SafeNet Authentication Service

Configuration Guide

SAS Agent for FreeRADIUS

SafeNet. THE DATA PROTECTION COMPANY

Document Information

Product Version	SAS Agent for FreeRADIUS1.04
Document Part Number	007-012432-001, Rev B
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

Applicability

The information in this document applies to:

- **SafeNet Authentication Service (SAS)** - A cloud authentication service of SafeNet Inc.
- **SafeNet Authentication Service – Service Provider Edition (SAS-SPE)** - The software used to build a SafeNet authentication service.
- **SafeNet Authentication Service – Private Cloud Edition (SAS-PCE)** - A term used to describe the implementation of SAS-SPE/PCE.

Note: References to BlackShield and CRYPTOCard reflect CRYPTOCard branding prior to acquisition by SafeNet. Over time these references will change to reflect SafeNet branding including program installation locations.

Environment

Supported Platforms	Red Hat Enterprise 5.9 x64

Overview

The following components are included in the SAS FreeRADIUS solution:

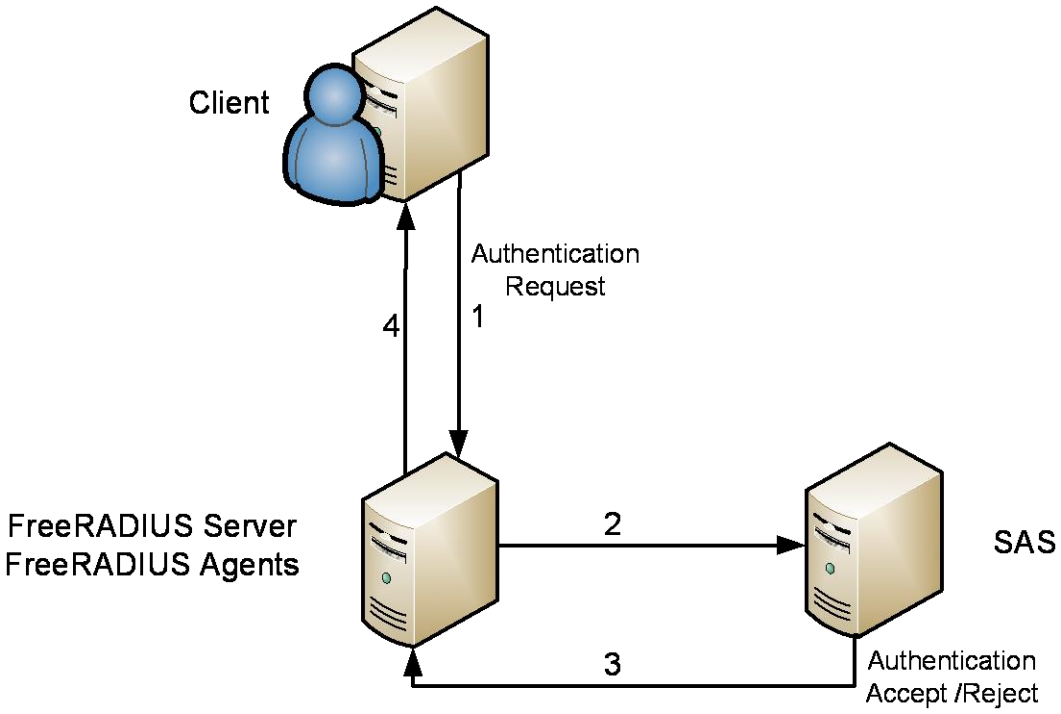
- FreeRADIUS Agent
- FreeRADIUS Server
- FreeRADIUS Updater (not described in this document).

The SafeNet Authentication Service FreeRADIUS Agent is a strong authentication agent that enables RADIUS clients to communicate with SafeNet Authentication Service through the RADIUS protocol. This document explains in detail how to configure FreeRADIUS to authenticate against SafeNet Authentication Service.

This agent uses an encrypted key file to communicate with the authentication server. This ensures all authentication attempts made against the server are from valid recognized agents. To accomplish this, a key file is loaded and registered with SafeNet Authentication Service agents, and then a matching key is registered with the server.

