**Paper SAS2142-2018**

# Multi-Factor Authentication with SAS® and Symantec VIP

Jody Steadman, Mike Roda, SAS Institute Inc.

## ABSTRACT

Organizations with strict security requirements might require that SAS® integrate with a multi-factor authentication (MFA) solution such as Symantec's Validation Identification Protection (VIP). While most organizations already secure VPN access with MFA, the need for it in a SAS environment might stem from the sensitive nature of the data SAS is accessing or compliance-related requirements. MFA requires that an authenticating user not only provide a valid user name and password, but also a one-time password in the form of a security code or approve a push notification sent to a registered mobile application. This paper will show how components in both SAS 9.4 and SAS® Viya® can be configured to integrate with Symantec VIP, so that any authentication attempt into SAS would require that a user not only successfully enter a user name and password, but also successfully respond to the other configured authentication factors.

## INTRODUCTION

The differences between SAS® 9.4 and SAS® Viya® architecture demand different approaches to integrating Symantec VIP MFA. A typical installation of SAS Visual Analytics for SAS 9.4 includes SAS Web Server, which is a reverse proxy, passing requests to the back-end web application servers. SAS Web Server uses Apache HTTPD and can integrate with a number of authentication mechanisms. The versatility of Apache HTTPD enables the configuration of Symantec's VIP Service at the proxy level. Since the user is already thoroughly vetted with user name, password, and MFA at the proxy, the web application server does not require further authentication of the user and the subsequent calls to Pluggable Authentication Modules (PAM) . SAS Enterprise Guide relies on the SASAUTH utility to provide authentication. Essentially, anything that can be done in PAM to authenticate can be done with SASAUTH. With the middle tier configured for web authentication at the proxy and therefore not relying on PAM, Symantec VIP can be configured in the SASAUTH configuration file to allow for split-password MFA. With split-password MFA, the user enters their user name and password and an appended security code into their respective fields when prompted by SAS Enterprise Guide.

Although SAS Viya also leverages the Apache HTTPD server as a reverse proxy on bare-OS environments, as of version 3.3, it does not support authentication done in the proxy. Nevertheless, SAS Viya 3.3 does support PAM authentication so that Symantec VIP can be configured in the PAM configuration file to allow for MFA. This has some advantages over using the Apache HTTPD module. By triggering MFA from the authentication itself, rather than a specific web login address, clients that do not use web logins but still authenticate with user credentials (for example, to obtain access tokens) also benefit from MFA. For example, a client that obtains access tokens can still take advantage of MFA. Other clients, such as the SAS Mobile BI application, scripts, and other software that authenticate, can call the REST APIs. Fortunately, with the exception of the programming-only environment and the SAS Studio application, SAS Viya does not repeatedly authenticate user credentials after the initial login. This results in only a single push notification to the user's device.

## SYMANTEC VIP ARCHITECTURE

The Symantec VIP architecture used in this example consists of VIP Manager, an Enterprise Gateway, a Self-Service Portal, and Symantec's VIP service cloud. The VIP Manager is a web page hosted by Symantec where VIP administrators log in to configure and administer VIP for their environment. The VIP Manager also contains links to documentation and various client downloads for integrating with specific technologies, configurations, and licensing information. The Symantec Enterprise Gateway is a service hosted at a site that provides specific configuration options like integration with AD/LDAP, configuration of

validation services such as RADIUS, and configuration of the identity provider. SAS Enterprise Guide connections use an integration to PAM, which is configured to communicate with the RADIUS server on the Symantec Enterprise Gateway. The Self-Service Portal is used to authenticate users who would like to register an application or key fob for use in the environment. The portal is integrated with Enterprise AD/LDAP. Users authenticate to the portal with those credentials, and then they register the serial number of their application or key fob for use in the environment. The VIP Service Cloud is hosted by Symantec, so it is not something that is configured by the end user. The integration point with SAS Visual Analytics uses the VIP Service Cloud for authentication with RADIUS.

This is a very high-level overview of the Symantec components used in this example. Reading Symantec's documentation on architecture, installation, and configuration is highly recommended. More information about Symantec VIP is available in Symantec's product documentation [1] or by contacting your Symantec Rep.

Symantec software is not shipped as a part of any SAS solution, so proper licensing from Symantec is required in order to use Symantec VIP.

