SafeNet Authentication Service

Integration Guide

SAS Using RADIUS Protocol with Apache HTTP Server

Document Information

Document Part Number	007-012944-001, Rev. A
Release Date	February 2015

Trademarks

Disclaimer

SafeNet makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, SafeNet reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon SafeNet to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

SafeNet invites constructive comments on the contents of this document. These comments, together with your personal and/or company details, should be sent to the address or email below.

Contact Method	Contact Information
Mail	SafeNet, Inc. 4690 Millennium Drive Belcamp, Maryland 21017, USA
Email	TechPubs@safenet-inc.com

Contents

Third-Party Software Acknowledgement

This document is intended to help users of SafeNet products when working with third-party software, such as Apache HTTP Server.

Material from third-party software is being used solely for the purpose of making instructions clear. Screen images and content obtained from third-party software will be acknowledged as such.

Description

SafeNet Authentication Service delivers a fully automated, versatile, and strong authentication-as-a-service solution.

With no infrastructure required, SafeNet Authentication Service provides smooth management processes and highly flexible security policies, token choice, and integration APIs.

Apache HTTP Server, also referred to as Apache, is the world's most widely-used web server software. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Most commonly used on a UNIX-like system, the software is available for a wide variety of operating systems, including UNIX, FreeBSD, Linux, Solaris, Novell NetWare, OS X, Microsoft Windows, OS/2, TPF, OpenVMS, and eComStation.

This document describes how to:

- Deploy multi-factor authentication (MFA) options in Apache HTTP Server using SafeNet OTP authenticators managed by SafeNet Authentication Service.
- Configure Apache HTTP Server to work with SafeNet Authentication Service in RADIUS mode.

It is assumed that the website is already hosted on Apache HTTP Server and is running prior to implementing multi-factor authentication using SafeNet Authentication Service.

Apache HTTP Server can be configured to support multi-factor authentication in several modes. The RADIUS protocol will be used for the purpose of working with SafeNet Authentication Service.

Applicability

The information in this document applies to:

- **SafeNet Authentication Service (SAS)**—SafeNet's cloud-based authentication service.
- **SafeNet Authentication Service – Service Provider Edition (SAS-SPE)**—A server version that is used by Service Providers to deploy instances of SafeNet Authentication Service.
- **SafeNet Authentication Service – Private Cloud Edition (SAS-PCE)**—A server version that is used to deploy the solution on-premises in the organization.

Environment

The integration environment that was used in this document is based on the following software versions:

- **SafeNet Authentication Service – Private Cloud Edition (SAS-PCE)**
- **Apache HTTP Server 2.2.15 running on RedHat 6.3 (32-bit)**

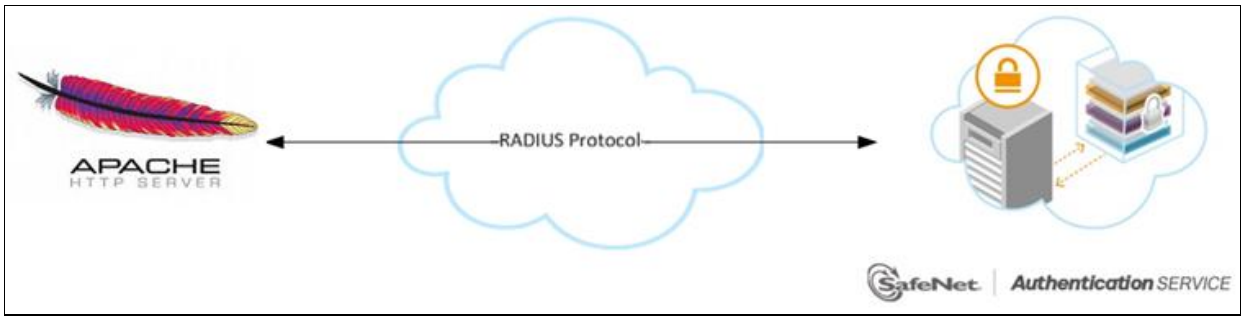
Audience

This document is targeted to system administrators who are familiar with Apache HTTP Server and are interested in adding multi-factor authentication capabilities using SafeNet Authentication Service.

