

## Best Practices in SAS® 9 Security Configurations

Larry Noe, SAS Institute Inc, Cary, NC

### ABSTRACT

SAS®9 provides many enhancements in the area of single sign-on technology. This paper presents several best-practice configurations for systems that are based on Windows and systems that are based on other operating systems. These configurations maximize the use of single sign-on technology and minimize the necessity to store and pass system credentials. For all systems, we focus on identity-passing technology in SAS® 9, and new ways to configure servers for better security in common UNIX and z/OS deployments. In addition, for Windows, we will discuss the configuration of Integrated Windows Authentication for single sign-on.

### INTRODUCTION

SAS®9.2 offers new technologies for easier deployment and more secure use of back-end servers. The use of SAS token authentication allows the passing of client identity for better auditing and increased security. Most servers that are defined in metadata are set up to use SAS token authentication, which is not configurable. The SAS Workspace Server, however, is a more complex server and might require some configuration. It can be set up for host authentication, Integrated Windows authentication, or SAS token authentication. This paper explains SAS token authentication and reviews in detail the configuration options for SAS Workspace Servers. In addition, I will examine the benefits of SAS token authentication with respect to previous releases in a common UNIX and z/OS deployment scenario.

### SAS TOKEN AUTHENTICATION

SAS token authentication is a method of authentication that is used by the SAS system to pass client identity safely between applications and SAS servers after a user has successfully authenticated to the SAS Metadata Server. These tokens are generated and authenticated by the SAS Metadata Server. They consist of random strings with limited lifetime and one-time usability. By default, these tokens are always passed over the network encrypted for added security. Applications generate SAS tokens to allow users to connect to SAS servers without re-authenticating with external credentials. The tokens remove the need to store external credentials in the metadata repository, remove the risk of sending external credentials over the network, and improve performance by removing the requirement of sometimes expensive, external authentication.

Three servers that are defined in metadata are automatically configured for SAS token authentication: the SAS® Stored Process Server, the SAS® OLAP Server, and the SAS® Table Server. The authentication mechanism that is used for these servers cannot be modified. Any application that configures a connection to one of these servers by reading the server metadata will use SAS token authentication to establish an authenticated connection under the client user's identity. An example of a typical workflow is presented here:

1. Using a SAS application such as SAS® Enterprise Guide®, SAS® OLAP Cube Studio, or a web application, a user uses some supported technology, such as username/password or Integrated Windows authentication, to make an authenticated connection to the SAS® Metadata Server.. This establishes client identity on the back-end.
2. The user requests access to a cube associated in metadata with the SAS OLAP Server.
3. The application reads the server configuration information about the SAS OLAP Server and requests a SAS token from the SAS Metadata Server.
4. The application forwards the SAS token to the SAS OLAP Server to create an authenticated client connection to the server.
5. The SAS OLAP Server sends the SAS token to the SAS Metadata Server for validation and receives the client's user name as part of the response.
6. The SAS OLAP Server validates that the user has access to the requested cube and it returns the requested cube results.

SAS token authentication easily accomplishes single sign-on while propagating client identity for access control and auditing purposes.

## SAS STANDARD WORKSPACE SERVER CONFIGURATION

The SAS Workspace Server is a different kind of server than the ones we have just discussed. The complexity in configuring a SAS Workspace Server comes mostly from the fact that workspace servers are launched with user-supplied credentials by the SAS Object Spawner. In SAS 9.2, these credentials can be a host user name and password or, on Windows systems only, an Integrated Windows authentication token. Servers such as the SAS Metadata Server, the SAS OLAP server, and the SAS Table server are not launched by the SAS Object Spawner. They authenticate clients and act on their behalf, but they do not run under client-supplied credentials. Supporting the ability to run under client-supplied credentials increases the number of options available for authentication and process launching. The SAS® Stored Process server, although it is also launched by the SAS Object Spawner, is not launched under client-supplied credentials. It is a multi-user server launched under credentials obtained by the SAS Object Spawner.

### SERVER ACCESS SECURITY

The properties screen of the Logical Workspace Server using the SAS® Management Console offers a number of options to consider. The first important new feature is Server Access Security, is enabled by default. Server Access Security instructs the SAS Object Spawner, or the server itself, to validate that the client who is requesting access to the server in the metadata repository actually has access to that server. This prevents direct connections to a server by a client application that has bypassed the SAS® Metadata Server. Servers such as the SAS OLAP server and SAS Table Server do not enable Server Access Security by default because they enforce metadata access controls themselves and already determine what objects are accessible by connecting clients.

