SafeNet Authentication Service

Integration Guide

SAS Using RADIUS Protocol with Linux PAM Modules

Document Information

Document Part Number	007-012556-001, Rev. B
Release Date	October 2014

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 Linux PAM Modules.

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.

PAM modules can be used in Linux environments to provide an additional level of security with a given service or application that is PAM aware.

This document describes how to:

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

It is assumed that the Linux PAM Modules environment is already configured and working with static passwords prior to implementing multi-factor authentication using SafeNet Authentication Service.

Linux PAM Modules 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 (SAS)** – SafeNet's cloud-based authentication service
- Linux PAM Modules 1.3.16 on RHEL 6.4 and SuSe 11 (64-bit OS)

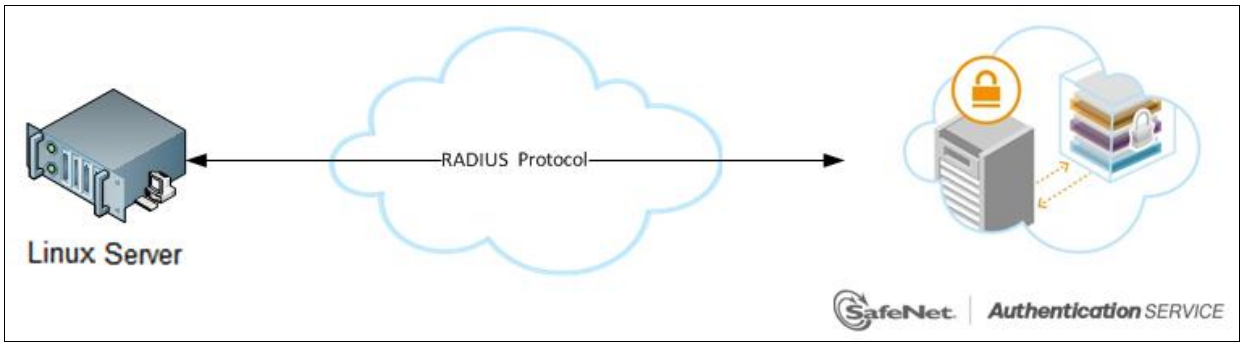
Audience

This document is targeted to system administrators who are familiar with the Linux PAM Modules and are interested in adding multi-factor authentication capabilities using SafeNet Authentication Service.

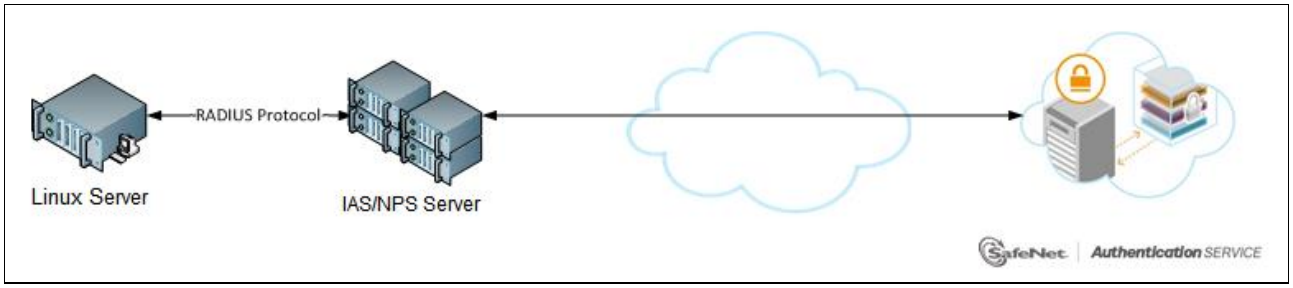
RADIUS-based Authentication using SAS Cloud

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 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

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.

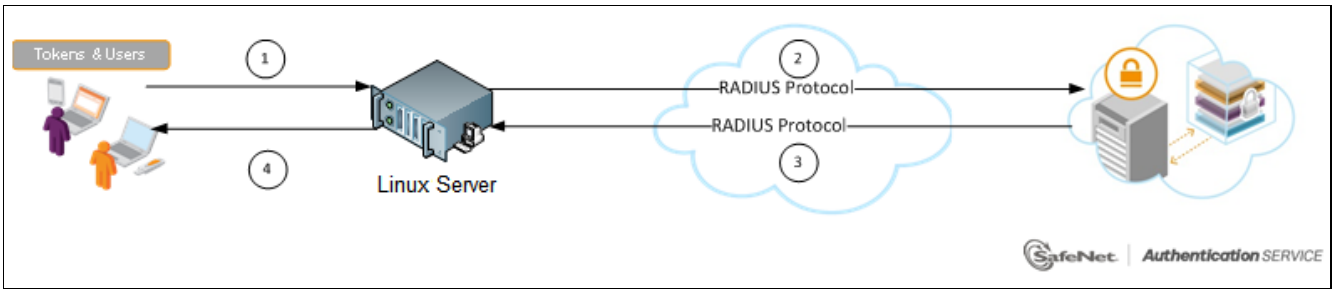
  For more information on how to install and configure the SAS FreeRADIUS Agent, refer to the SafeNet Support Portal.