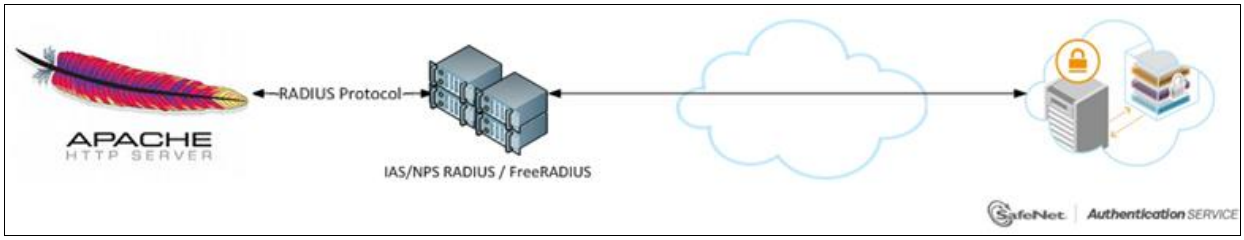
RADIUS-based Authentication using SAS Cloud

SAS Cloud provides two RADIUS mode topologies:

- **SAS Cloud hosted RADIUS service** – A RADIUS service that is already implemented in the SAS cloud environment and can be used without any installation or configuration requirements.



- **Local RADIUS hosted on-premises** - A RADIUS agent that is implemented in the existing customer's RADIUS environment. The agent forwards the RADIUS authentication requests to the SAS Cloud environment. The RADIUS agent can be implemented on a Microsoft NPS/IAS or FreeRADIUS server.



For more information on how to install and configure SAS Agent for IAS/NPS, refer to:
http://www2.safenet-inc.com/sas/implementation-guides/sfnt-updates/SAS-Agents-IASNPS.pdf

For more details on how to install and configure FreeRADIUS, refer to the *SAS FreeRADIUS Agent Configuration Guide*.

This document demonstrates the solution using the SAS Cloud hosted RADIUS service.

RADIUS-based Authentication using SAS-SPE and SAS-PCE

In addition to the pure cloud-based offering, SafeNet Authentication Service comes with two on-premises versions:

- **SafeNet Authentication Service – Service Provider Edition (SPE)** – An on-premises version of SafeNet Authentication Service targeted at Service Providers interested in hosting SAS in their data center.
- **SafeNet Authentication Service – Private Cloud Edition (PCE)** – An on-premises version of SafeNet Authentication Service targeted at organizations interested in hosting SAS in their private cloud environment.

For both on-premises versions, SAS can be integrated with the following solutions that serve as local RADIUS servers:

- **Microsoft Network Policy Server (MS-NPS) or the legacy Microsoft Internet Authentication Service (MS-IAS)**—SafeNet Authentication Service is integrated with the local RADIUS servers using a special on-premises agent called SAS Agent for Microsoft IAS and NPS.

  For more information on how to install and configure the SAS Agent for Microsoft IAS and NPS, refer to the following document:

  http://www2.safenet-inc.com/sas/implementation-guides/sfnt-updates/SAS-Agents-IASNPS.pdf

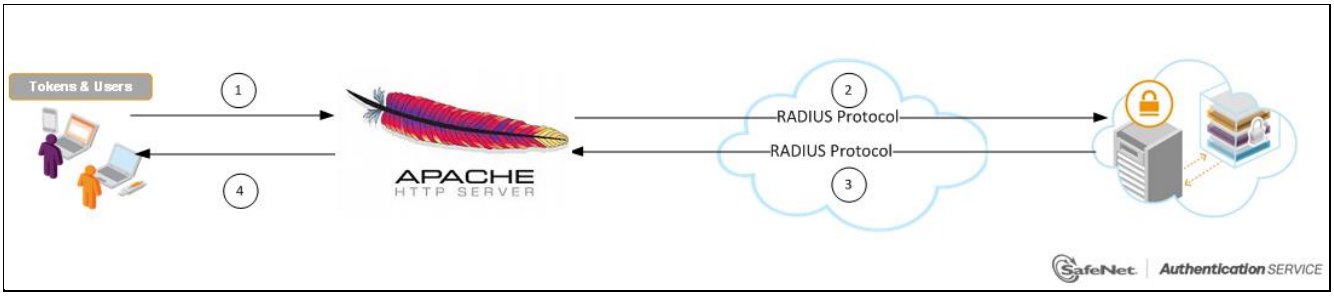
- **FreeRADIUS**—The SAS FreeRADIUS Agent is a strong authentication agent that is able to communicate with SAS through the RADIUS protocol.

