

# Tune Your SAS® Authentication by Facing Your Daemons

Sirshendu(Sirsh) Ghosh, Royal Bank of Scotland

## ABSTRACT

One of the most common concerns for large organizations is the risk of performance degradation resulting from an increase in demand of SAS® workloads. The most common pattern in a Linux platform is the delegation of authentication requests to the corporate Active Directory via Pluggable Authentication Modules (PAM) using the System Security Services Daemon (SSSD), an area that is not sufficiently understood nor explored as yet. A SAS administrator is expected to minimize the impact that an increase in user concurrency brings to the SAS login elapse time which can cause performance degradation.

Did you know that such performance problems can be addressed by enabling and configuring a brand new multithreaded SAS authentication daemon available on the latest version of SAS? Did you also know that the authentication process can be further improved by configuring SSSD caching options which will avoid multiple calls to Active Directory by the same user session?

Through a combination of both these daemons, we can remarkably improve the performance of your SAS® Platform to support a large number of concurrent users.

It's time to face your daemons.

## INTRODUCTION

The authentication process is a key element for any SAS application. In recent times, SAS environments are built on Linux platforms with its authentication method as PAM to allow it to integrate with the organization's Active Directory. If the servers are using the SSSD service to communicate to AD via PAM then SSSD becomes a very important service and as such one must make sure that the authentication process is well tuned.

Ask yourself this, what happens when there are concurrent SAS users trying to authenticate at the same time? Traditionally, the SAS authentication service (sasauth) has been single-threaded and hence at high concurrency rates you are likely to experience a "queuing" of authentication requests. There are two areas where we can tune to make sure concurrent authentications become faster – the SAS authentication service and the SSSD caching options.

SAS has released a new authentication daemon called "sas-services-daemon" for SAS 9.4M5 and M6 to improve on the concurrent nature of authentication requests by making it a configurable multi-threaded service. If we implement this along with few SSSD caching options, we can make the concurrent authentication process really fast.

## IMPLEMENTATION

### SAS AUTHENTICATION DAEMON

For SAS 9.4M5, SAS has released a hotfix (SAS Note 62837 and Hotfix HF2B6Q) which will introduce the new SAS authentication daemon called "sas-services-daemon".

Once the hotfix is installed, we need to configure the daemon following the instruction given in the hotfix implementation note (Installation Instructions for Hot Fix Container D2L001 on Linux x64).

We do not need to implement the hotfix for SAS 9.4M6 as it comes bundled with the install. In both versions the use of the daemon is optional, meaning that this is disabled by default and instead it's the sasauth service that the SAS application will use.

Please note that this daemon is bit different than the traditional sasauth in terms of how it is configured. For sasauth, it gets initiated by the authentication request from the SAS Metadata Server but the sas-services-daemon is a daemon which will always run on the servers and each authentication request will make use of it. This daemon can also be configured to start automatically at reboot on the server by init.d and can also be started manually using Linux's "service" or "systemctl" command.

It is recommended that appropriate sudo rules created for restricted SAS environments; this best practice will better adhere with proper production governance process by ensuring we are not unnecessarily exposed to the use of "root" account. Example sudo rules for the SAS install account are :

User <sas install account> may run the following commands on this host:

```
(root) NOPASSWD: service sas-services-daemon <status/start/stop/restart> [RHEL6]
```

```
(root) NOPASSWD: systemctl <status/start/stop/restart> sas-services-daemon [RHEL7]
```

Two important configuration parameters are :

- a) SAS\_USE\_SASSD in <SASCONFIG>/Lev#//level\_env\_usermods.sh
- b) numThreads in !SASROOT/utilities/bin/sas-services-daemon.conf

The *SAS\_USE\_SASSD* variable controls whether or not you want to use this multithreaded daemon. If you export the variable with the value 1 then it will use the daemon otherwise SAS will continue to use sasauth for authentication request.

*numThreads* is an important parameter which will determine how many concurrent threads will be used to delegate the request to SSSD. Based on our experience, if the value of the *numThreads* is equal to the number of cores in the server, we get the best result, however, it is recommended that the right value is derived from results of testing its performance as it may vary for other SAS environments.

## **SSSD CACHING OPTIONS**

Along with the multi-threaded authentication daemon, SSSD can also be tuned for optimal performance. Our testing suggests that if we use the SSSD caching options below, we get better performance for concurrent authentication requests.

These three options are :

- a) cache\_credentials
- b) entry\_cache\_timeout
- c) enum\_cache\_timeout
- d) memcache\_timeout
- e) ignore\_group\_members

By setting *cache\_credentials* to true, it is possible to enable offline credentials caching, which stores credentials (after successful login) as part of the user account in the SSSD cache.

*cached\_auth\_timeout* prevents SSSD from performing a backend authentication if it still has cached credentials that are younger than the value of this parameter.

*account\_cache\_expiration* is number of days entries are left in cache after last successful login before being removed during a cleanup of the cache. 0 means keep forever. The value of this parameter must be greater than or equal to *offline\_credentials\_expiration*.

*entry\_cache\_timeout* is how many seconds should nss\_sss consider entries to be valid for before asking the backend again.

Setting the *ignore\_group\_members* option to True makes all groups appear as empty, thus downloading only information about the group objects themselves and not their members, providing a significant performance boost. This should be only used for SAS Metadata Server hosts where it's not collocated with Compute Tier.

Below are the recommended values for these parameters based on our experience; however, they need to be derived from your own organization's security guidelines and performance experience.

cache\_credentials = true

entry\_cache\_timeout = 14400

enum\_cache\_timeout= 600

memcache\_timeout= 600

ignore\_group\_members = true

## PERFORMANCE UPLIFT METRICS

Here are some stats "before and after" configuring these two daemons:

Before		
Concurrency	AD via SSSD (Using SAS - Proc Permttest)	SAS Logon
1	0.09 sec	3 sec
30	2.79 sec (Avg)	6 sec
50	4.06 sec (Avg)	12 sec

After		
Concurrency	AD via SSSD (Using SAS - Proc Permttest)	SAS Logon
1	0.06 sec	2 sec
30	0.43 sec (Avg)	2 sec
50	0.78 sec (Avg)	6 sec

## CONCLUSION

As you can see from the uplift metrics, after tuning these two daemons, the authentication performance of the SAS platform improves significantly. Hence, you should not be afraid of controlling your daemons to improve your own platform and users' experience.

## ACKNOWLEDGMENTS

The hotfix was swiftly created for RBS with our request and I would like to thank SAS R&D and the account team for their quick turn around with it.

## RECOMMENDED READING

- <http://support.sas.com/documentation/installcenter/en/ikfdtnunxcg/66380/PDF/default/config.pdf>
- <http://support.sas.com/kb/62/837.html>
- <http://ftp.sas.com/techsup/download/hotfix/HF2/D/D2L/D2L001/xx/lax/D2L001la.html>

## CONTACT INFORMATION

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

Sirshendu(Sirsh) Ghosh  
Royal Bank of Scotland Group  
Regents House, London, N1 8XB  
Phone: +44 7917791566  
Email: Sirshendu.Ghosh@rbs.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.