RADIUS Authentication Flow using SAS

The following sub-sections describe the dataflow of a multi-factor authentication transaction for Linux PAM Modules. The dataflow is described for:

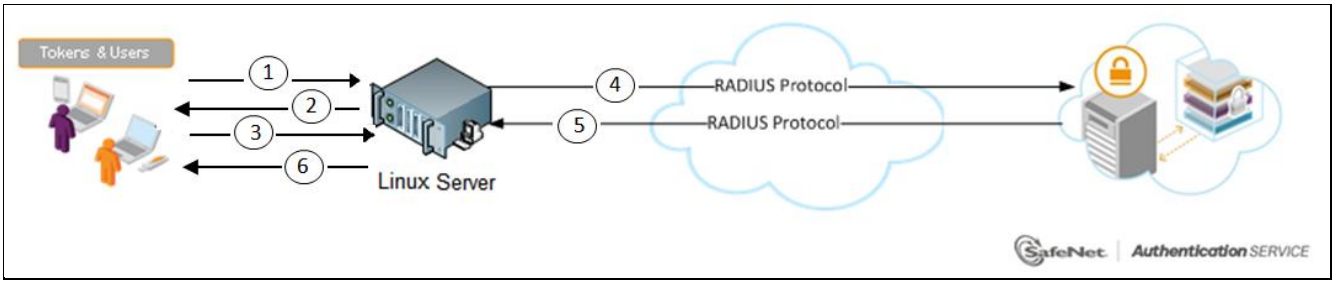
- One-step authentication
- Two-step authentication

Dataflow for One–step Authentication



1. A user attempts to log on to the PAM-aware Linux service using an OTP authenticator.
2. The Linux 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 the Linux server.
4. The user is granted or denied access to the Linux service based on SAS reply.

Data Flow for Two–step Authentication



1. A user attempts to log on to the PAM-aware Linux service using the username/password combination.
2. On successful authentication, the Linux server requests an OTP.
3. The user provides the OTP to the Linux server.
4. The Linux server sends a RADIUS request with the user's credentials (username and OTP) to SafeNet Authentication Service for validation.
5. The SAS authentication reply is sent back to the Linux server.
6. The user is granted or denied access to the Linux service based on the SAS reply.

Prerequisites

To enable SafeNet Authentication Service to receive RADIUS requests from Linux PAM Modules, ensure the following:

- End users can use the PAM-aware Linux services with a static password before configuring the Linux PAM Modules to use RADIUS authentication.
- Ports 1812/1813 are open to and from the Linux 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 RADIUS client for encryption, decryption, and digital signature purposes.
- Download the RADIUS PAM modules from this website: http://freeradius.org/pam_radius_auth/

  (*This URL is a third-party website and could change at any time.)
- Extract the contents of the **tar.gz** file (PAM modules you downloaded) to a directory. Ensure that you have sufficient privileges to read and write to that directory.
- The user name in Linux and the user ID in SAS must be identical.

Configuring SafeNet Authentication Service

The deployment of multi-factor authentication using SAS with Linux PAM Modules using RADIUS protocol requires:

- Synchronizing Users Stores to SAS
- Authenticator Assignment in SAS
- Adding Linux PAM Modules as an Authentication Node in SAS
- Checking the SAS RADIUS IP address

Synchronizing Users Stores 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 section on “creating users” in 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.

Authenticator Assignment in SAS

SAS supports a number of authentication methods that can be used as a second authentication factor for users who are authenticating through Linux PAM Modules.

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.

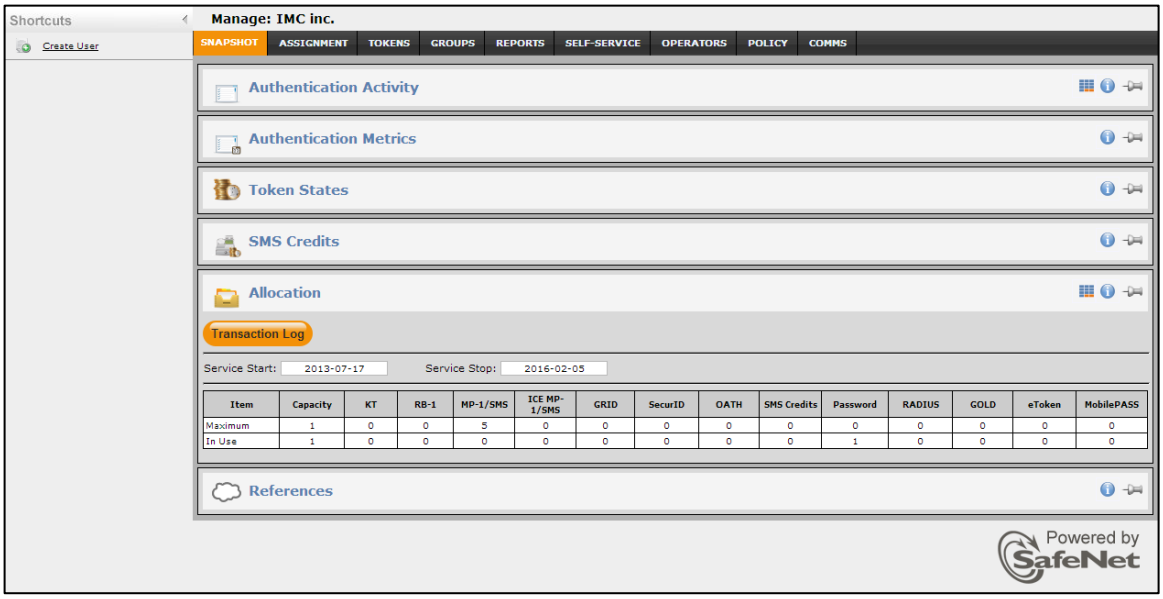
http://www2.safenet-inc.com/sas/implementation-guides/sfnt-updates/SAS-SPE-SubscriberAccountOperatorGuide.pdf