  For more information on how to install and configure the SAS FreeRADIUS Agent, refer to the SafeNet Support Portal.

RADIUS Authentication Dataflow using SAS

SafeNet Authentication Service communicates with a large number of VPN and access-gateway solutions using the RADIUS protocol.

The image below describes the data flow of a multi-factor authentication transaction for Apache HTTP Server.



1. A user attempts to access a website hosted on Apache HTTP Server using an OTP authenticator.
2. Apache HTTP Server sends a RADIUS request with the user's credentials to SafeNet Authentication Service for validation.
3. The SAS authentication reply is sent back to Apache HTTP Server.
4. The user is granted or denied access to the website hosted on Apache HTTP Server based on the OTP value calculation results from SAS.

RADIUS Prerequisites

To enable SafeNet Authentication Service to receive RADIUS requests from Apache HTTP Server, ensure the following:

- End users can access the website hosted on Apache HTTP Server before configuring Apache HTTP Server to use RADIUS authentication.
- Ports 1812/1813 are open to and from Apache HTTP Server.
- A shared secret key has been selected, providing an added layer of security by supplying an indirect reference to a shared secret key. It is used by a mutual agreement between the RADIUS server and the RADIUS client for encryption, decryption, and digital signature purposes.

Configuring SafeNet Authentication Service

The deployment of multi-factor authentication using SAS with Apache HTTP Server using RADIUS protocol requires the following:

- Synchronizing Users Store to SafeNet Authentication Service, page 7
- Assigning Authenticators in SAS, page 8
- Adding Apache HTTP Server as an Authentication Node in SAS, page 8
- Checking the SAS RADIUS Address, page 10

Synchronizing Users Store to SafeNet Authentication Service

Before SAS can authenticate any user in your organization, you need to create a user store in SAS that reflects the users that would need to use multi-factor authentication. User records are created in the SAS user store using one of the following methods:

- Manually, one user at a time using the **Create User** shortcut
- Manually, by importing one or more user records via a flat file
- Automatically, by synchronizing with your Active Directory/LDAP server using the SAS Synchronization Agent

For further details on importing users to SafeNet Authentication Service, refer to the *SafeNet Authentication Service Subscriber Account Operator Guide:*

http://www2.safenet-inc.com/sas/implementation-guides/sfnt-updates/SAS-SPE-SubscriberAccountOperatorGuide.pdf

All SafeNet Authentication Service documentation can be found on the SafeNet Knowledge Base site.

Assigning Authenticators in SAS

SAS supports a number of authentication methods that can be used as second authentication factor for users authenticating through Apache HTTP Server.

The following authenticators are supported:

- eToken PASS
- KT-4 Token
- MP-1 Software Token
- MobilePASS

Authenticators can be assigned to users in two ways:

- **Manual provisioning** – Assign an authenticator to users one by one.
- **Provisioning rules** – The administrator can set provisioning rules in SAS so that the rules will be triggered when group memberships and other user attributes change. An authenticator will be assigned automatically to the user.

Refer to “provisioning rules” in the *SafeNet Authentication Service - Subscriber Account Operator Guide* to learn how to provision the different authentication methods to the users in the SafeNet Authentication Service user store.