## HOW IT WORKS

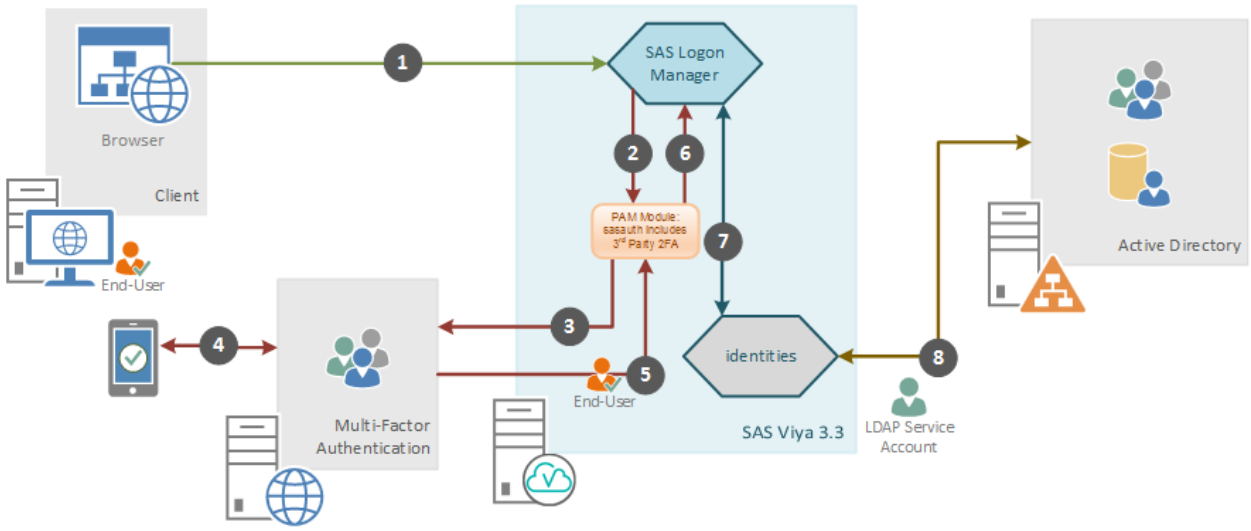


**Figure 1**

1) A connection to SAS Logon Manager is made via a web browser, which returns the login page for user credentials.

2) SAS Logon Manager hands the supplied credentials to the system (PAM).

3) PAM steps through the sasauth or sasauth-viya file to the point where it reaches the 2FA modules, checks group membership to determine if 2FA is required for the user, and determines if a password or password plus appended security code was supplied. The Symantec Enterprise Gateway is then contacted.

4) The Symantec Enterprise Gateway determines if the security code is correct or sends a push notification the user's registered application. If a user supplies the appropriate security code or a push notification is accepted, the Symantec Enterprise Gateway allows the authentication to continue and passes it back to PAM.

5) The rest of the sasauth or sasauth-viya file is processed, where the password is checked to determine if the authentication can continue. The result is passed to SAS Logon Manager.

6) SAS Logon is notified of the result of the authentication request and access is given accordingly.

7) SAS Logon Manager passes the information to the identities service.

8) User information is queried in LDAP, appropriate identity and group membership information is determined for the user, and the result returned to SAS Logon Manager.

## SAS 9.4

A full order of SAS 9.4M4 was used for this example. Since the example was functional in nature and not performant, the hardware consisted of a single virtual machine with 6 CPUs, 24GB of RAM, a 16GB disk for the OS, and 100GB disk for the SAS installation. RHEL 6.3 was used as the OS.

Here are some notable configuration options included during this installation of SAS:

Use PAM - This option enables PAM as the authentication method for SAS. This is not an optional setting as authentication in SAS Enterprise Guide with MFA is configured at the PAM level.

Use Token Authentication - This option enables SAS Token Authentication. This could be an optional setting, but it was set in the example environment.

Use External Accounts - Use of this option denotes that SAS users (like sastrust) are external accounts. This could be an optional setting, but it was set in the example environment.

Use Proxy - Use of this option sets up Apache HTTPD as a reverse proxy to the web application servers. Requests are filtered and then forwarded to the web application servers if filter requirements are met. This is not an optional setting as SAS Visual Analytics connections use an MFA configured proxy for authentication and subsequent web authentication to the web application servers.

### INSTALLING AND CONFIGURING APACHE VERSION 2.4

At the time of this writing, the SAS web server is based on Apache HTTPD version 2.2 and Symantec VIP integration with Apache HTTPD requires version 2.4, which includes a number of required modules for MFA to work properly. There are two options. One option is to install an Apache 2.4 based HTTPD external reverse proxy in front of the SAS web server. This is a standard practice for many customers that use their own proxy or load balancer in front of SAS. The configuration for installing an external reverse proxy is covered in the Advanced Topics section of *SAS® 9.4 Intelligence Platform: Middle-Tier Administration Guide* [2]. The other option is to swap out Apache HTTPD 2.2 and replace with 2.4. This is the approach taken in this paper. Note that this might not be an officially supported configuration.

To configure Apache 2.4 for use with SAS, perform the following steps:

1. Stop SAS using the sas.servers script.
2. Obtain Apache HTTPD version 2.4 and relevant Apache Modules. Binaries in RPM format are the easiest and quickest. These are available in the Red Hat Software Collections repository[3]. The default installation location for those RPMs is `/opt/rh/httpd24`. Here is a list of Apache HTTPD 2.4 related RPMs installed on the example machine.

   httpd24-httpd-tools
   httpd24-mod_ssl
   httpd24-runtime
   httpd24-apr-util
   httpd24-httpd
   httpd24-mod_session
   httpd24-apr
   httpd24

3. The httpdctl script used by SAS to start and stop Apache needs to be changed to reflect the location of the Apache HTTPD 2.4 installation. Edit the script and change the settings to their respective values. The script was located in `/opt/sasconfig/Lev1/Web/WebServer/bin` in the example, but the location is user configurable at SAS installation time. Here is an example.

   apache_root="/opt/rh/httpd24/root"
   apache_bin="$apache_root/usr/sbin/httpd"

LD_LIBRARY_PATH="$apache_root/usr/lib${LD_LIBRARY_PATH:+:}$LD_LIBRARY_PATH"