Adding a Linux Server as an Authentication Node in SAS

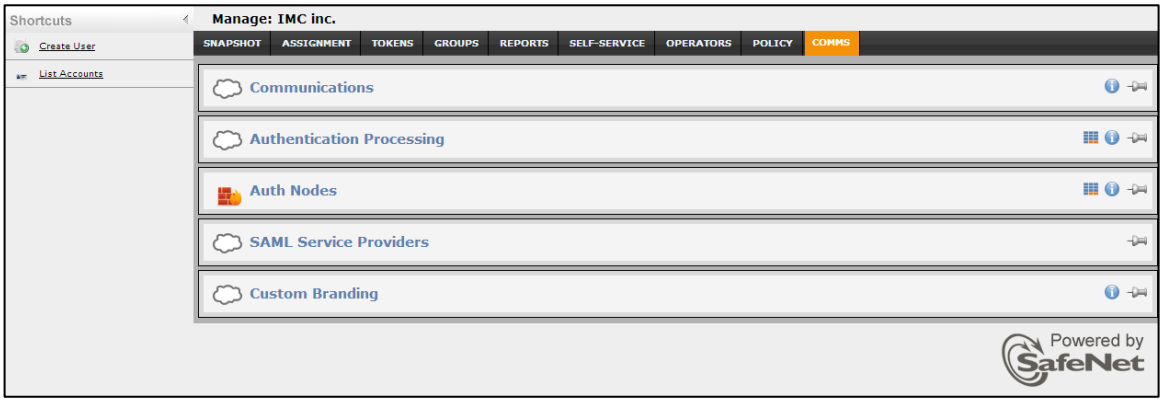
Add a RADIUS entry in the SAS **Authentication Nodes** module to prepare it to receive RADIUS authentication requests from a Linux server. You will need the IP address of the Linux server and the shared secret to be used by both SAS and the Linux server.

To add an Authentication Node in SAS:

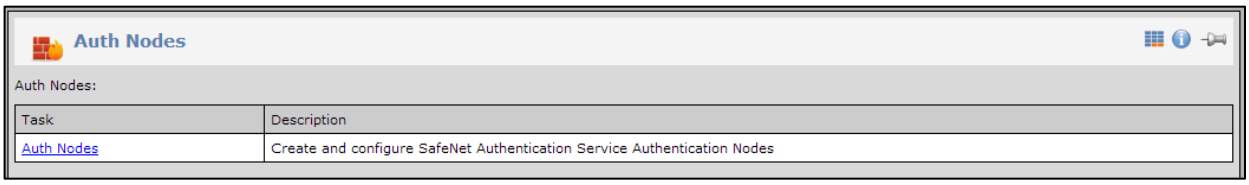
1. Log in to the SAS console with an Operator account.



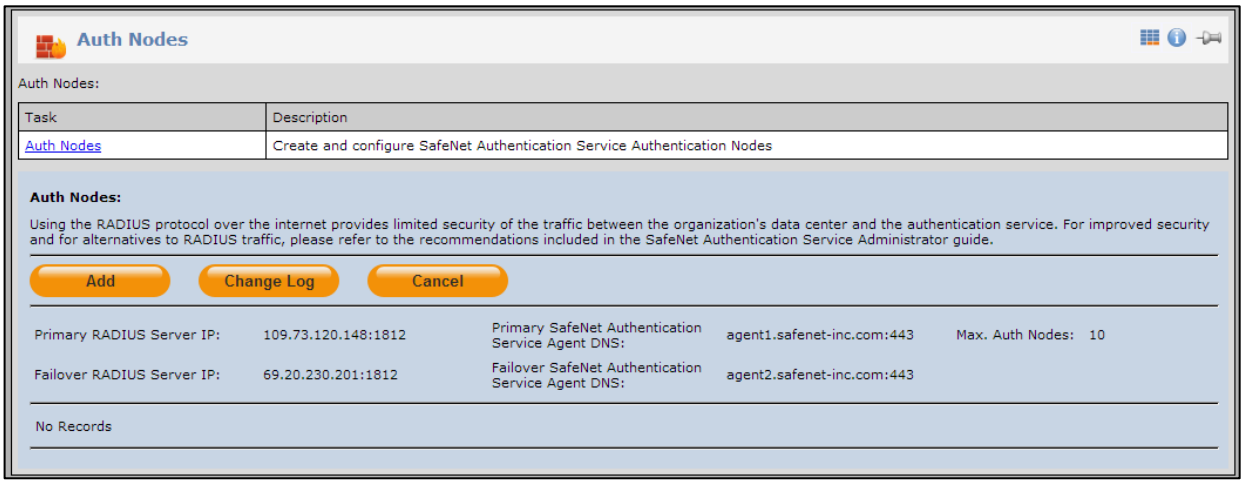
2. Click the **COMMS** tab, and then select the **Auth Nodes** module.