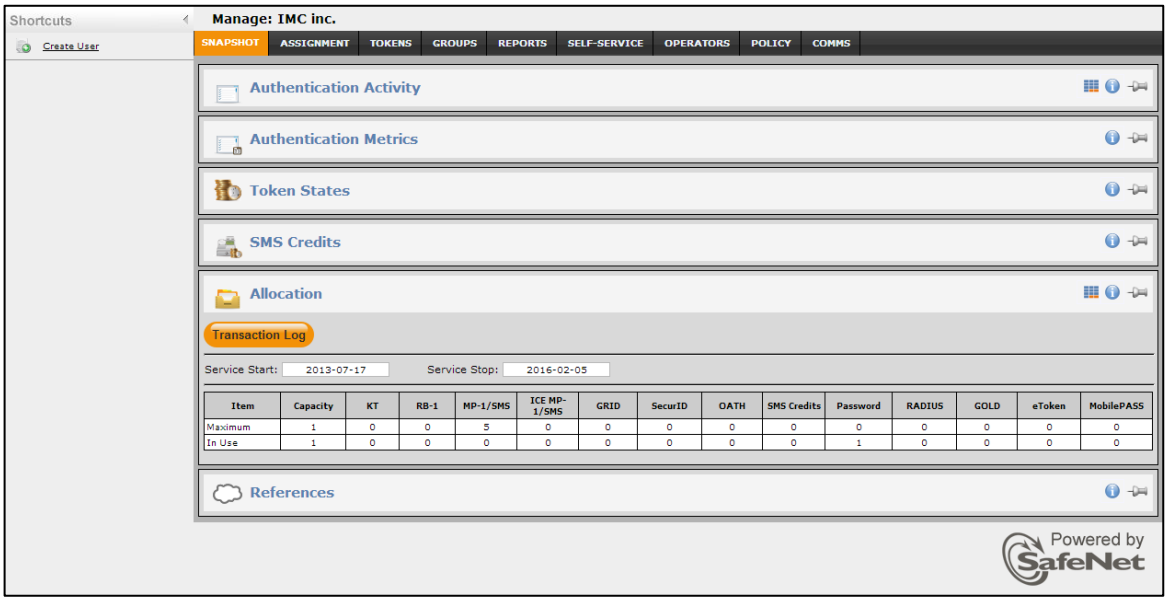
http://www2.safenet-inc.com/sas/implementation-guides/sfnt-updates/SAS-SPE-SubscriberAccountOperatorGuide.pdf

Adding Apache HTTP Server as an Authentication Node in SAS

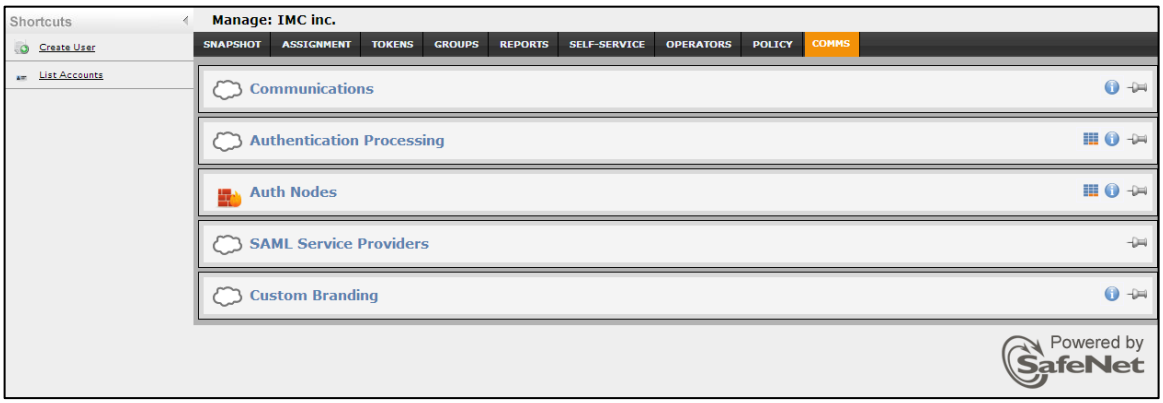
Add a RADIUS entry in the SAS **Authentication Nodes** module to prepare it to receive RADIUS authentication requests from Apache HTTP Server. You will need the IP address of Apache HTTP Server and the shared secret to be used by both SAS and Apache HTTP Server.

To add an Authentication Node in SAS:

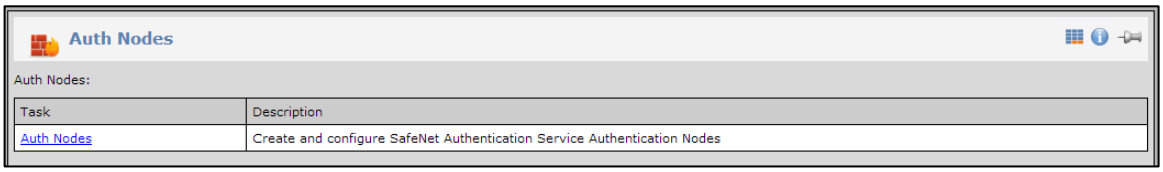
1. Log in to the SAS console with an Operator account.