### THE AUTHENTICATION SERVICE SECTION AND HOST AUTHENTICATION

The next new feature in the Logical Workspace Server properties is the Authentication Service section. The default authentication method of a SAS Workspace Server is standard username/password authentication. This method of authentication is the same in SAS® 9.2 as it was in SAS® 9.1. An application that needs a workspace server for a client request will search for appropriate credentials using the Authentication Domain name of the workspace server with which the workspace server can be launched. These must always be host credentials and can either be cached, that is, reuse a login that was entered when the user connected to the metadata server, or looked up in metadata. When they are looked up in metadata, the credentials can either be personal credentials or from a group of which the user is a member. This is the only supported configuration in SAS 9.1, and it is still a common scenario in SAS 9.2.

#### INTEGRATED WINDOWS AUTHENTICATION

SAS 9.2 offers additional flexibility. The field labeled **Security Package** tells an application which type of host authentication the server requires. In addition to standard username/password authentication, SAS 9.2 supports Integrated Windows authentication on Microsoft® Windows systems. Integrated Windows Authentication (IWA) allows a user's identity to be passed safely using the underlying protocols of Kerberos or NTLM, which are supported by Windows. The drop-down list provides three choices: **Kerberos**, **NTLM**, and **Negotiate**.

The recommended choice for IWA is **Negotiate**, which allows Windows to choose the appropriate protocol for the client identity and the server that is being requested from the list that is specified in the **Security Package List** field. The default security package list is **Kerberos**, **NTLM**. These choices match the defaults for SAS servers. It is vital for client applications and servers to have matching protocol selections. The ability to select the direct protocols of Kerberos and NTLM is provided for completeness, and it is supported on the servers. It does, however, require option changes in addition to other client-side changes. If a site wants to force a specific protocol, the easiest solution is to modify the default list that is provided in this section and to add options to the server configuration to change the default list to specify only the desired protocol: either Kerberos or NTLM, but not both.

The final field in the configuration of IWA for workspace servers is the **Service Principal Name** or **SPN**. This field should almost always be left blank. In a default configuration, servers launched as Windows services will register a default name that the client also constructs and uses to establish a connection to this server. If the default names cannot be registered, or if the site wants to use some other naming convention, an SPN would have to be supplied

here and in other client configuration profiles. For more information on SPNs, please consult Microsoft Windows documentation.

## SAS TOKEN AUTHENTICATION FOR WORKSPACE SERVERS

SAS 9.2 also offers the ability to use SAS token authentication for SAS Workspace Servers. This might seem counter-intuitive because a host-based credential is required to launch a workspace server, but the key lies in providing a host launch credential to the SAS Object Spawner during configuration. The setup is similar to that of the SAS® Stored Process Server. We will discuss this type of configuration after reviewing the method that it is intended to replace, which is the use of group credentials.

### USING GROUP LOGIN CREDENTIALS

For many UNIX sites, and possibly z/OS sites, IT departments do not want to create and manage back-end credentials for client users. The alternative is to use a standard LDAP or Active Directory server against which client users authenticate. The SAS Metadata Server has the ability to configure alternate authentication providers to perform this task. This solves the problem of not creating back-end accounts for all users, but back-end credentials are still required to launch a workspace-server process. The solution to this is to create a group or a small set of groups, and associate a host login with each group. Users are then made members of these groups to allow them access to this host login. The host login is used by applications to launch the necessary workspace servers for application users. This was the only way to reduce the number of back-end credentials in SAS 9.1. It is still supported in SAS 9.2 for backward compatibility. Note that there is a cost in terms of security when using this approach: a server is launched under a shared back-end credential, so no individual access controls are possible for physical OS access to resources. On Microsoft Windows boxes, users almost always have host, domain credentials, which means this sort of solution is atypical for those environments.

### COMMON PROBLEMS WITH GROUP LOGINS

For many IT shops, group logins are certainly an improvement over having to maintain back-end credentials on systems other than Windows; however, it has three main problems:

- Only the group login is sent to the SAS Object Spawner.
- Library pre-assignment and all metadata access take place under the group-login identity.
- Users have access to the back-end login credential.

It becomes extremely difficult to track client usage of workspace servers in this configuration. Because the SAS Object Spawner sees only a request using the group login, it has no idea which individual client is making the request. It can audit the request only as the group. The inability to connect running workspace servers with clients who are using those workspace servers creates an auditing problem for IT.

When the server is launched using the group login, the identity of the group login is used to make a connection to the metadata server. Library pre-assignment occurs using the group metadata identity. This means that every library that is needed by every user in the group must be pre-assigned for correct operation. This is a security risk because abuse would be possible if great care is not taken in assigning group memberships and setting up libraries.

There is also a potential disconnect between what the client can see in metadata and what the group can see. If a client application sees a pre-assigned library while it is connected as the client, code will be generated that does not assign the library. If the setup is flawed and the group is not allowed access to that library, the library will not be pre-assigned at server startup, and the code will fail.

Care must be taken that every defined library is made accessible by both appropriate clients and by the group to which they belong containing their back-end host credential. This configuration can also cause performance problems as the number of pre-assigned libraries increases. In addition, any other metadata access done from running code will also be done under the group identity unless overriding credentials are specified directly. Individual user identity is not propagated in this setup.

