Windows servers and the service account for UNIX servers. The uniqueness constraint is against only the SPN and not the account, so multiple SPNs can be registered against the same account.

The consideration about using DNS aliases also applies to the SAS middle tier and the use of aliases or proxy servers for the SAS middle tier. Remember that the client constructs the SPN for the SAS middle tier based on the URL of the SAS Logon Manager. This URL is specified in the metadata properties for the SAS Logon Manager. Therefore, if you use a proxy server and have defined the connection for the SAS Logon Manager against the proxy server, the SPN of the proxy server is the one required by the KDC. This means that even if you have a horizontal cluster for the SAS middle tier running across several machines, you need to register only the single SPN of the proxy server used as the entry to the cluster.

Microsoft provides a command-line tool for interacting with the SPNs registered in Active Directory. The SETSPN tool enables users with sufficient domain privileges to add, edit, and delete SPNs against user and computer objects in Active Directory. SETSPN also allows domain users to search Active Directory for SPNs. The version of SETSPN released with Windows 2008 R2 and Windows 7 has additional functionality not available in earlier releases. In this version of SETSPN, the command-line switch -Q allows for searching Active Directory for a specific SPN. Figure 12 shows the output from SETSPN -Q.



```
c:\>setspn -Q HTTP/ABC-SRV01.abc.co.uk
Checking domain DC=abc,DC=co,DC=uk
CN=Java_AppSrv01,OU=Users,DC=abc,DC=co,DC=uk
        HTTP/ABC-SRV01.abc.co.uk

Existing SNP found!

c:\>_
```

Figure 12: Example Output for SETSPN -Q

Once the SPN is in place, the KDC will be able to issue the correct service ticket. From the review of the Kerberos authentication process, we know the tickets are encrypted. The Kerberos on Windows domain controllers supports different encryption types and key lengths. Versions of Windows prior to 2008R2 default to using DES-CBC encryption. However, in Windows 7 and Windows 2008 R2, the DES-CBC encryption type is disabled by default and the RC4-HMAC type is used instead. The encryption type used is important when creating the Kerberos keytab file used by the SAS middle tier application servers. The encryption type used to create the keytab file should match the encryption type used by the domain controllers.

Within the service ticket we have authorization data. In the Microsoft implementation, this field is significant because it contains the user's security identifier (SID) and the SIDs of groups the client belongs to. Therefore, if the end user is a member of a large number of groups within Active Directory, the size of the ticket can grow. Microsoft provides the following formula for calculating the ticket size:

$$\text{TicketSize} = 1200 + 40\,d + 8\,s$$

This formula uses the following values:

- d: The number of domain local groups a user is a member of plus the number of universal groups outside the user's account domain plus the number of groups represented in SID history.
- s: The number of security global groups that a user is a member of plus the number of universal groups in a user's account domain.
- 1200: The estimated value for ticket overhead. This value can vary depending on factors such as DNS domain name length, client name, and other factors.

In extreme cases, the size of the Kerberos ticket can cause issues. With SAS middle tier SPNEGO authentication, the service ticket is sent as part of the HTTP headers. JBoss has a default header size of 8 KB along with the Apache HTTP server. So in this case, if the service ticket is larger than the allowed header size, the authentication process will fail. With SAS 9.4, the middle tier is configured with a default header size of 16 KB, so further customization is not required.

Finally, as a consideration for Kerberos authentication, we need to consider accessing additional resources after the initial logon. Kerberos provides a seamless logon process, but issues can arise when the end user attempts to access another service after logging in. In the SAS terminology, these attempts are termed outbound authentication. Examples of such outbound authentication are accessing a SAS Workspace Server once logged in to the middle tier, accessing a third-party database, or accessing a network share from SAS Enterprise Guide. Kerberos provides a mechanism for accessing additional resources in the form of delegation.