2. Click the **COMMS** tab, and then select the **Auth Nodes** module.



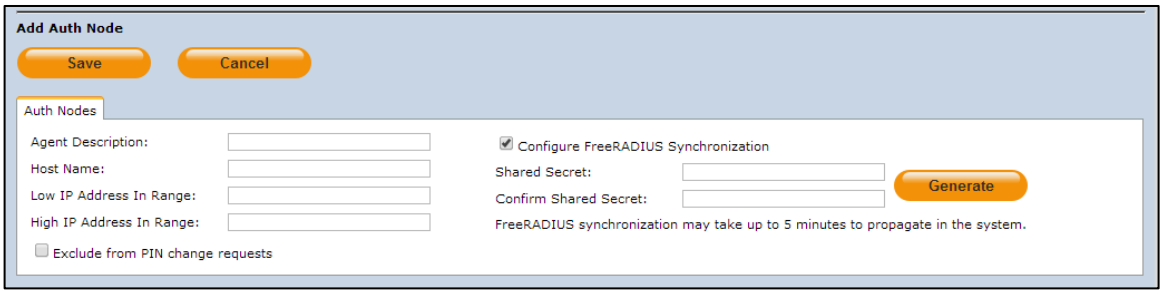
3. In the **Auth Nodes** module, click the **Auth Nodes** link.



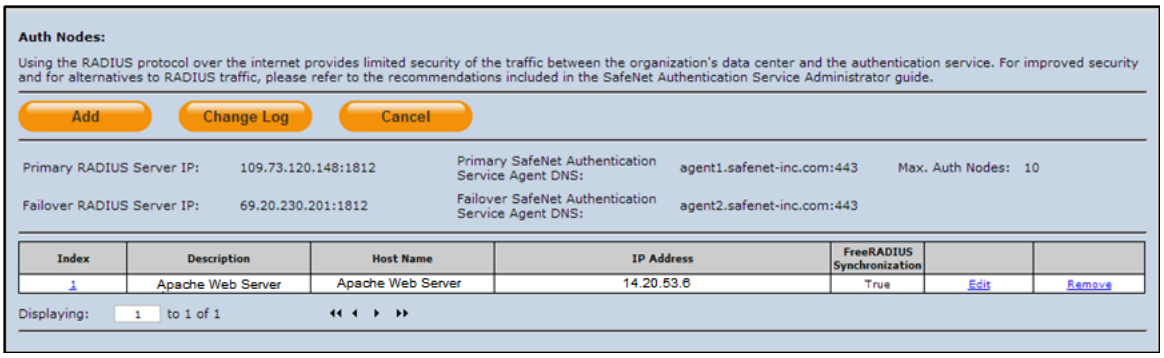
4. Click **Add**.

5. In the **Add Auth Nodes** section, complete the following fields, and then click **Save**:

Agent Description	Enter a host description.
Host Name	Enter the name of the host that will authenticate with SAS.
Low IP Address In Range	Enter the IP address of the host that will authenticate with SAS.
Configure FreeRADIUS Synchronization	Select this option.
Shared Secret	Enter the shared secret key.
Confirm Shared Secret	Re-enter the shared secret key entered above to confirm it.



The Auth Node is added to the system.