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

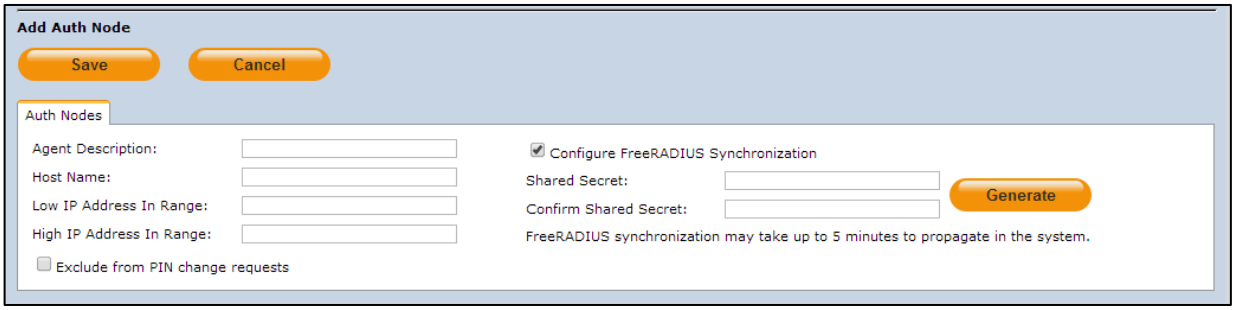


4. Click **Add**.

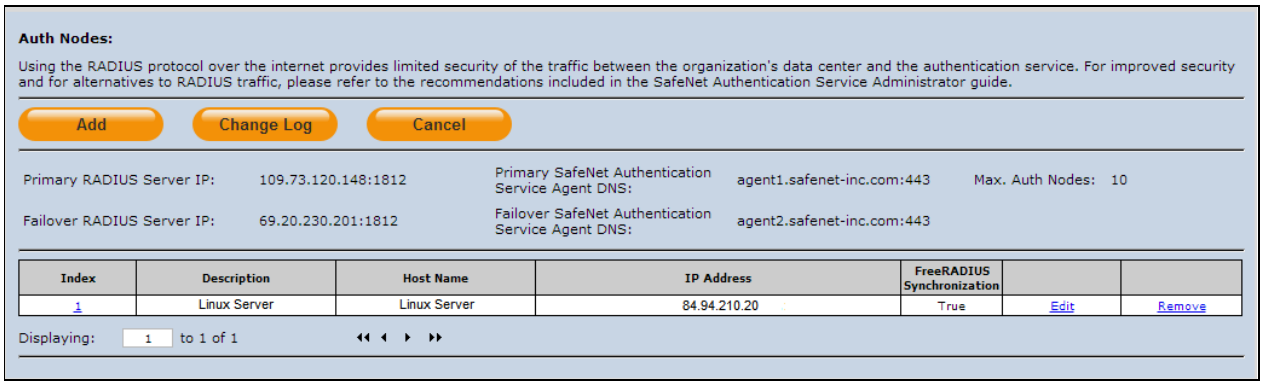


5. In the **Add Auth Nodes** section, complete the following fields, and then click **Save**:

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 to confirm it.



The Auth Node is added to the system.

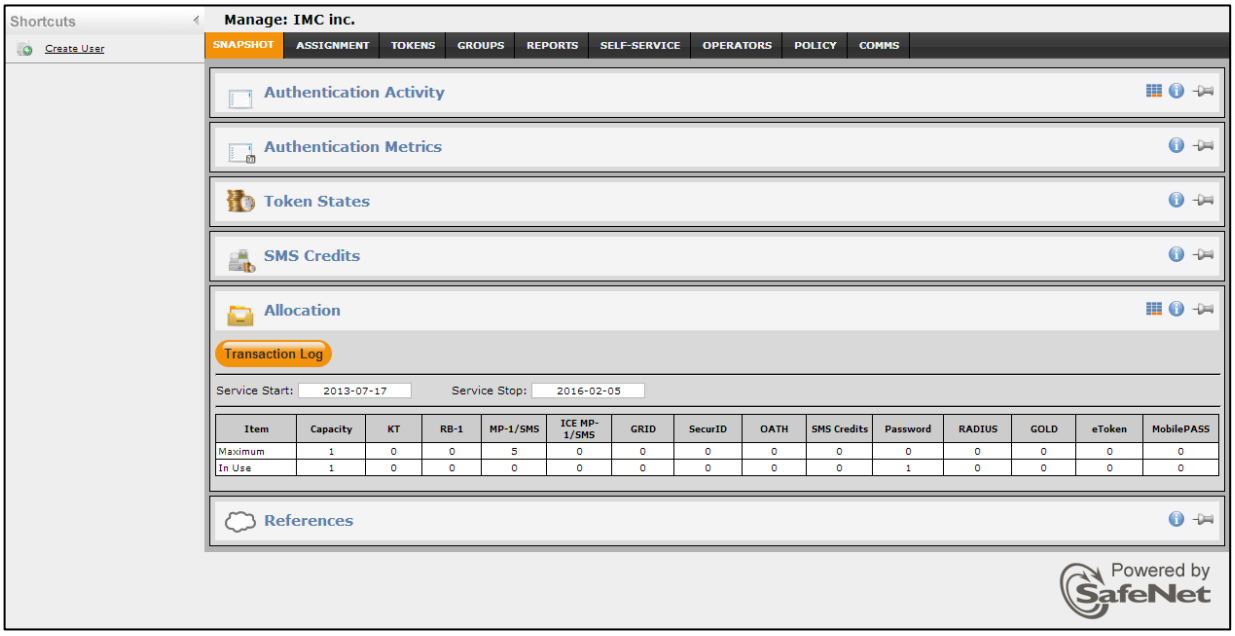


Checking the SAS RADIUS Address

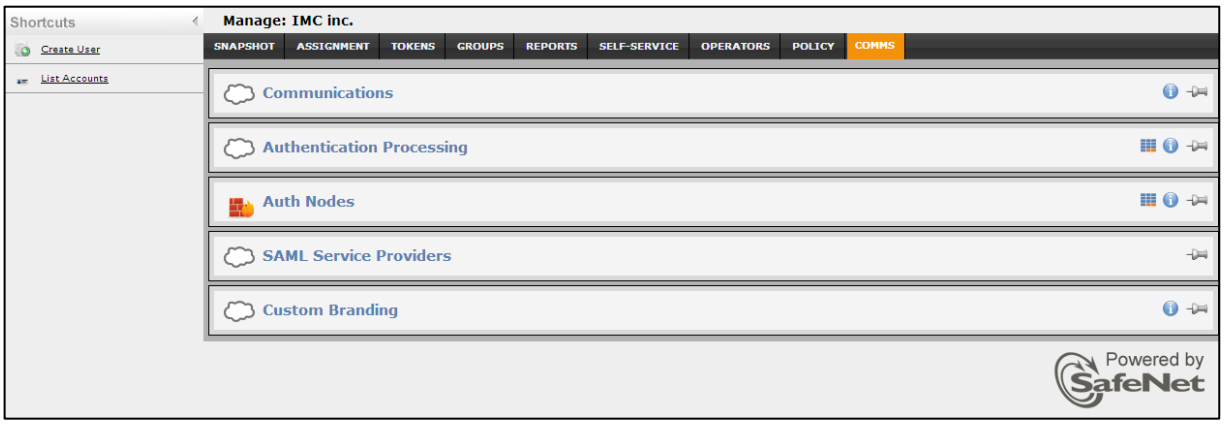
Before adding SafeNet Authentication Service as a RADIUS server on the Linux server, check the IP address of the SAS RADIUS server. The IP address will then be added to the Linux server as a RADIUS server at a later stage.