Delegation allows a SAS Enterprise Guide user to pass their credentials to the SAS object spawner. The SAS object spawner then impersonates the user for access to additional resources. This requires the account running the SAS object spawner to have the security permission “trusted for delegation”. By default on a Windows server, the account running the SAS object spawner is the local system account. It is advisable to change the account running the SAS object spawner before granting the additional security permission. Otherwise, you are enabling any process running on that server to impersonate end users.

Delegation does not solve all issues for outbound authentication. For users connecting to the SAS middle tier from a browser with SPNEGO, the credentials are not forwarded from the Java application server. Therefore, there are no Kerberos credentials to be reused. In the case of web-authenticated users an alternative must be used. For example, an outbound credential could be saved in metadata to enable access to a third-party database. Or to allow access to a standard SAS Workspace Server a launch credential could be used for the SAS Workspace Server.

COMMON ISSUES WITH KERBEROS

In this final section, we present some common Kerberos issues and solutions. First, if the SPN is not set or has been set incorrectly, the following error messages are shown:

- KDC_ERR_C_PRINCIPAL_UNKNOWN: Client not found in Kerberos database
- KDC_ERR_S_PRINCIPAL_UNKNOWN: Server not found in Kerberos database
- KRB_AP_ERR_MODIFIED: Message stream modified
- KDC_ERR_MUST_USE_USER2USER: Server principal valid for user2user only
- KDC_ERR_PRINCIPAL_NOT_UNIQUE: Multiple principal entries in database
- KRB_AP_ERR_BAD_INTEGRITY: Integrity check on decrypted field failed

Use the SETSPN command to check the validity of the SPN and, if necessary, manually correct the SPN.

If duplicate SPNs are registered, the following errors are returned:

- KDC_ERR_PRINCIPAL_NOT_UNIQUE: Multiple principal entries in database
- KRB_AP_ERR_MODIFIED: Message stream modified errors

Again, use the SETSPN command to find the duplicate SPN entries and delete them.

If there are issues with the encryption types used, the following error is returned:

- KDC_ERR_ETYPE_NOTSUPP: KDC has no support for encryption type

Here you need to consider the type of encryption expected by the KDC. Remember that for Windows 7 and Windows 2008 R2, DES is disabled by default.

Time synchronization is important to the participants in Kerberos authentication. Issues with time synchronization return errors:

- KRB_AP_ERR_SKEW: Clock skew too great
- KRB_AP_ERR_TKT_EXPIRED: Ticket expired
- KDC_ERR_PREAUTH_FAILED: Pre-authentication information was invalid
- KDC_ERR_NEVER_VALID: Requested start time is later than end time
- KRB_AP_ERR_TKT_NYV: Ticket not yet valid

For these cases you should look to synchronize the time between the three Kerberos heads: the server, the KDC, and the client.

Issues with the size of service tickets present an error message

- KRB_ERR_RESPONSE_TOO_BIG: Response too big for UDP, retry with TCP

This error itself should not pose a problem. The communication will automatically switch to TCP and be retried. However, if there are firewalls involved, the required TCP port 88 will need to be open for this to succeed. The size of the token is more likely to cause issues with SAS middle tier authentication when the size is greater than the allowed HTTP header. This is unlikely to result in any specific error messages as an incomplete and hence invalid ticket is received. In these cases, the middle tier applications, such as JBoss or Apache HTTP server, will need to be configured to allow larger HTTP headers.

CONCLUSION

The Kerberos authentication protocol is an industry standard authentication protocol providing strong authentication for client/server applications. In this paper, we have reviewed how the Kerberos protocol operates and seen in some detail the steps involved in the authentication process. Understanding the Kerberos process, we have then reviewed how the SAS Business Analytics Framework is able to leverage Kerberos for the different types of authentication within the framework.

We have seen how Kerberos can provide single sign-on from a range of SAS clients to a range of SAS servers, including the SAS middle tier. We have reviewed how the implementation of Kerberos has changed across several releases of the SAS Business Analytics Framework from SAS 9.2 to SAS 9.4. We have also identified the cases where Kerberos authentication is not yet possible.