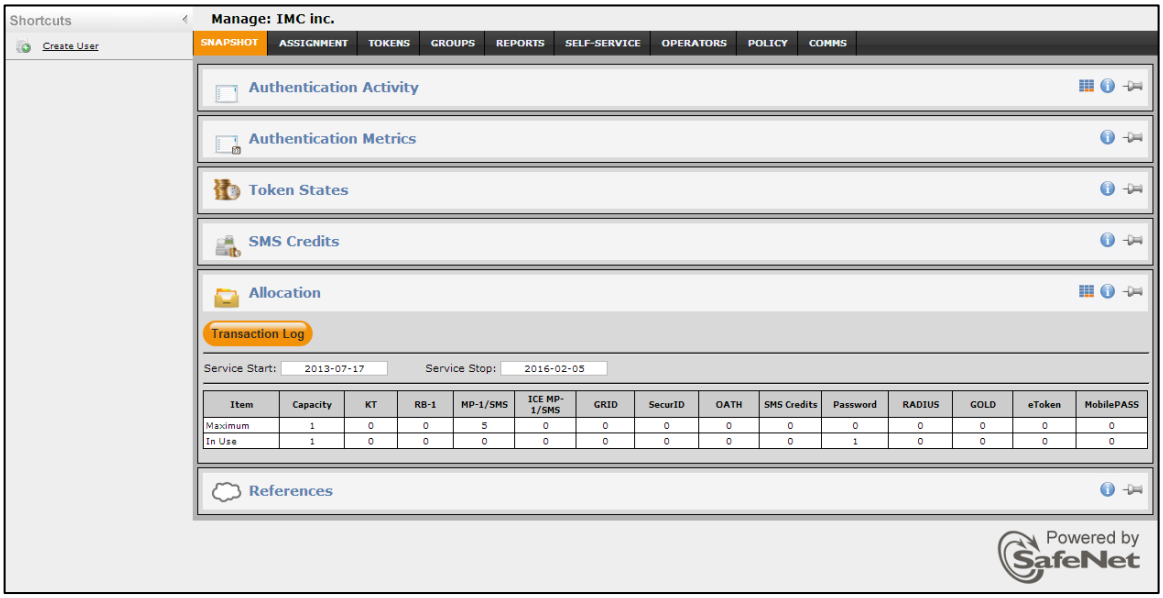


Checking the SAS RADIUS Address

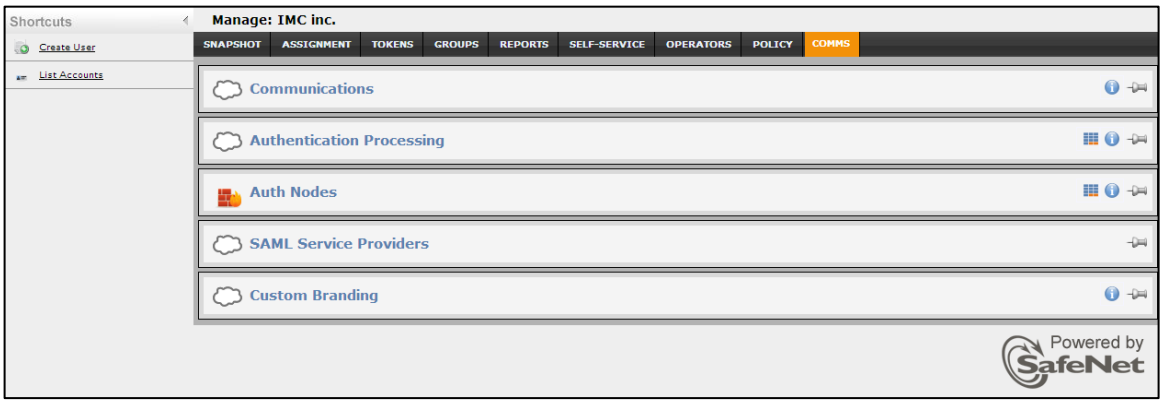
Before adding SafeNet Authentication Service as a RADIUS server in Apache HTTP Server, check the IP address of the SAS RADIUS server. The IP address will then be added to Apache HTTP Server as a RADIUS server at a later stage.

To check the IP address of the SAS RADIUS server:

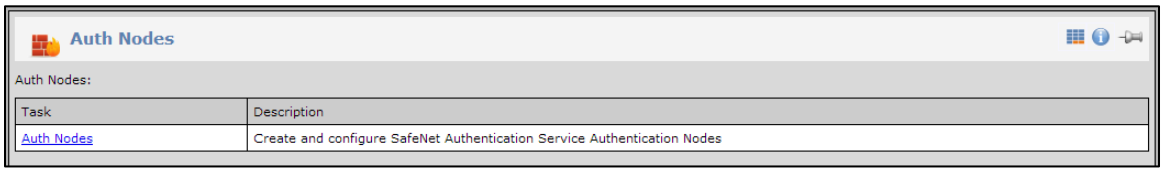
1. Log in to the SAS console with an Operator account.