Finally, each member of the group has legitimate access to the group credential because it is required in order to launch a workspace server. Given standard tools or custom XML, a user could obtain the back-end login. Most sites would prefer that users do not have access to such credentials. Setting up a group credential obscures access, but it does not prevent access.

### SETTING UP SAS TOKEN AUTHENTICATION

SAS token authentication offers many advantages, and it is fairly simple to set it up instead of group credentials. The first step is to choose the **SAS token authentication** radio button on the **Options** tab of the Logical Workspace Server properties screen. The next step is to select an appropriate launch credential. The chosen launch credential will take the place of the group login as the identity under which workspace servers will be launched. The chosen login must be accessible by the SAS Object Spawner. Because the Object Spawner connects to the metadata server using the sastrust account, the credential must be accessible by the sastrust user in metadata. The best practice for adding such credentials is to create a group, add the host login to the group, and make the sastrust user a member of the group. For users migrating to SAS 9.2 from SAS 9.1, it is a simple matter to remove users as members of the groups that they have already created, and add the sastrust user as the only member instead.

After refreshing or restarting the SAS Object Spawner, the SAS Workspace Server is now configured for SAS token authentication. Applications treat the workspace server the same way that they treat any other SAS server configured in this manner: a SAS token is requested and sent to the Object Spawner to request a workspace server. The Object Spawner validates that the user has access to the requested server (if Server Access Security is enabled), and launches the server under the specified launch credential.

This configuration solves all of the problems outlined in the section describing group credentials. The Object Spawner gets requests for servers with a SAS token, so the client identity is known and logged for audit purposes. Each launched server can be connected to an individual client request by user name and process ID. Library pre-assignment takes place under client identity, not group identity. The Object Spawner creates a new SAS token attached to the launched server that is used by the server to connect to the SAS Metadata Server under client identity. This creates an audit trail and performs library pre-assignment of client-accessible libraries instead of group-accessible libraries, which increases performance by limiting the number of pre-assigned libraries and reduces the security exposure of group assignment. In addition, all metadata access from running code can be done under client identity. This includes such things as the use of the Metadata LIBNAME Engine, proc OLAP, DATA step functions, and so on. Finally, only the Object Spawner has access to the launch credential through its connection to the metadata server as the sastrust. Users should have no access to the back-end credential through any means. The Object Spawner accesses the appropriate credential on behalf of the requesting client user instead of relying on the user to have access to a back-end group login.

Obviously, the workspace server process itself still runs under a back-end credential for all clients that are accessing that server. This requires that care be taken to prevent unwanted back-end access or access to SAS data sets because physical, OS access will still occur under the identity of the back-end login.

### ADDITIONAL CONFIGURATION OPTIONS

It is possible with group logins to set up multiple groups and make different client users members of different groups. This allows some separation of users into distinct back-end accounts when more security is required for physical or metadata access. The same thing can be accomplished using SAS token authentication. The Server Manager in the SAS Management Console allows an administrator to create multiple servers under a single Logical Workspace Server. Each of these servers can have a distinct launch credential. Metadata permissions are then used to control access to the different servers to force users to a specific server with a specific launch credential. There is tremendous flexibility in this configuration. Not only can individual-access controls be used along with group-access controls, but the commands and options can be modified for each server defined. Some servers could be allowed larger memory sizes or work libraries, some servers could be allowed to do X commands, others not, and so on. Any options and configuration files could be modified on a per command basis. In addition, all of the server definitions can specify the same port. The spawner can still distinguish between requests based on the server name in the client request, so no additional ports need to be opened regardless of the number of server definitions required for security separation.

The group login model also offers the flexibility of allowing power users to launch servers under their own back-end credentials. Using the old configuration, these users would not be members of any groups with a back-end login. This can be accomplished only in the new configuration by defining an additional server context with an additional logical

workspace server, and configuring that workspace server for the desired method of host authentication: either username/password or IWA. Power users with back-end credentials would use that server context for their work. The other server context would be used for general-purpose users and testing by power users, which provides a good separation of function and back-end access. Metadata access controls would be used to prevent any access by general-purpose users to the power-user server context, but without back-end credentials available to them, accessing this server context would be useless anyway.

## **CONCLUSION**

SAS® 9.2 offers substantial improvements in single sign-on using industry standard technologies such as Integrated Windows Authentication, as well as other methods to accomplish safe client identity passing. Improved configuration methods provide better security on all supported systems. SAS token authentication reduces the need for stored credentials and allows seamless communication between SAS clients and servers.

## **RECOMMENDED READING**

The most important information about security and server configuration in SAS 9 is the *SAS 9.2 Intelligence Platform Security Administration Guide*.

## **CONTACT INFORMATION**

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

Larry Noe  
SAS Institute Inc.  
100 SAS Campus Drive  
Cary, NC 27511  
Work Phone: 919-677-8000  
E-mail: Larry.Noe@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.