4. The file `/opt/sasconfig/Lev1/Web/WebServer/conf/httpd.conf` needs many edits in order to work with Apache HTTPD 2.4, so those changes are not itemized here. Your path to httpd.conf might be different because the location is user configurable at SAS installation time. Use the file below as a template.

```
ServerRoot "/opt/sasconfig/Lev1/Web/WebServer" Listen 7980
LoadModule authz_core_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_authz_core.so"
LoadModule lbmethod_byrequests_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_lbmethod_byrequests.s
o"
LoadModule slotmem_shm_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_slotmem_shm.so"
LoadModule unixd_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_unixd.so"
LoadModule access_compat_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_access_compat.so"
LoadModule mpm_worker_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_mpm_worker.so"
LoadModule authn_file_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_authn_file.so"
LoadModule authn_core_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_authn_core.so"
LoadModule authz_host_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_authz_host.so"
LoadModule authz_groupfile_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_authz_groupfile.so"
LoadModule authz_user_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_authz_user.so"
LoadModule auth_basic_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_auth_basic.so"
LoadModule cache_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_cache.so"
LoadModule cache_disk_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_cache_disk.so"
LoadModule filter_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_filter.so"
LoadModule deflate_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_deflate.so"
LoadModule log_config_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_log_config.so"
LoadModule env_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_env.so"
LoadModule expires_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_expires.so"
LoadModule headers_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_headers.so"
LoadModule setenvif_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_setenvif.so"
LoadModule proxy_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_proxy.so"
LoadModule proxy_connect_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_proxy_connect.so"
LoadModule proxy_http_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_proxy_http.so"
```

```
LoadModule proxy_balancer_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_proxy_balancer.so"
LoadModule ssl_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_ssl.so"
LoadModule mime_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_mime.so"
LoadModule status_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_status.so"
LoadModule autoindex_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_autoindex.so"
LoadModule dir_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_dir.so"
LoadModule alias_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_alias.so"
LoadModule rewrite_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_rewrite.so"
ServerAdmin root@localhost ServerName some.host.com:7980
DocumentRoot "/opt/sasconfig/Lev1/Web/WebServer/htdocs"
DefaultRuntimeDir logs/
<Directory / >
Options FollowSymLinks
AllowOverride none
Order deny,allow
Deny from all
</Directory>
<Directory "/opt/sasconfig/Lev1/Web/WebServer/htdocs">
Options None
AllowOverride None
Order allow,deny
Allow from all
</Directory>
<IfModule dir_module>
DirectoryIndex index.html
</IfModule>
<FilesMatch "^\.ht">
Order allow,deny
Deny from all
Satisfy All
</FilesMatch>
ErrorLog "|/opt/rh/httpd24/root/usr/sbin/rotatelogs -l
/opt/sasconfig/Lev1/Web/WebServer/logs/error_%Y-%m-%d-%H.%M.log 50M"
LogLevel warn
<IfModule log_config_module>
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
<IfModule logio_module>
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\" %I %O" combinedio
</IfModule>
CustomLog "|/opt/rh/httpd24/root/usr/sbin/rotatelogs -l
/opt/sasconfig/Lev1/Web/WebServer/logs/access_%Y-%m-%d-%H.%M.log 50M"
common
</IfModule>
<IfModule alias_module>
ScriptAlias /cgi-bin/ "/opt/sasconfig/Lev1/Web/WebServer/cgi-bin/"
</IfModule>
```

```
<Directory "/opt/sasconfig/Lev1/Web/WebServer/cgi-bin">
AllowOverride None
Options None
Order allow,deny
Allow from all
</Directory>
ForceType text/plain
<IfModule mime_module>
TypesConfig conf/mime.types
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/vnd.emscripten.memory.init.file .mem
</IfModule>
LimitRequestFieldSize 16384 SendBufferSize 16384
Include conf/extra/httpd-mpm.conf Include conf/extra/httpd-info.conf
Include conf/extra/httpd-default.conf
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed startup file:/dev/urandom 512
SSLRandomSeed connect builtin
SSLRandomSeed connect file:/dev/urandom 512
</IfModule>
Include conf/sas.conf
```

5. Execute `/opt/sasconfig/Lev1/Web/WebServer/bin/httpdctl start` to confirm that file syntax is ok. Note any errors in configuration at start up and resolve them. Execute `/opt/sasconfig/Lev1/Web/WebServer/bin/httpdctl stop` to ensure that HTTPD is down. Repeat this step as necessary until syntax is correct.
6. After the syntax is correct, you are finished with configuring Apache HTTPD 2.4 for use with SAS.
7. Stop Apache HTTPD before proceeding.

## CONFIGURING MFA FOR SAS 9.4 VISUAL ANALYTICS

Before configuring SAS Visual Analytics with MFA, make sure Symantec's Apache integration document is fully understood and that you have a working Symantec VIP environment. The steps in this section are in part a summarization of that document with respect to what it is needed to configure SAS.

Configuration of SAS Visual Analytics with MFA occurs in the two places. The first place is on the Symantec Enterprise Gateway. Symantec's Apache integration document details the steps that must be completed[4]. For example, one of these steps is to create a RADIUS service for Apache HTTP. The second place where configuration occurs is on the Apache Web Server itself. These steps are described in this section.