To check the IP address of the SAS RADIUS server:

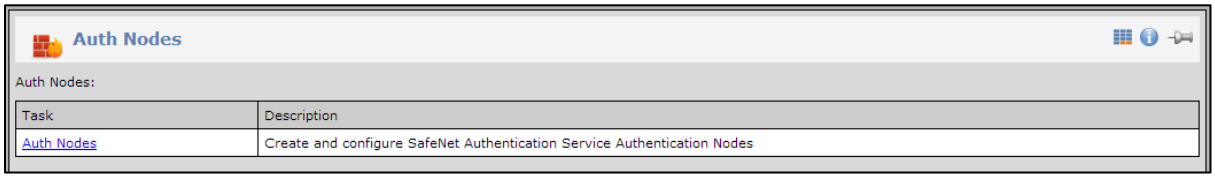
1. Log in to the SAS console with an Operator account.



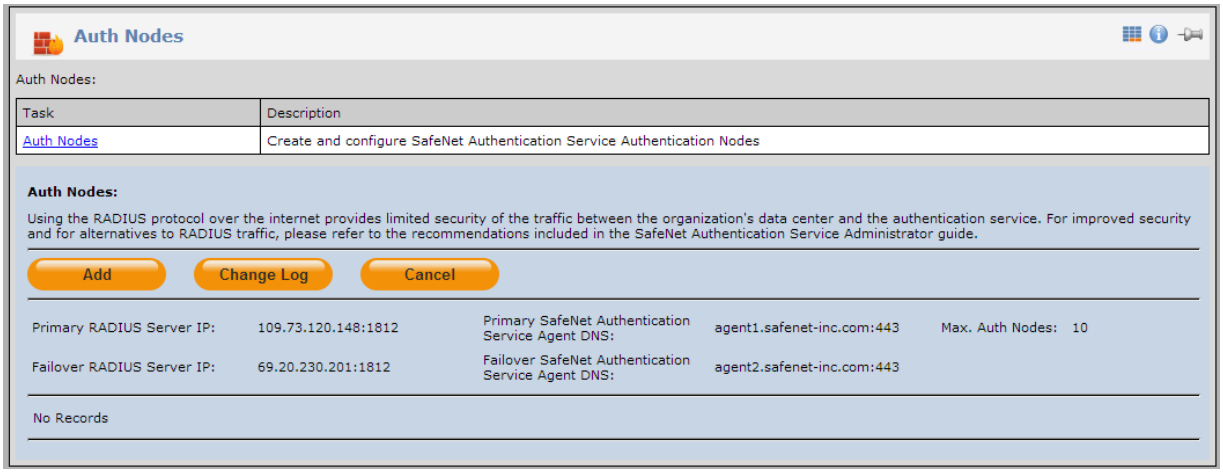
2. Click the **COMMS** tab, and then select the **Auth Nodes** module.



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



The SAS RADIUS server details are displayed.



Configuring Linux PAM Modules

Configure the Linux PAM Modules to use RADIUS authentication for PAM-aware applications. To achieve this, you need to:

- Compile the PAM module
- Add a RADIUS server for the PAM module
- Secure the RADIUS server configuration
- Configure the application-specific configuration files

Compiling the PAM Module

1. Log in as a root user.
2. Go to the directory where the PAM module package is extracted.
3. To compile the PAM module for your system, run the following command:

   make
4. To copy **pam_radius_auth.so** to **/lib/security**, run the following command:

   cp pam_radius_auth.so /lib/security/pam_radius_auth.so

Adding a RADIUS Server for the PAM Module

The FreeRADIUS PAM module searches for the **server** file in the **/etc/raddb** directory. This file contains the location of the RADIUS servers, the shared secret, and the order in which each RADIUS server will be checked.

To add a RADIUS server:

1. Verify that the **/etc/raddb** directory exists. If it does not exist, run the following command to create it:

   mkdir /etc/raddb
2. A generic server configuration file, **pam_radius_auth.conf**, exists in the directory where the PAM module package is extracted. Run the following command to move the **pam_radius_auth.conf** file to the **/etc/raddb** directory and rename it to **server**:

   mv pam_radius_auth.conf /etc/raddb/server

3. Open the **server** file. Below is an example of the **server** file:

```
#  server[:port] secret [timeout]
#
#  the port name or number is optional.  The default port name is
#  "radius", and is looked up from /etc/services The timeout field is
#  optional.  The default timeout is 3 seconds.
#
#  If multiple RADIUS server lines exist, they are tried in order.  The
#  first server to return success or failure causes the module to return
#  success or failure.  Only if a server fails to response is it skipped,
#  and the next server in turn is used.
#
#  The timeout field controls how many seconds the module waits before
#  deciding that the server has failed to respond.
#
# server[:port]         shared_secret      timeout (s)
#127.0.0.1              secret             1
#other-server           other-secret       3
109.73.120.148:1812     1111               5
#
# having localhost in your radius configuration is a Good Thing.
#
# See the INSTALL file for pam.conf hints.
```

4. Enter the RADIUS server details in the following format:

   Server:Port shared_secret timeout

   where:

   - A RADIUS port must be specified in the server file. The default RADIUS port numbers are 1812 or 1645.
   - The **timeout** field controls the time for which the module waits before deciding if the server has failed to respond. This setting is optional.
   - If multiple entries for the RADIUS server exist, they are tried in the order specified. If a server fails to respond, it is skipped and the next server is tried.