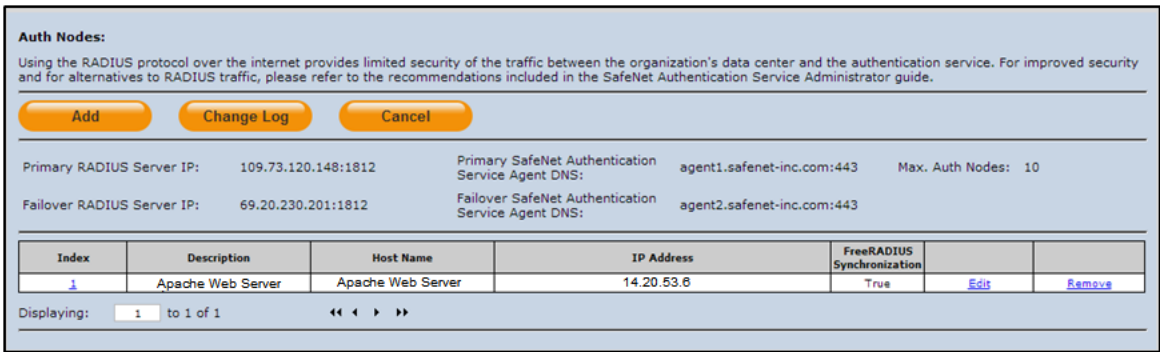
2. Click the **COMMS** tab, and then select the **Auth Nodes** module.



3. Click the **Auth Nodes** link.



The SAS RADIUS server details are displayed.



Configuring Apache HTTP Server

Configure Apache HTTP Server for RADIUS authentication with SafeNet Authentication Service. To achieve this, perform the following:

- Installing the mod_auth_radius Module, page 12
- Configuring Apache for RADIUS Authentication, page 12

NOTE: To perform all the operations below, log in as a root user in Linux.

Installing the mod_auth_radius Module

Install the **mod_auth_radius** module, which is the Apache RADIUS authentication module. It allows any Apache HTTP Server to become a RADIUS client for authentication and accounting requests.

To install the mod_auth_radius module:

1. Download the **mod_auth_radius-1.5.8.tar** package from the following site: http://freeradius.org/mod_auth_radius/
2. Extract the package to a temporary location.
3. Browse to the directory **/usr/sbin** and execute the following command:

   apxs –i –a –c <*location*>/mod_auth_radius.2.0.c

   where, <*location*> is the path to the extracted **mod_auth_radius** package.

> **NOTE:** If **apxs** is not present, install the **httpd-devel** package.
>
> On 64-bit operating systems, 32-bit libraries should be installed, which are used by the **mod_auth_radius** module.

   If the module is installed successfully, the following message is displayed:

   [activating module 'radius_auth' in /etc/httpd/conf/httpd.conf]

Configuring Apache for RADIUS Authentication

Configure Apache HTTP Server so that it acts as a RADIUS client and forwards the authentication request to SafeNet Authentication Service.

1. Open the **httpd.conf** file located at **/etc/httpd/conf**.
2. At the end of the **httpd.conf** file, add the following section, and then save it:

```
alias /home "/var/www/html/home.html"
<IfModule radius_auth_module>
AddRadiusAuth 109.73.120.148:1812       1111    5
AddRadiusCookieValid    20
</IfModule>

<LocationMatch "/home">
        Order Allow,Deny
        AuthType        Basic
        Require valid-user
        Satisfy         any
        AuthName        "SafeNet Authentication"
        AuthBasicProvider radius
        AuthBasicAuthoritative off
```