NOTE: A sample key file (Agent.bsidkey) has been installed for evaluation purposes; however, we strongly recommended that you generate your own key file for a production environment, as the sample file is publicly distributed.

Authentication Flow



1. Client sends authentication request to FreeRADIUS server.
2. FreeRADIUS server sends a web-service request to SAS.
3. If authentication is successful, SAS returns an affirmative response to FreeRADIUS server.
4. FreeRADIUS server sends acceptance of authentication request to Client.

About FreeRADIUS Server

The FreeRADIUS is a radius server which receives authentication data using the RADIUS protocol.

The RADIUS protocol itself (http://en.wikipedia.org/wiki/RADIUS) is simple protocol running over IP/UDP or IP/TCP. The protocol itself does not depend on IP/UDP or IP/TCP.

The FreeRADIUS server (http://freeradius.org/) can be integrated with LDAP, LINUX user names. rlm extensions can be written for communication with any external server.

Configuration of FreeRadius server is not trivial. It has the following directory structure:

var

etc

share

include

sbin

bin

lib64

lib ---------- DirList (1)

NOTE: On some systems lib64 or lib may be absent. It uses lib or lib64 for the location of .so files.

The following information refers to specific installations, depending on the compilation:

var directory could be /var

On a SAS FreeRADIUS server: **/opt/freeradius/freeradius-server-2.2.0/var**

Tip: Look for directory **log/radius**

etc directory could be /etc

On a SAS FreeRADIUS server it is /opt/freeradius/freeradius-server-2.2.0/etc

Tip: look for directory raddb

share directory could be /usr/share

On a SAS FreeRADIUS server it is /opt/freeradius/freeradius-server-2.2.0/share

Tip: look for directory freeradius

include directory could be /usr/include

On a SAS FreeRADIUS server it is /opt/freeradius/freeradius-server-2.2.0/include

Tip: look for directory freeradius

sbin directory could be /sbin (less likely) or /usr/sbin (more likely)

On a SAS FreeRADIUS server it is /opt/freeradius/freeradius-server-2.2.0/sbin

Tip: look for binary program radiusd

bin directory could be /bin (less likely) or /usr/bin (more likely)

On a SAS FreeRADIUS server it is /opt/freeradius/freeradius-server-2.2.0/bin

Tip: look for binary program radclient

lib directory could be /lib (less likely) or /usr/lib (more likely)

On a SAS FreeRADIUS server it is /opt/freeradius/freeradius-server-2.2.0/lib

Tip: look for rlm_*.so files

lib64 directory could be /lib64 (less likely) or /usr/lib64 (more likely)

On a SAS FreeRADIUS server it is /opt/freeradius/freeradius-server-2.2.0/lib64

Tip: look for rlm_*.so files

FreeRADIUS is application software for compiling natively and running on Linux/UNIX type environments. It can also be compiled and run on recent Mac OS and Solaris platforms.

Native compilation and running on Windows is not supported.

Control is exercised by changing information in the directories. Depending on the configure-time compilation flags used by the developer, the directories mentioned in “DirList (1)” for every FreeRADIUS server do exist and must be located.

In SAS FreeRADIUS server, they are all found in a separate folder in: **/opt/freeradius/freeradius-server-x.x.x**

After identifying each of the directories for the FreeRADIUS server, each must be configured as required for the FreeRADIUS configuration.

Configuring FreeRADIUS Server

As explained in the previous section, we need to identify the directories for FreeRADIUS in DirList (1).

The following must be taken into consideration before configuring FreeRADIUS Server:

- All commands and filenames in Linux are CASE sensitive; therefore, the exact case must be entered. After entering a file name, press the Tab key so that the rest of the file name is not automatically filled in.
- All Linux commands that are executed on a Linux system require granted permissions to run the specific commands.

Allow Traffic from External IP Addresses

There is a **raddb** directory in the **etc** directory. The folder contains a file called **radiusd.conf**.

