THALES

SafeNet Agent for FreeRADIUS 3.2.0

INSTALLATION AND CONFIGURATION GUIDE

Document Information

Product Version	3.2.0
Document Part Number	007-012432-002
Release Date	May 2021

Revision History

Revision	Date	Reason
E	March 2021	
F	May 2021	Thales Rebranding and Template Migration

Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

CONTENTS

PREFACE

Audience

This document is intended for personnel responsible for maintaining your organization's security infrastructure. It is assumed that the users of this document are proficient with security concepts.

All products manufactured and distributed by Thales are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

Document Conventions

This section describes the conventions used in this document.

Command Syntax and Typeface Conventions

This document uses the following conventions for command syntax descriptions, and to highlight elements of the user interface.

Convention	Description
bold	The bold attribute is used to indicate the following: > Command-line commands and options (Type **dir /p**.) > Button names (Click **Save As**.) > Check box and radio button names (Select the **Print Duplex** check box.) > Window titles (On the **Protect Document** window, click **Yes**.) > Field names (**User Name:** Enter the name of the user.) > Menu names (On the **File** menu, click **Save**.) (Click **Menu** > **Go To** > **Folders**.) > User input (In the **Date** box, type **April 1**.)
italic	The italic attribute is used for emphasis or to indicate a related document. (See the *Installation Guide* for more information.)
Double quote marks	Double quote marks enclose references to other sections within the document.
<variable>	In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.

[optional] [<optional>]	Square brackets enclose optional keywords or <variables> in a command line description. Optionally enter the keyword or <variable> that is enclosed in square brackets, if it is necessary or desirable to complete the task.
[a | b | c] [<a> | <b> | <c>]	Square brackets enclose optional alternate keywords or variables in a command line description. Choose one command line argument enclosed within the braces, if desired. Choices are separated by vertical (OR) bars.
{ a | b | c } { <a> | <b> | <c> }	Braces enclose required alternate keywords or <variables> in a command line description. You must choose one command line argument enclosed within the braces. Choices are separated by vertical (OR) bars.

Notifications and Alerts

Notifications and alerts are used to highlight important information or alert you to the potential for data loss or personal injury.

Tips

Tips are used to highlight information that helps to complete a task more efficiently.

> **TIP:** This is some information that will allow you to complete your task more efficiently.

Notes

Notes are used to highlight important or helpful information.

> **NOTE:** Take note. Contains important or helpful information.

Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss.

> **CAUTION!** Exercise caution. Contains important information that may help prevent unexpected results or data loss.

Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury.

> ****WARNING** Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.**

Related Documents

The "**SafeNet Agent for FreeRADIUS: Customer Release Notes**" document contains related or additional information.

Support Contacts

If you encounter a problem while installing, registering, or operating this product, please refer to the documentation before contacting support. If you cannot resolve the issue, contact your supplier or Thales Customer Support.

Thales 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 Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at https://supportportal.thalesgroup.com, is where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable database of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

> **NOTE:** You require an account to access the Customer Support Portal. To create a new account, go to the portal and click the **REGISTER** link.

Telephone Support

The support portal also lists telephone numbers for voice contact (Contact Us).

Email Support

You can also contact technical support by email at technical.support.DIS@thalesgroup.com.

CHAPTER 1: Introduction

About FreeRADIUS Server

The SafeNet Agent for FreeRADIUS is a highly secure, enterprise authentication agent that enables RADIUS clients to communicate with SafeNet Authentication Service (SAS) and SafeNet Trusted Access (STA) using SSL/TLS.