After you have met all prerequisites and a RADIUS server is configured on the Symantec Enterprise Gateway, proceed with the steps below.

1. Ensure that SAS is shutdown. If not, stop SAS using the sas.servers script.
2. Edit the symCAuth_radius.conf file and place it in the Apache HTTPD configuration directory. In the example, the configuration directory was `/opt/sasconfig/Lev1/Web/WebServer/conf`. You will need to know the IP and port details of the RADIUS server that you configured on the Symantec Enterprise Gateway and the IP address of the machine or interface that Apache HTTPD is using. Use the sample below as a template for setting up the symCAuth_radius.conf file.

```
authnType=radius
[RadiusCommon]
```

```
callAttempts=3
readTimeout=5
contextPoolCapacity=10
otpSize=6
clientIp=192.168.1.1
validationMode=ULO
stepupAuthentication=false
authenticationLevel=2
[RadiusServer#1]
serverIp=192.169.1.2
serverPort=1813
serverSecret=::encrypted::somesecret
```

3. The Apache httpd.conf file needs additional options to accommodate MFA. This is the same httpd.conf file that you modified when configuring for Apache HTTPD 2.4 earlier in this document. At the beginning of the LoadModule section, add the lines below. Adjust paths accordingly.

```
LoadModule authn_symc_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_authn_symc.so"
LoadModule auth_form_module "/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_auth_form.so"
LoadModule session_module "/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_session.so"
LoadModule request_module "/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_request.so"
LoadModule session_cookie_module
"/opt/rh/httpd24/root/usr/lib64/httpd/modules/mod_session_cookie.so"
```

After the directory directive for `/opt/sasconfig/Lev1/Web/WebServer/htdocs`, add the following lines:

```
<Location "/SASLogon/login">
AuthFormProvider symc
AuthType form
AuthName "VIP Authentication Protected Area"
Session On
SessionCookieName session path=/SASLogon
Require valid-user
ErrorDocument 401 /login_error.html
AuthSymcConf /opt/sasconfig/Lev1/Web/WebServer/conf/symcAuth_radius.conf
AuthSymcSplitPassword false
RewriteEngine On
RewriteCond %{LA-U:REMOTE_USER} (.+)
RewriteRule . - [E=RU:%1,NS]
RequestHeader set X-Remote-User "%{RU}e" env=RU
</Location>
```

4. The SASLogon.xml file used by the web application server needs to be configured for web authenticationwith Apache HTTP. The `/opt/sasconfig/Lev1/Web/WebAppServer/SASServer1_1/conf/Catalina/localhost` directory contained this file in the examle environment, but the location might differ depending on the location of the SAS installation. For this example, add the line below to the end of the context directive.

```
<Valve className="com.sas.vfabrictcsvr.authenticator.PrincipalFromRequestHeadersValve">
```

5. The initial login page used by MFA needs to be placed in the document root of Apache HTTP. In the example, the location of the document was `/opt/sasconfig/Lev1/Web/WebServer/htdocs`. The file must be called login_error.html. Copy and paste the following into login_error.html.

```
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" class="bg">
<head>
<meta charset="UTF-8" />
<link rel="shortcut icon" href="themes/default/images/favicon.ico" />
<title>SAS® Logon Manager</title>
<!-- [if IE 9] -->
<link type="text/css" rel="stylesheet" href="themes/default/css/sas_ie.css" />
<link type="text/css" rel="stylesheet" href="themes/default/css/sas.css" />
<meta name="viewport" content="initial-scale=1" />
</head>
<div id="nonModal" class="block">
<img src="themes/default/images/transparent.png" class="logo" alt="" /><!--customizable logo-->
<h1 class="logotext" alt="">Sign In to SAS<sup class="reg">®</sup></h1><!--Sign In to SAS-->
<div class="message" style='display:none;' id="nocookie-message" aria-hidden="true">
<h2 class="primary">This application requires that your browser accept cookies.</h2>
<p class="secondary">Change your browser settings accordingly.</p>
</div>
<div id="loginbox">
<form id="myForm" class="minimal" action="" method="post"><!--form container-->
<label for="username" id="username1">User ID:</label>
<input id="username" name="httpd_username" tabindex="3" aria-labelledby="username1
message1 message2 message3" autofocus="true" type="text" value="" autocomplete="off"/>
<label for="password">Password:</label>
<input id="password" name="httpd_password" tabindex="4" type="password" value="" size="25"
autocomplete="off"/>
<label for="otp">Security Code:</label>
<input id="http_securitycode" name="httpd_securitycode" tabindex="4" type="password" value=""
size="25" autocomplete="off"/>
<input type="hidden" name="lt" value="LT-3-z4r9T5Pofef01xmFGVoO9ouvrQv3cb" aria-
hidden="true" />
<input type="hidden" name="execution" value="e2s1" aria-hidden="true" />
<input type="hidden" name="_eventId" value="submit" aria-hidden="true" />
<input type="submit" class="btn-submit" name="Sign In"/ value="Login">
<div class="aboutcontainer"> <!--about link-->
<a href="#openModal" onClick="$('#openModal').show()" class="about" title="About">About</a>
<div class="copyright"><!--copyright statement-->
© 2002-2015 SAS Institute Inc.
</div>
<img src="themes/default/images/saslogo.svg" class="saslogo" />
</div>
</form>
</div>
</div>
<div id="openModal" class="modalDialog" style="z-index: 9999;"><!--modal container-->
<div>
<div class="test">
<div></div>
<div><a href="" onClick="$('#openModal').hide();" title="Done" class="done">Done</a></div><!--
done button-->
</div>
<!-- about dialog content -->
<br />
<p>Product name: SAS<sup>®</sup> Logon Manager</p>
<p>Release: 9.4</p>
<h2>Legal Notices</h2>
<p>Copyright 2002-2015, SAS Institute Inc., Cary, NC, USA. All Rights Reserved.
```

```
This software is protected by copyright laws and international treaties.</p>
<h3>U.S. Government Restricted Rights</h3>
<p>Use, duplication, or disclosure of this software and related documentation by the United
States government is subject to the license terms of the Agreement with SAS Institute Inc.
pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a) and DFAR
227.7202-4 and, to the extent required under United States federal law, the minimum restricted
rights as set out in FAR 52.227-19 (DEC 2007).</p>
<h3>Third-Party Software Usage</h3>
<h4>Central Authentication Service</h4>
<p>Copyright © 2007, JA-SIG, Inc. All rights reserved.</p>
<p>Redistribution and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met:</p>
<ul>
<li><p>Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.</p></li>
<li><p>Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials provided with
the distribution.</p></li>
<li><p>Neither the name of the JA-SIG, Inc. nor the names of its contributors might be used to
endorse or promote products derived from this software without specific prior written
permission.</p></li>
</ul>
<p>THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.</p>
<p><a href="https://www.apereo.org/cas/license" style="color:white"
target="_blank">https://www.apereo.org/cas/license</a></p>
</div>
</div>
</html>
```