This configuration file contains about 800 lines. Administrators managing the server need to familiarize themselves with this file.

radiusd.conf

Compile time decisions are displayed at the head of the file:

prefix = /opt/freeradius/freeradius-server-2.2.0

exec_prefix = ${prefix}

sysconfdir = /opt/freeradius/freeradius-server-2.2.0/etc

localstatedir = /opt/freeradius/freeradius-server-2.2.0/var

sbindir = /opt/freeradius/freeradius-server-2.2.0/sbin

logdir = ${localstatedir}/log/radius

raddbdir = ${sysconfdir}/raddb

radacctdir = ${logdir}/radacct

These lines assist in correcting errors when identifying the FreeRADIUS directories.

There are many configuration options, including **max_request_time, cleanup_delay, max_requests** etc. All can be reconfigured (see FreeRADIUS documentation for details).

There is a **listen** section that can reconfigured for different port numbers or for binding the listener to specific IP addresses if the machine has more than two IP addresses and it is not required to bind only to a specific IP.

There is typically a reference to read the allowed clients:

$INCLUDE clients.conf

clients.conf

This file must be configured to allow traffic from other computers.

This is the list of IP addresses from which traffic will be allowed into the server. Traffic whose IP addresses are not included in this file will be ignored by the server.

To allow all traffic from 192.168.40 subnet with a shared secret, such as “TopNotchSecret123”, add the following lines:

```
client 192.168.40.0/8 {
        secret =   TopNotchSecret123
        shortname = TopNotch-Network-40
}
```

The following is recommended:

- Configure appropriate and safe shared secrets for radius traffic.
- Allow only required traffic by specifying their IP addresses, rather than allowing all the traffic.
- By default, all traffic from 127.0.0.1 is allowed. Depending on the needs of the system this decision can be changed.

Authentication Test

Advanced users can skip this step and proceed to “Integrating FreeRADIUS Agent with SAS Server” on page 10.

A quick authentication test can be performed before SAS integration.

In the **/etc/raddb** there is a **users** file. This file can be changed for accepting authentication and rejecting authentication.

Save the **users** file:

cp users users.saved

Allow authentication for user “steve”:

uncomment the line for steve by changing:

#steve Cleartext-Password := "testing"

TO

steve Cleartext-Password := "testing"

Reject Authentications for **lameuser** by changing entries related to him as follows:

lameuser Auth-Type := Reject

Reply-Message = "Your account has been disabled."

Start **radiusd** (located in sbin folder of FreeRADIUS server) in debugging mode so that its outputs can be seen:

./radiusd –Xx

From another-window, send authentication for **steve**:

./radtest steve testing 127.0.0.1 1812 testing123

Expected result: authentication success

(**radtest** is located in the **bin** folder of FreeRADIUS server.)

From another window, send authentication for **lameuser**:

/radtest lameuser passDoesNotMatter 127.0.0.1 1812 testing123

Expected result: authentication fail

After validating two FreeRADIUS authentications, take the following steps:

STOP the radiusd server by using **Ctrl+C**.

Restore the users file.

cp users.saved users

rm users.saved

Successful execution of this step proves that the FreeRADIUS server is configured and running properly.

Integrating FreeRADIUS Agent with SAS Server

SAS Agent integrates FreeRADIUS server with SAS authentication server. SAS developers have written an rlm plugin for the FreeRADIUS server which has defined a reprogrammable, publishable interface. This rlm is free and is released as a GPL program similar to FreeRADIUS.

To this publishable interface described in the previous step, SAS developers have also added their proprietary suite of protocols which handles authentication with SAS Authentication Server.

Installation of cryptocard-freeradius-agent

Install cryptocard-freeradius-agent software using **rpm** or **deb**, depending on the Linux distribution.

The installer will create some files in **/usr/local/cryptocard/freeradius**.

The files from this directory will be used for integrating with the FreeRADIUS server.

Upgrading cryptocard-freeradius-agent

To upgrade cryptocard-freeradius-agent, do one of the following:

- Install cryptocard-freeradius-agent from the RPM package; the installed version will be updated.
- Uninstall the previous version and install the new version.

Overview of /usr/local/cryptocard/freeradius

The **rlm_mschap-versions.tar** and **rlm_challAvecAuth-versions.tar** files contain the specific **rlm** files required for integrating with the server.