Securing the RADIUS Server Configuration

Once the configuration in the **server** file is complete, it must be secured in order to prevent tampering. Run the following commands to secure the **server** file:

chown root /etc/raddb/

chmod -rwx /etc/raddb

chmod -rwx /etc/raddb/server

Configuring the Application-specific Configuration Files

Configure the PAM-aware application you would like to implement. SafeNet only supports the PAM-aware applications listed in the **/etc/pam.d** directory or in the **pam.conf** file. The image below lists all the applications in the **/etc/pam.d** directory.

```
[root@rhel-6 pam.d]# ls /etc/pam.d
atd             fingerprint-auth     kdm           password-auth     setup                    sudo-i                      system-config-network-cmd
authconfig      fingerprint-auth-ac  kdm-np        password-auth-ac  smartcard-auth           su-l                        system-config-users
authconfig-gtk  gdm                  kppp          polkit-1          smartcard-auth-ac        su_min                      xdm
authconfig-tui  gdm~                 kscreensaver  poweroff          smtp                     system-auth                 xdm~
chfn            gdm-autologin        ksu           ppp               smtp.postfix             system-auth~                xserver
chsh            gdm-fingerprint      login         reboot            sshd                     system-auth-ac
config-util     gdm-password         login~        remote            ssh-keycat               system-config-authentication
crond           gdm-smartcard        newrole       rhn_register      su                       system-config-date
cups            gnome-screensaver    opcontrol     run_init          subscription-manager     system-config-kdump
cvs             halt                 other         runuser           subscription-manager-gui system-config-keyboard
eject           kcheckpass           passwd        runuser-l         sudo                     system-config-network
[root@rhel-6 pam.d]#
```

PAM-aware applications (su, halt, reboot, etc.) that require root authentication should not use SAS authentication.

In the sub-sections below, the configuration required for some of the PAM-aware applications is discussed.

NOTE: To avoid loss of any configuration, you must take back up of the existing PAM aware file (for example, login, telnet, and gdm) before deleting its content to test SAS authentication.

Configuration for One-step Authentication

For one-step authentication, the configuration is explained for the following PAM-aware applications:

- login
- telnet
- sshd
- gdm

login

The **login** PAM file affects the local console login sessions. To enable SAS authentication, delete the existing content of the **login** PAM file and then add the following content:

```
#%PAM-1.0
auth        required    /lib/security/pam_radius_auth.so
account     required    /lib/security/pam_permit.so
account     required    /lib/security/pam_unix.so
password    required    /lib/security/pam_unix.so
session     required    /lib/security/pam_console.so
session     required    /lib/security/pam_unix.so
```

telnet

The **telnet** service can be authenticated using SAS. The **login** PAM file affects telnet sessions for SUSE Linux and the **remote** PAM file affects telnet sessions for RedHat Linux.

To enable SAS authentication, delete the existing content of the **login** PAM file and then add the following content:

```
#%PAM-1.0
auth        required    /lib/security/pam_radius_auth.so
account     required    /lib/security/pam_permit.so
account     required    /lib/security/pam_unix.so
password    required    /lib/security/pam_unix.so
session     required    /lib/security/pam_unix.so
```

sshd (OpenSSH)

For security reasons and compatibility with the FreeRADIUS PAM module, you must have at least SSHD version 2.4 for F-Secure or SSH2 version 2.9 for OpenSSH.

SAS will provide support only for versions of OpenSSH/OpenSSL included with RedHat or any updates provided by RedHat.

To enable SAS authentication, delete the existing content of the **sshd** PAM file and then add the following content:

```
#%PAM-1.0
auth        required    /lib/security/pam_radius_auth.so
account     required    /lib/security/pam_unix.so
password    required    /lib/security/pam_unix.so
session     required    /lib/security/pam_unix.so
account     required    /lib/security/pam_access.so
session     required    /lib/security/pam_console.so
```

The following configuration must be done in the **sshd_config** file, which is located under the **etc/ssh/** directory:

```
PasswordAuthentication          yes
PermitEmptyPasswords            no
ChallengeResponseAuthentication no
UsePrivilegeSeparation          no
UsePAM                          yes
```

gdm (graphical desktop logon)

To enable SAS authentication, delete the existing content of the **gdm-password** PAM file and then add the following content:

```
#%PAM-1.0
auth        required    /lib/security/pam_radius_auth.so
auth        required    /lib/security/pam_nologin.so
account     required    /lib/security/pam_access.so
account     required    /lib/security/pam_unix.so
password    required    /lib/security/pam_unix.so
session     required    /lib/security/pam_unix.so
```

NOTE:

- For some operating systems, the **gdm** file may be used instead of the **gdm-password** file.
- SAS does not support challenge-response with graphical logon.
- To globally enable a graphical logon on startup, edit the **/etc/ inittab** file. Change **id:3:initdefault:** to **id:5:initdefault:**

Configuration for Two-step Authentication

PAM-aware applications can also be configured for two-step authentication. The end user will have to enter both the Linux password and the one-time password (OTP).

Two-step authentication can be configured to work in two ways:

- The user first enters the Linux password and then the OTP.
- The user first enters the OTP and then the Linux password.

For two-step authentication, the configuration is explained for the following PAM-aware applications:

- sshd
- gdm

sshd

If you want the user to enter the Linux password first:

To enable SAS authentication, delete the existing content of the **sshd** PAM file and then add the following content:

```
#%PAM-1.0
auth        requisite   /lib/security/pam_unix.so          not_set_pass
auth        required    /lib/security/pam_radius_auth.so
account     required    /lib/security/pam_unix.so
password    required    /lib/security/pam_unix.so
```

```
session     required    /lib/security/pam_unix.so
account     required    /lib/security/pam_access.so
session     required    /lib/security/pam_console.so
```

If you want the user to enter the OTP first:

To enable SAS authentication, delete the existing content of the **sshd** PAM file and then add the following content:

```
#%PAM-1.0
auth        requisite   /lib/security/pam_radius_auth.so
auth        required    /lib/security/pam_unix.so
password    required    /lib/security/pam_unix.so
session     required    /lib/security/pam_unix.so
account     required    /lib/security/pam_access.so
session     required    /lib/security/pam_console.so
```

The following configuration must be done in the **sshd_config** file, which is located in the **etc/ssh/** directory:

```
PasswordAuthentication          no
PermitEmptyPasswords            no
ChallengeResponseAuthentication yes
UsePrivilegeSeparation          no
UsePAM                          yes
```

gdm

If you want the user to enter the Linux password first:

To enable SAS authentication, delete the existing content of the **gdm-password** PAM file and then add the following content:

```
#%PAM-1.0
auth        requisite   /lib/security/pam_unix.so          not_set_pass
auth        required    /lib/security/pam_radius_auth.so
account     required    /lib/security/pam_access.so
account     required    /lib/security/pam_unix.so
password    required    /lib/security/pam_unix.so
session     required    /lib/security/ pam_unix.so
```

If you want the user to enter the OTP first:

To enable SAS authentication, delete the existing content of the **gdm-password** PAM file and then add the following content:

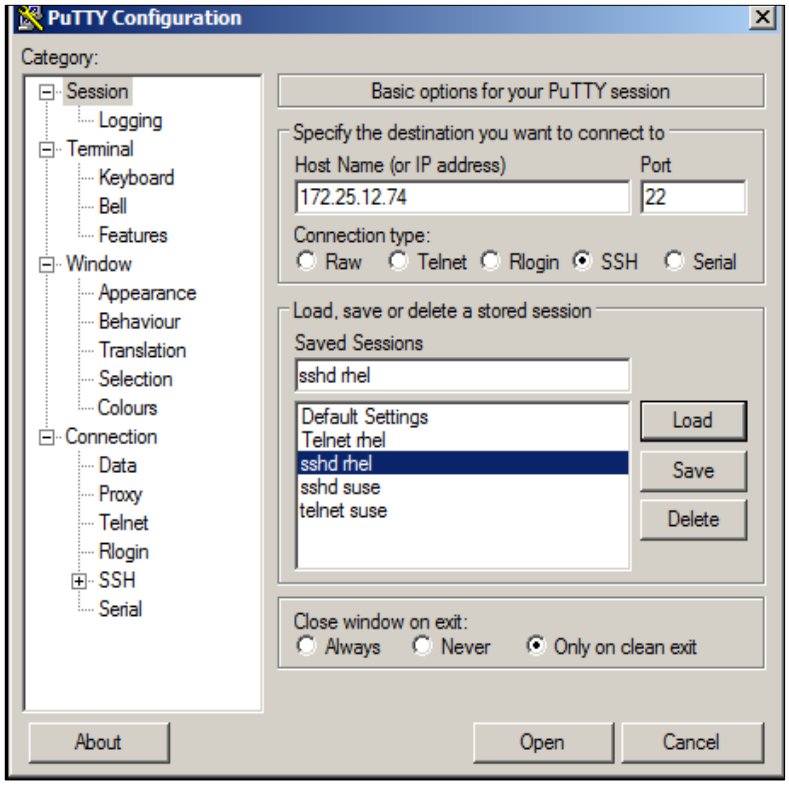
```
#%PAM-1.0
auth        requisite     /lib/security/pam_radius_auth.so
auth        required      /lib/security/pam_unix.so
account     required      /lib/security/pam_access.so
account     required      /lib/security/pam_unix.so
password    required      /lib/security/pam_unix.so
session     required      /lib/security/pam_unix.so
```

Running the Solution

After configuring the PAM aware Linux services and the RADIUS server, the user is ready to authenticate to these services with the help of SafeNet OTP authenticator instead of the static password. The Linux user requiring authentication should be added in SAS and a token should be provisioned.

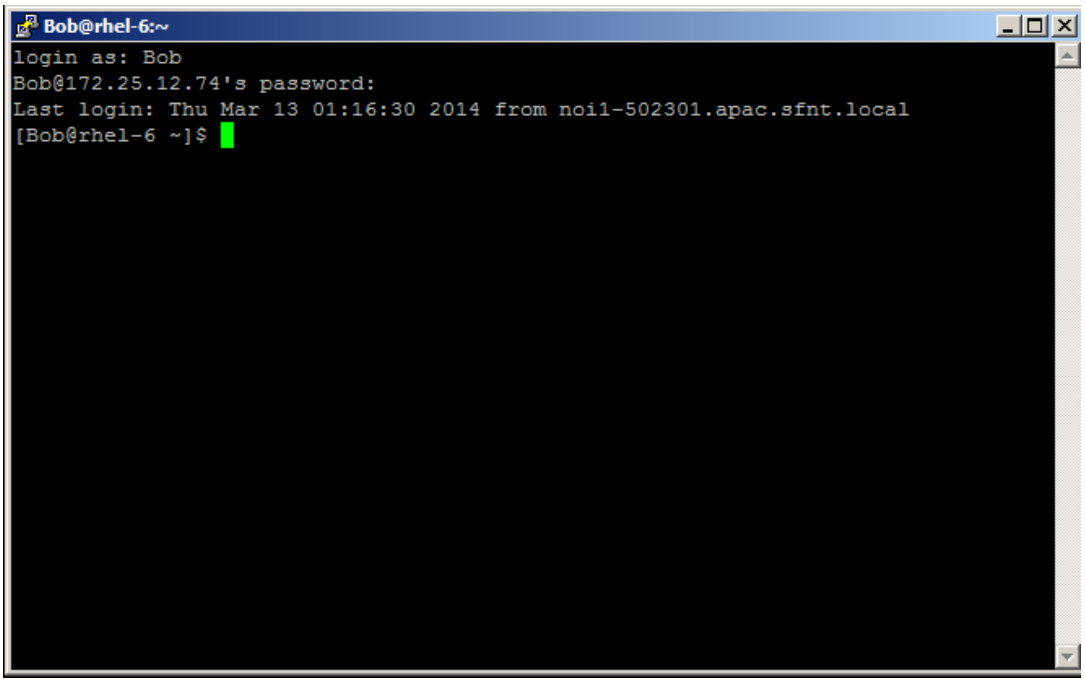
Running the SSHD Service: One-step Authentication

1. Open PuTTY.
2. In the **Host Name** box, enter the IP address of the Linux server.
3. For **Connection Type**, select **SSH** and then click **Open**.



(The screen image above is from PuTTY®. Trademarks are the property of their respective owners.)