6. If you are not configuring MFA for SAS 9.4 Enterprise Guide you can start SAS services and proceed with testing the installation.

The web authentication document details the steps that are required to set up web authentication between Apache HTTPD and the web application servers, including setting up a shared secret. [2] *It is highly recommended, if not required, that a shared secret be configured between Apache HTTPD and the web application server.* By not using a shared secret, the web application server will accept authentication from any machine that presents the proper user information in the header.

## CONFIGURING MFA FOR SAS 9.4 ENTERPRISE GUIDE CONNECTIONS

Symantec's PAM integration works with minimal changes to the SASAUTH PAM file. Symantec's PAM integration document should be thoroughly understood before proceeding. The steps in this section assume you have met the prerequisites and have the necessary configuration in the Symantec Enterprise Gateway.

The SSHD configuration section of Symantec's PAM integration document[5] should be used to test MFA for PAM. It is the easiest to set up and most readily available of the examples that exist on systems. You

need root privileges in order to make changes and you should have root in case you are locked out as the root account does not require MFA by default.

Once PAM integration is confirmed, proceed with the following steps.

1. Identify all SAS related users that are authenticated externally by SAS. These can include, but might not be limited to, sasadm, sastrust, webanon, and sas.
2. Place all of these users in a single system group. The group name sasaccts was used in the example and will be referenced in the next step.
3. Set up and edit the `/etc/raddb/vrsn_otp` file per Symantec's documentation. Make sure you include a no2fa line in the file, which basically excludes users within the specified group from MFA. The example file below was used in the example and includes the no2fa option configured for members of the sasaccts system group.

```
no2fa sasaccts
192.168.1.104:1812 somesecret 5 3
```

4. Add the lines below to the `/etc/pam.d/sasauth` file. These lines must be placed at the top of their respective sections (auth and account).

```
auth required /lib/security/pam_vrsn_otp.so split_password
account required /lib/security/pam_vrsn_otp.so split_password
```

   The following is a complete sasauth PAM file that can be used as an example.

```
#%PAM-1.0
auth required /lib/security/pam_vrsn_otp.so split_password
auth required pam_env.so
auth optional pam_krb5.so
auth sufficient pam_unix.so nullok try_first_pass
auth requisite pam_succeed_if.so uid >= 500 quiet
auth sufficient pam_sss.so use_first_pass
auth required pam_deny.so
account required /lib/security/pam_vrsn_otp.so split_password
account required pam_unix.so
account sufficient pam_localuser.so
account sufficient pam_succeed_if.so uid < 500 quiet
account [default=bad success=ok user_unknown=ignore] pam_sss.so
account required pam_permit.so
```

5. Start SAS and proceed with testing.

## SAS 9.4 VA TESTING

MFA for SAS VA testing requires that you have Symantec's VIP Access application installed on a supported mobile device. After installation, you must register the application in the self-service portal.

1. After installation and registration, open `http://va_host:7980/SASVisualAnalyticsHub` in your browser, where "va_host" equals the host name of the machine serving SAS Visual Analytics. You should see a login page with user name, password, and security code.
2. Enter a valid user name and password while leaving the security code field blank. Leaving security code blank will force a push notification to your registered mobile applications. If successful, you should see the SAS Visual Analytics Hub landing page. Log off and close the browser.
3. Open the login page again as described in step 1. Enter a valid user name, password, and current security code from your mobile application. If successful, you should see the SAS Visual Analytics Hub landing page.

### SAS 9.4 ENTERPRISE GUIDE TESTING

MFA for SAS Enterprise Guide requires that you have Symantec's VIP Access application installed on a supported mobile device. After installation, you must register the application in the self-service portal.