The **lib** directory has SafeNet proprietary libraries required for communicating with the SAS server for handling communication, encryption, and failover from primary to secondary.

The **Open Source Licenses** directory displays the open source licenses.

Agent.bsidkey is the key for encrypting the authentication string with the server.

CRYPTOCard-license.txt licensing from SafeNET.

The **cryptocardFreeRadiusConfig** file has configuration sections that need to be configured for successful deployment.

Post installation configuration for cryptocard-freeradius-agent

This comprises two steps:

- Integration of SafeNet rlm libraries with freeradius.
- Configuring cryptocardFreeRadiusConfig

Integration of SafeNet rlm libraries with FreeRADIUS.

Use the correct version of the rlm files from rlm_challAvecAuth-versions.tar and rlm_mschap-versions.tar for integration with FreeRADIUS server.

Create a temporary directory and get the tar files:

mkdir -p /tmp/freeRadAgentConfig

cd /tmp/freeRadAgentConfig

cp /usr/local/cryptocard/freeradius/rlm_challAvecAuth-versions.tar .

cp /usr/local/cryptocard/freeradius/rlm_mschap-versions.tar .

tar xvf rlm_challAvecAuth-versions.tar

tar xvf rlm_mschap-versions.tar

After this step the rlm files are located in **/tmp/freeRadAgentConfig/rlm_challAvecAuth-versions** and **/tmp/freeRadAgentConfig/rlm_mschap-versions**.

The correct rlm versioned file will be integrated with the FreeRADIUS server.

Integration of rlm_challAvecAuth

Identify the version of the FreeRADIUS server to integrate with.

In this documentation FreeRADIUS version 2.2.0 will be used.

Go to the rlm lib or lib64 directory of the FreeRADIUS server you have identified

in DirList (1). In this example, the directory is: **/opt/freeradius/freeradius-server-2.2.0/lib64**

cd /opt/freeradius/freeradius-server-2.2.0/lib64

cp /tmp/freeRadAgentConfig/rlm_challAvecAuth-versions/rlm_challAvecAuth-2.2.0.so .

ln -sv rlm_challAvecAuth-2.2.0.so rlm_challAvecAuth.so

Add AuthType

Go to share/freeradius directory with the help from DirList (1)

In this example, it is:

/opt/freeradius/freeradius-server-2.2.0/share/freeradius

cd /opt/freeradius/freeradius-server-2.2.0/share/freeradius

Discover existing auth-types by running the command:

grep Auth-Type dictionary.freeradius.internal

Output as follows:

```
ATTRIBUTE       Auth-Type                       1000    integer
ATTRIBUTE       Post-Auth-Type                  1014    integer
#       For send/recv CoA packets (like Auth-Type, Acct-Type, etc.)
VALUE   Auth-Type               Local           0
VALUE   Auth-Type               System          1
VALUE   Auth-Type               SecurID         2
VALUE   Auth-Type               Crypt-Local     3
VALUE   Auth-Type               Reject          4
VALUE   Auth-Type               ActivCard       5
VALUE   Auth-Type               EAP             6
VALUE   Auth-Type               ARAP            7
VALUE   Auth-Type               Accept          254
VALUE   Auth-Type               PAP             1024
VALUE   Auth-Type               CHAP            1025
VALUE   Auth-Type               PAM             1027
VALUE   Auth-Type               MS-CHAP         1028
VALUE   Auth-Type               MSCHAP          1028
VALUE   Auth-Type               Kerberos        1029
VALUE   Auth-Type               CRAM            1030
VALUE   Auth-Type               NS-MTA-MD5      1031
VALUE   Auth-Type               SMB             1033
VALUE   Auth-Type               MS-CHAP-V2      1034
```

VALUE Post-Auth-Type Local 0

Add a new Auth-Type using suitable next number.

In this example following line is added to dictionary.freeradius.internal

VALUE Auth-Type challAvecAuth 1035

The line is added immediately after the following line:

VALUE Auth-Type MS-CHAP-V2 1034

Add challAvecAuth module

Identify etc/raddb/modules directory from DirList (1)

In this example it is /opt/freeradius/freeradius-server-2.2.0/etc/raddb/modules

Go to the identified directory

cd /opt/freeradius/freeradius-server-2.2.0/etc/raddb/modules

copy a file

cp -v /usr/local/cryptocard/freeradius/challAvecAuth .