4. On the SSH login window, enter the user name for which the token is provisioned.



(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)

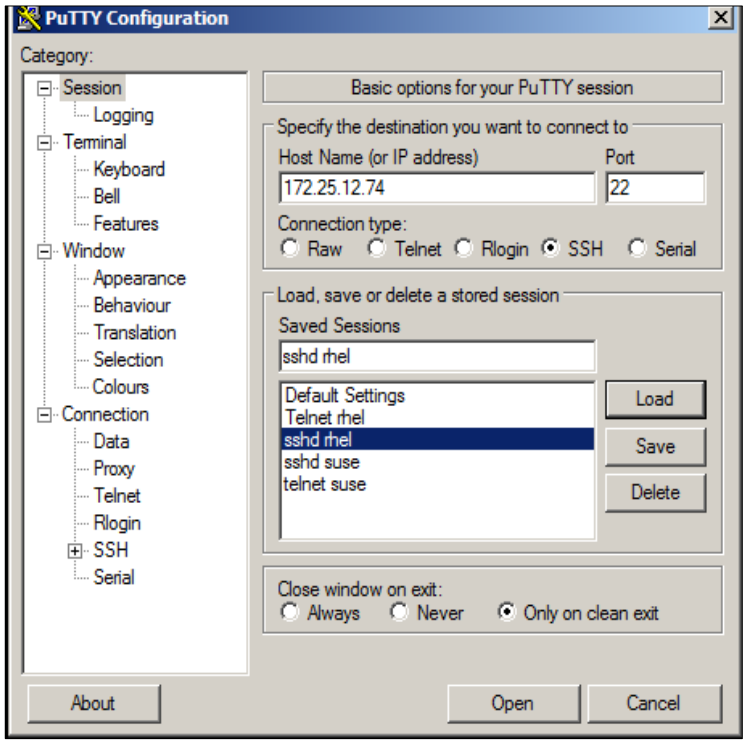
5. Generate a one-time password.
6. Enter the one-time password as the user's password and then press **Enter**.

   The user is successfully logged in.

Running the SSHD Service: Two-step Authentication

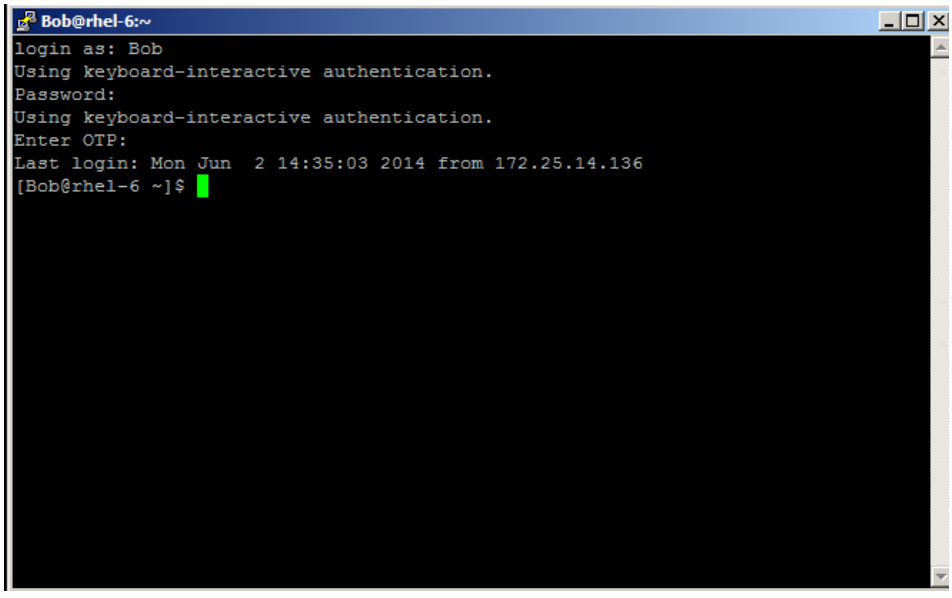
In this solution, the user would enter the password first and then the OTP.

1. Open PuTTY.
2. In the **Host Name** box, enter the IP address of the Linux server.
3. For **Connection Type**, select **SSH** and then click **Open**.



(The screen image above is from PuTTY®. Trademarks are the property of their respective owners.)

4. On the SSH login window, enter the **user name** for which the token is provisioned.



(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)

5. Enter the **Password** and then press **Enter**.
6. Generate a one-time password.
7. Enter the one-time password and then press **Enter**.

   The user is successfully logged in.

Appendix: Changing a Label in the PAM Module

With two-step authentication, when the system prompts for a one-time password (OTP), a **Password** field is displayed for entry of the OTP. You can change the text/label for the **Password** field if desired.

To change the Password field text/label:

1. Go to the directory where the PAM module package is extracted.
2. Open the **pam_radius_auth.c** file in a text editor.
3. Go to line 1152 (only for PAM version 1.3.17) and change the text "**Password:** " with the text of your choice; for example, "**Enter OTP:** ", as shown below.

```
if(password) {
  password = strdup(password);
  DPRINT(LOG_DEBUG, "Got password %s", password);
}

/* no previous password: maybe get one from the user */
if (!password) {
  if (ctrl & PAM_USE_FIRST_PASS) {
    retval = PAM_AUTH_ERR;    /* use one pass only, stopping if it fails */
    goto error;
  }

  /* check to see if we send a NULL password the first time around */
  if (!(ctrl & PAM_SKIP_PASSWD)) {
    retval = rad_converse(pamh, PAM_PROMPT_ECHO_OFF, "Enter OTP: ", &password);
    PAM_FAIL_CHECK;

  }
} /* end of password == NULL */

build_radius_packet(request, user, password, &config);
/* not all servers understand this service type, but some do */
add_int_attribute(request, PW_USER_SERVICE_TYPE, PW_AUTHENTICATE_ONLY);

/*
```

For other versions of PAM, search for the similar section of code.

4. Save the file and close it.
5. Compile the PAM module to apply the changes.

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.	