1. Open Enterprise Guide and create a profile for the SAS Metadata Server you will be connecting to from SAS Enterprise Guide to perform SAS operations. When creating the profile, do not store user name and password. SAS Enterprise Guide must prompt for user name and password since the password entered changes at each login.
2. Make the profile active and you should be prompted for a user name and password. In the user name field, enter a valid user name. In the password field, enter a valid password for the user and the security code from the VIP Access application on your mobile device. For example, if your password is "password" and the security code is "123456", then enter "password123456".
3. You should be successfully logged in to the SAS Metadata Server in SAS Enterprise Guide.

## SAS VIYA

SAS Viya 3.3 was used for this example. Since the example was functional in nature, and not performant, the hardware consisted of a single virtual machine with 6 vCPUs, 64GB of RAM, and a 50GB disk for the OS, and a 100GB disk for the SAS install. RHEL 7.4 was used as the OS.

The host was integrated with Microsoft Active Directory using SSSd. Since Symantec VIP is integrated with LDAP, it is important that the host itself resolve users from the same user store used by Symantec VIP. SSSd is one of the ways to accomplish this.

There are no notable options at install time. All applicable steps for the software order were performed following SAS documentation.

Most of the configuration performed is around the OS integration with Symantec VIP, the ability to resolve users in LDAP, and the grouping of users that require or do not require MFA. From a purely SAS perspective, the only requirements are changes to the web tier for PAM authentication, configuration changes to the SAS Viya sasauth file, and modification of the SAS services using LDAP for authentication.

### SYMANTEC VIP RADIUS VALIDATION SERVER CONFIGURATION

A custom RADIUS Validation Server configuration was needed for integration with SAS Viya. The applicable options used in the configuration are in Figure 2 below. For more details about these options, please see Symantec's documentation.