Permission, owner and group membership on the challAvecAuth file must be similar to other files in the directory. For example:

-rw-r----- 1 root radiusd 147 2010-03-16 08:25 challAvecAuth

Configure challAvecAuth for authorize and authenticate

Go to etc/raddb directory with the help from DirList (1)

In this example it is /opt/freeradius/freeradius-server-2.2.0/etc/raddb

Add one line below to the authorize section:

challAvecAuth

for the file default in /opt/freeradius/freeradius-server-2.2.0/etc/raddb/sites-enabled

Add the three lines below to the **authenticate** section:

```
Auth-Type challAvecAuth {
        challAvecAuth
        }
```

for file default in /opt/freeradius/freeradius-server-2.2.0/etc/raddb/sites-enabled

To use inner-tunnel perform the following steps:

Add one line below to the authorize section:

challAvecAuth

for the file inner-tunnel in /opt/freeradius/freeradius-server-2.2.0/etc/raddb/ sites-enabled

Add the three lines below to the authenticate section:

Auth-Type challAvecAuth {

challAvecAuth

}

for the file inner-tunnel in /opt/freeradius/freeradius-server-2.2.0/etc/raddb/ sites-enabled

If you have more inner-tunnel servers, challAvecAuth Authentication can configure them using their authorize and authenticate sections.

configure /usr/local/cryptocard/freeradius/cryptocardFreeRadiusConfig

The file has 32 sections numbered from 0 to 31. Some are explained below:

- Section 0: Configuration Version Number. This is used for upgrade, do not change.
- Section 1: This is logging level for rlm_challAvecAuth.so and rlm_mschap.so only which run directly under full control of radiusd process. It has 2 configuration options and 0 is default.
- Section 2: This is the directory for logging the suite of communication libraries in /usr/local/cryptocard/freeradius/lib. Generally it is left as /usr/local/cryptocard/freeradius/log unless Customer needs a different directory for this logging.
- Section 3: This is the logging level for communication libraries in /usr/local/cryptocard/freeradius/lib
- Section 4: The Agent.bsidkey file for authentication string encryption supplied by BlackShield-ID Server. File contains sensitive information.
- Section 5: Customization or localization of messages for Challenge reply.
- Section 16: Primary TokenValidator IP Address.

The most important is Section 16. Add here the IP address of the primary TokenValidator.

The IP passing mechanism can be used and can be configured in Section 7.

Linking with correct version of openssl

The FreeRADIUS server uses openssl for many purposes. Do the following to ensure that FreeRADIUS is properly linked to the correct and tested version of openssl libraries.

Identify lib directory from DirList (1).

In this example it is /opt/freeradius/freeradius-server-2.2.0/lib64

Note: on x86 systems, the directory is /opt/freeradius/freeradius-server-2.2.0/lib

From the identified lib directory run the following commands:

ln -sv /usr/local/cryptocard/freeradius/lib/libssl.so.1.0.0 libssl.so.6

ln -sv /usr/local/cryptocard/freeradius/lib/libcrypto.so.1.0.0 libcrypto.so.6

Running radiusd server

FreeRADIUS server is the radiusd binary, using several dynamic loadable libraries known as rlm libraries.

Identify sbin directory from DirList (1).

In this example it is /opt/freeradius/freeradius-server-2.2.0/sbin

Running in debug mode

goto sbin directory from DirList (1).

cd /opt/freeradius/freeradius-server-2.2.0/sbin

./radiusd -Xx

To stop the program, press Ctrl-C.

Running without debug and for production

goto sbin directory from DirList (1).

cd /opt/freeradius/freeradius-server-2.2.0/sbin

./radiusd

To stop the program, identify the process id for radiusd using the ps command and send a TERM signal.

Depending on the Linux Distribution and start-up run level entries can be added.

Example: For SuSE this directory is /etc/init.d

Scripting is generally uses bourne shell or bash shell scripting.

Basic System administration guide can be looked-up for accomplishing the same goal.