In addition, we have reviewed some of the constraints for using Kerberos authentication. We have seen how the use of Kerberos is tied into the domain trust structure and seen some of the prerequisites of using Kerberos. Of these prerequisites, we have identified the clear importance of the Service Principal Name and how this is central to the Kerberos authentication protocol.

Finally, we have reviewed some of the common issues that arise when using Kerberos and identified solutions to those error messages. Although Kerberos authentication issues are beyond the scope of this paper, it should be made easier by the understanding of the authentication process we have presented. Understanding the steps in the authentication process is the first and biggest step in being able to troubleshoot issues.

REFERENCES

Massachusetts Institute of Technology. “Kerberos: The Network Authentication Protocol.” Available at http://web.mit.edu/kerberos/.

Microsoft. "Microsoft Kerberos (Windows)." Available at http://msdn.microsoft.com/en-us/library/windows/desktop/aa378747(v=vs.85).aspx.

Microsoft. "Kerberos Authentication Overview." Available at http://technet.microsoft.com/en-us/library/hh831553.aspx.

Microsoft. "How the Kerberos Version 5 Authentication Protocol Works." Available at http://technet.microsoft.com/en-us/library/cc772815%28v=ws.10%29.aspx.

Quest Authentication Services. Available at http://www.quest.com/authentication-services/.

Rogers, Stuart and Heesun Park. 2011. "Single Sign-On Configuration and Troubleshooting for SAS 9.2 Enterprise BI Web Applications." Proceeding of the SAS Global 2011 Conference. Available at http://support.sas.com/resources/papers/proceedings11/365-2011.pdf .

SAS Institute Inc. 2010. "Configuring Integrated Windows Authentication for IBM WebSphere with SAS 9.2 Web Applications." Available at http://support.sas.com/resources/thirdpartysupport/v92m3/appservers/IWAWebSphere.pdf.

SAS Institute Inc. 2012. "Configuring Integrated Windows Authentication for Oracle WebLogic with SAS 9.2 Web Applications." Available at http://support.sas.com/resources/thirdpartysupport/v92m3/appservers/IWAWebLogic.pdf.

SAS Institute Inc. 2010. "Configuring Integrated Windows Authentication for JBoss with SAS 9.2 Web Applications." Available at http://support.sas.com/resources/thirdpartysupport/v92m3/appservers/IWAJBoss.pdf.

SAS Institute Inc. 2011. "Configuring Integrated Windows Authentication for IBM WebSphere 7.0 with SAS 9.3 Web Applications." Available at http://support.sas.com/resources/thirdpartysupport/v93/appservers/IWAWebSphere.pdf.

SAS Institute Inc. 2012. "Configuring Integrated Windows Authentication for Oracle WebLogic with SAS 9.3 Web Applications." Available at http://support.sas.com/resources/thirdpartysupport/v93/appservers/IWAWebLogic.pdf.

SAS Institute Inc. 2012. "Configuring Integrated Windows Authentication for JBoss with SAS 9.3 Web Applications." Available at http://support.sas.com/resources/thirdpartysupport/v93/appservers/IWAJBoss.pdf.

SAS Institute Inc. SAS 9.2 Intelligence Platform Documentation. Available at http://support.sas.com/92administration

SAS Institute Inc. SAS 9.3 Intelligence Platform Documentation. Available at http://support.sas.com/93administration

SAS Institute Inc. SAS 9.4 Intelligence Platform Documentation. Available at a future date at http://support.sas.com/94administration

SAS Institute Inc. "Configuration Guide for SAS 9.4 Foundation for UNIX Environments." Available at a future date at http://support.sas.com/documentation/installcenter

ACKNOWLEDGMENTS

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

- Larry Noe
- Saravana Chandran
- Mike Roda

CONTACT INFORMATION

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

Stuart Rogers
SAS Institute Inc.
SAS Campus Drive
E-mail: stuart.rogers@sas.com