AuthRadiusAuthoritative on

AuthRadiusCookieValid 15

AuthRadiusActive on

</LocationMatch>

Each of these directives is explained below:

alias /home "/var/www/html/home.html"	**alias /home "/var/www/html/home.html"** creates an alias **home** for the path **/var/www/html/home.html**, where the website is present.
AddRadiusAuth	This directive tells Apache to authenticate against RADIUS. You need to specify the name or IP address of the RADIUS server, port number, shared secret for the web client, and the timeout period Apache should wait before giving up and assuming no response will be sent.
AddRadiusCookieValid	This directive specifies, in minutes, the length of time that the cookie sent in the response to the end user from the web client is valid. Setting this value to zero (0) signifies that the cookie will be valid forever.
AuthType	This module requires basic authentication.
AuthName	The contents of this string are included in the password prompt presented to the user.
AuthBasicAuthoritative	This directive ensures that other authentication types are not used for this particular site area.
AuthRadiusAuthoritative	This directive tells Apache to consider all RADIUS responses authoritative; that is, the RADIUS responses are "the final answer" for authentication.
AuthRadiusCookieValid	This directive specifies, in minutes, the length of time that the cookie sent in the response to the end user from the web client is valid. The server takes the lower of the two values, the **AddRadiusCookieValid** directive set here and the **AddRadiusCookieValid** directive set in the previous step, and sets the cookie to expire at that interval.
AuthRadiusActive	This directive turns on RADIUS authentication globally for the site.
Require valid-user	This directive ensures that only valid users can access the site. All the authentication requests are processed on the RADIUS server.

3. Use the following command to restart the **httpd** service:

   service httpd restart

Running the Solution

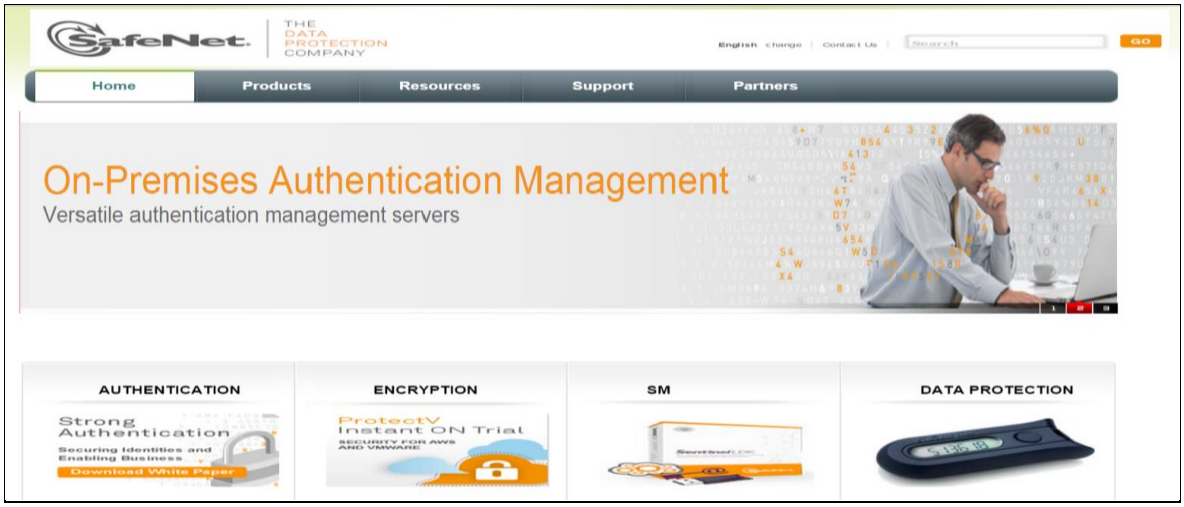
After successfully installing the **mod_auth_radius** module and configuring the Apache HTTP Server for RADIUS Authentication, test the final solution.

For this integration, the SafeNet MP-1 software token is configured for authentication with the SAS solution.

1. In a web browser, open your protected website hosted on the Apache HTTP Server.
2. On the login window, enter your user name.

3. Generate an OTP using the SafeNet MP-1 software token, and enter it in the **Password** field. Click **OK**.

   If the credentials are valid, the user is redirected to the protected website.



Support Contacts

If you encounter a problem while installing, registering, or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, contact your supplier or SafeNet Customer Support. SafeNet Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between SafeNet and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Contact Method	Contact Information	
Address	SafeNet, Inc. 4690 Millennium Drive Belcamp, Maryland 21017 USA	
Phone	United States	1-800-545-6608
	International	1-410-931-7520
Technical Support Customer Portal	https://serviceportal.safenet-inc.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the SafeNet Knowledge Base.	