Integration of rlm_mschap

SafeNet integration of FreeRADIUS includes full support for the mschapv2 protocol including password change.

The following changes to the existing server are necessary to use the mschapv2 as an authentication mechanism.

Ensure challAvecAuth fully configured and working.

Test PAP authentication

goto etc directory from DirList (1).

In this example it is /opt/freeradius/freeradius-server-2.2.0/etc

cd raddb/modules

Add the following 2 lines to mschap module (/opt/freeradius/freeradius-server-2.2.0/etc/raddb/modules/mschap):

use_mppe = yes

require_encryption = yes

Add the following 3 lines to mschap module:

external_module = yes

send_challenge_as_reject = yes

mschap_success_pack_limit_len = yes

Replace existing rlm_mschap.so with SafeNET supplied library

Go to the rlm lib or lib64 directory of the freeradius server you have identified in DirList (1). In this example the directory is /opt/freeradius/freeradius-server-2.2.0/lib64

cd /opt/freeradius/freeradius-server-2.2.0/lib64

rm rlm_mschap.so

rm rlm_mschap-2.2.0.so

cp /tmp/freeRadAgentConfig/rlm_mschap-versions/rlm_mschap-2.0.0.so .

ln -sv rlm_mschap.so rlm_mschap-2.2.0.so

create a symbolic link

ln -sv /usr/local/cryptocard/freeradius/lib/libchallAvecAuthImplement.so /usr/local/cryptocard/freeradius/lib/librlm_mschap_external.so.1

Configuring FreeRADIUS Agent Failover (Optional)

The FreeRADIUS Agent may be configured for failover. This is not necessary if there is only a single SAS Site that is running.

Modifying the cryptocardFreeRadiusConfig file

To modify the cryptocardFreeRadiusConfig file:

1. Browse to the directory where the **cryptocardFreeRadiusConfig** file resides by entering the following command:

   cd /usr/local/cryptocard/freeradius

2. Open the **cryptocardFreeRadiusConfig** file by entering the following command:

   vi cryptocardFreeRadiusConfig

3. Start editing by pressing **Insert** on the keyboard. The bottom of the command prompt will save **–INSERT--**.
4. Change the following values in the **cryptocardFreeRadiusConfig** file:
   - Section 9 - change the value from 1 to 2.
   - Section 24 - change the value from 127.0.0.1 to the IP address of the Secondary SAS Server.

   Note: Changes must be made to the cryptocardFreeRadiusConfig file on both the Primary and Secondary FreeRADIUS Server (If applicable).
5. Press **Esc** on the keyboard.
6. Enter **:wq**

   Restart the **radius** daemon from within the **/etc/init.d** directory

).

Working with FreeRADIUS

After FreeRADIUS has been configured, it is recommended that you work with FreeRADIUS in single user mode to ensure that it has been configured properly.

To start working with FreeRADIUS:

1. Go to the directory where the **radius daemon** resides by entering the following command:

   cd /opt/freeradius/freeradius-server-2.1.11/sbin
2. Before starting the **radiusd** daemon, enter the following commands:
   - ln -s /lib64/libssl.so.0.9.8e /usr/lib64/libssl.so.0.9.8
   - ln -s /lib64/libcrypto.so.0.9.8e /usr/lib64/libcrypto.so.0.9.8
3. Start the **radiusd daemon** in single user mode by entering the following command:

   ./radiusd -Xx
4. The last line displays **Ready to process** requests. This indicates that all configuration changes are correct.
5. If necessary, press **Ctrl + C** to terminate radius in single user mode.
6. Create a symbolic link from the **rc.radiusd** to the **/etc/init.d** directory by entering the following command:

   ln -s /opt/freeradius/freeradius-server-2.1.11/sbin/rc.radiusd /etc/init.d/radiusd
7. Start the **radiusd daemon** by entering the following command:

   /etc/init.d/./radiusd start

   The radius.log file is located in /opt/freeradius/freeradius-server-2.1.11/var/log/radius.

Ensure that the configuration changes above are performed on both the Primary and Secondary FreeRADIUS Server (If applicable).

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 phone 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.	