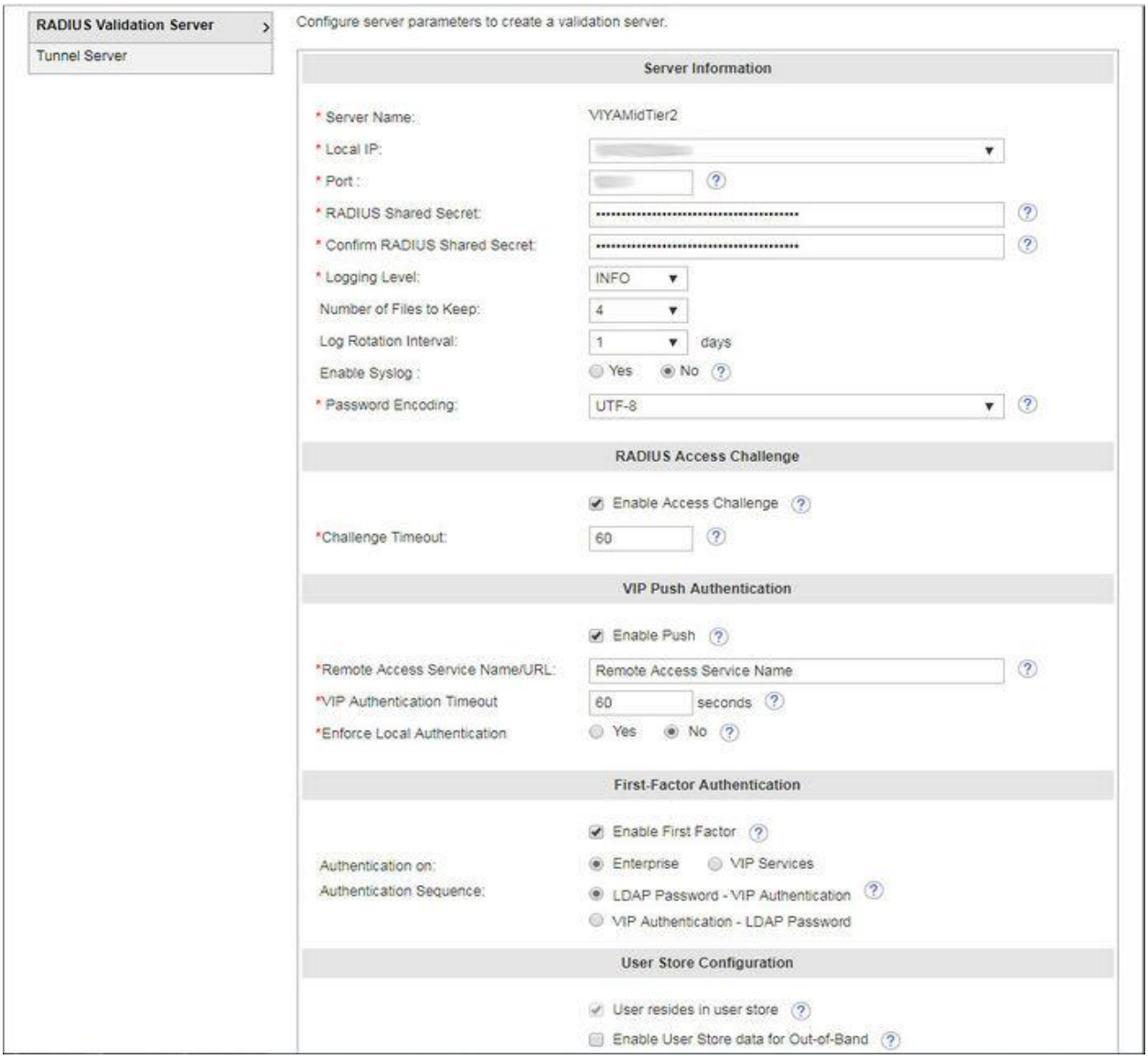


**Figure 2**

## CONFIGURING MFA FOR SAS VIYA VISUAL APPLICATIONS

Authentication in SAS Viya visual applications is handled by the SAS Logon Manager. Internally, the SAS Logon Manager uses a stack of authentication providers. By default this always includes support for authenticating internal database accounts such as the sasboot user, as well as authenticating against an external LDAP server such as Active Directory or OpenLDAP.

### Configuring PAM at the System Level

The next step involves configuring the SAS PAM file in `/etc/pam.d/sasauth-viya`. Since PAM is used by the SAS Viya web applications, this file must be configured so that the web application uses the Symantec VIP components and so that users requiring MFA are properly identified. This contents of this file are similar to what is used in SAS 9.4.

```
auth [default=ignore success=1] /lib64/security/pam_succeed_if.so user ingroup [no2fa]
auth [default=ignore success=1] /lib64/security/pam_succeed_if.so user ingroup [MFA]
auth [default=die success=done] pam_sss.so
```

```
auth [default=die success=done] /lib64/security/pam_vrsn_otp.so prompt=SecurityCode:
auth include pam_sepermit.so

account required pam_nologin.so
account [default=ignore success=1] /lib64/security/pam_succeed_if.so user ingroup [no2fa]
account [default=ignore success=1] /lib64/security/pam_succeed_if.so user ingroup [MFA]
account [default=bad success=ok user_unknown=ignore] pam_sss.so
account sufficient /lib64/security/pam_vrsn_otp.so prompt=SecurityCode:
account include system-auth

password [default=ignore success=1] /lib64/security/pam_succeed_if.so user ingroup [no2fa]
password [default=ignore success=1] /lib64/security/pam_succeed_if.so user ingroup [MFA]
password sufficient pam_sss.so use_authtok
password sufficient /lib64/security/pam_vrsn_otp.so prompt=SecurityCode:
password include system-auth

session optional pam_keyinit.so force revoke
session [default=ignore success=1] /lib64/security/pam_succeed_if.so user ingroup [no2fa]
session [default=ignore success=1] /lib64/security/pam_succeed_if.so user ingroup [MFA]
session sufficient pam_sss.so
session sufficient /lib64/security/pam_vrsn_otp.so prompt=SecurityCode:
session include system-auth
session required pam_loginuid.so
session required pam_selinux.so open env_params
```

### Enabling PAM Support for Web Applications

As a first step to enabling MFA, PAM is added to the stack of authentication providers. Proceed with the steps below:

1. Sign in to SAS Viya using an account (for example, sasboot) that is in the SAS Administrators group.
2. On the Assumable Groups prompt, select **Yes** to opt in to all of your assumable groups.
3. Go into SAS Environment Manager by clicking on the **Manage Environment** box or select it from the menu on the upper left of the screen.
4. Click on the wrench icon to go into the configuration area.
5. In the view menu, choose **Definitions** and then scroll down the list of definitions to find **sas.logon.pam**. Click the **New Configuration** button.
6. In the dialog box, move the slider over to enable sign-ins using PAM, change the service name to *sasauth-viya*. This is shown in Figure 3 below. Click **Save**.

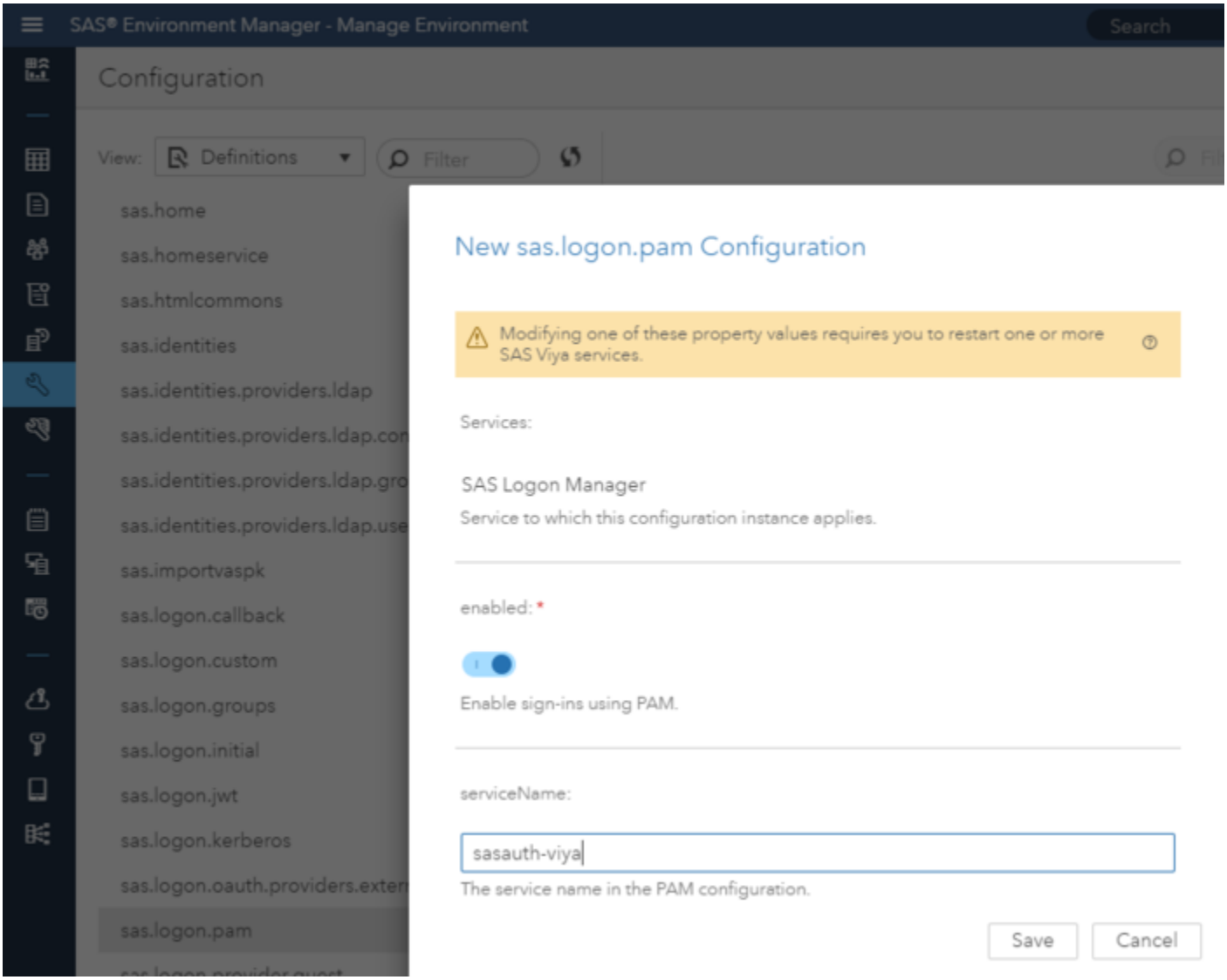


**Figure 3**

## Disabling LDAP Authentication in SAS Logon Manager

The next step involves disabling LDAP authentication. This is necessary since the SAS Logon Manager will attempt LDAP authentication before PAM. If not disabled, authentication through LDAP succeeds, and PAM authentication is skipped and MFA doesn't happen. Note that even with PAM authentication enabled, LDAP is still required in SAS Viya to get identity and group membership information. To disable LDAP for authentication, proceed with the steps below:

1. From the list of definitions in SAS Environment Manager, scroll down the list to find **sas.identities.providers.ldap.connection**. Click on the icon on the right to edit the definition.
2. In the dialog box, edit the list of services, select the SAS Logon Manager and click on the left arrow button to move it out of the selected items over to the available items. Click **OK,** and then save the configuration.
3. Do the same thing for the sas.identities.providers.ldap.user definition.

4. Finally, sign out of SAS Environment Manager and restart SAS Logon Manager from the command line by entering: `service sas-viya-saslogon-default restart`

## CONCLUSION

As demonstrated in this example, it is possible to configure SAS 9.4 and SAS Viya for MFA using underlying system configuration options and those available within SAS itself. The use of PAM makes integration easier and the tighter integration of SAS Viya with PAM decreases the steps required. Since Symantec VIP integrates with PAM, the SAS configuration steps in this paper can be used as a foundation to configure other MFA solutions using PAM with SAS as the integration point for both SAS and the MFA solution is PAM itself.

## REFERENCES


*SAS® 9.4 Intelligence Platform: Middle-Tier Administration Guide Fourth Edition.* 2016. SAS Institute Inc., Cary, NC. Available http://go.documentation.sas.com/?docsetId=bimtag&docsetTarget=titlepage.htm&docsetVersion=9.4&locale=en.

Symantec VIP. Symantec Corporation, Mountain View, CA, Available https://www.symantec.com/products/information-protection/identity/validation-id-protection.

*Symantec VIP Integration Guide for Apache HTTP Server*. 2016. Symantec Corporation, Mountain View, CA. Available https://support.symantec.com/en_US/article.INFO3327.html

*Symantec VIP Integration Guide for Pluggable Authentication Modules*. 2016. Symantec Corporation, Mountain View, CA. Available https://support.symantec.com/en_US/article.INFO3322.html.

"Using Apache httpd 2.4 on Red Hat Enterprise Linux 6." 2014. Red Hat, Inc. Available https://developers.redhat.com/blog/2014/10/01/using-apache-httpd-2-4-rhel6/.


## ACKNOWLEDGMENTS


The domain experts mentioned in this section made this example possible. Their insight into the inner workings of SAS and Symantec were invaluable during the creation process for this example.

Chuck Hunley, Principal Software Developer, SAS Research and Development

Stuart Rogers, Architecture and Security Lead, SAS Global PSD Enablement and Learning

Brian Wilson, Senior Manager, SAS IT Information Security


## CONTACT INFORMATION

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


Jody Steadman
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
jody.steadman@sas.com
http://www.sas.com

Mike Roda
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
mike.roda@sas.com
http://www.sas.com