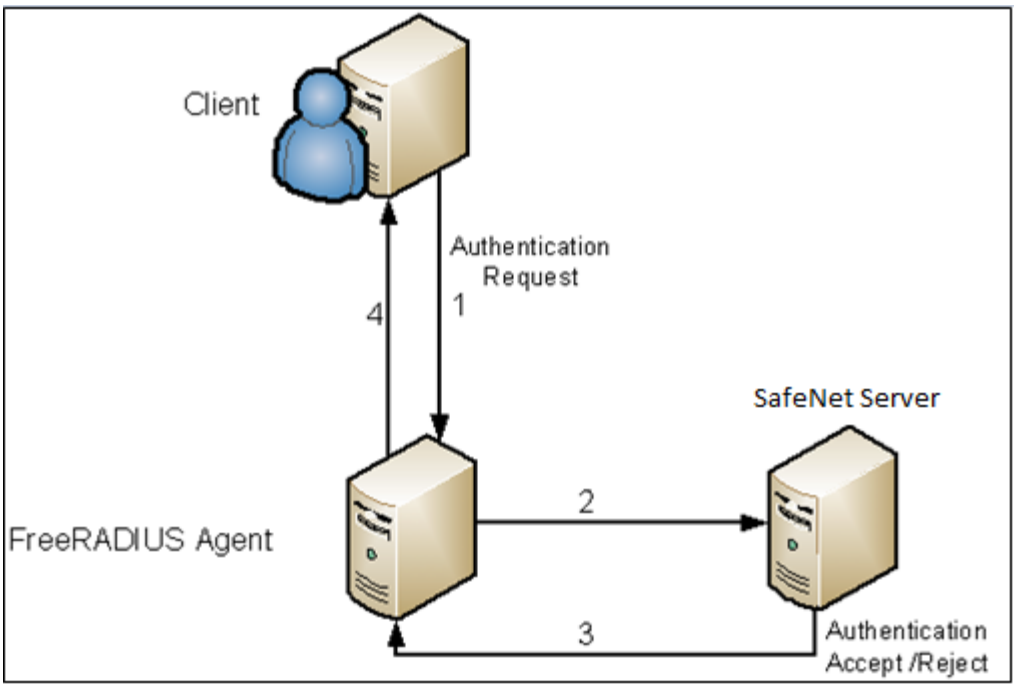
This document explains in detail how to install and configure FreeRADIUS agent to authenticate against SAS or STA.

The agent uses an encrypted key file to communicate with the SafeNet server. This ensures that all authentication attempts made against the server are from valid recognized agents. To accomplish this, a key file is generated at the SafeNet server and provided / loaded at the agent.

Authentication Flow

The following steps broadly depict the flow of actions for the agent solution:

1. The client sends an authentication request to the FreeRADIUS agent.
2. The FreeRADIUS agent sends a web-service request to the SafeNet server.
3. If authentication is successful, SafeNet server returns an affirmative response to the FreeRADIUS agent.
4. The FreeRADIUS agent sends acceptance of the authentication request to the client.



Compatibility

The information in this document applies to:

SafeNet servers

- SafeNet Authentication Service PCE/SPE 3.7 and later (SAS)
- SafeNet Trusted Access (STA)

Supported Platforms

- Red Hat Enterprise Linux 7.3
- Red Hat Enterprise Linux 8.3
- CentOS 7.3
- CentOS 8.3

Supported RADIUS Protocols

- PAP
- MSCHAP-v2
- PEAP

Supported LDAP Server for on-prem password validation

- Microsoft Active Directory

Prerequisites

- Before executing the agent's deployment script, ensure that Docker is installed. Refer to the https://docs.docker.com/get-docker/ link to install Docker.
- **Optional**: PEAP adds a TLS layer on the top of EAP and uses TLS to authenticate the server to the client. Web server certificate is required to use PEAP.

CAUTION! All commands and file names in Linux are case sensitive; therefore the exact case must be entered.

CAUTION! All commands executed on a Linux system require granted permissions to run specific commands.

Setting up FreeRADIUS API for SAS PCE/SPE

RADIUS API requests radius client's data from SAS PCE/SPE to dynamically update the clients in FreeRadius database.

In case of a fresh SafeNet server installation with MySQL database, RADIUS API encounters an issue with MySQL database (*MySQL EF6 DLL in GAC missing*). It is a limitation of MySQL Connector 6.10.7.

Before installation ensure that the following steps are performed:

1. After installing SafeNet server, install MySQL 6.10.7 Connector.
2. Configure SafeNet server with MySQL database.

Follow the steps:

For Windows Server 2008 R2 SP1

1. Copy the following text in a text file and save the file in the **.ps1** file format:

```
$config_text = @"
<?xml version="1.0"?>
<configuration>
<startup useLegacyV2RuntimeActivationPolicy="true">
<supportedRuntime version="v4.0.30319"/>
<supportedRuntime version="v2.0.50727"/>
</startup>
</configuration>
"@
$config_text| Out-File $pshome\powershell.exe.config
$config_text| Out-File $pshome\powershell_ise.exe.config
```

2. Save it and rename the file extension to **.ps1**.
3. Run as Administrator in the PowerShell.

NOTE: For Windows Server 2008 R2 SP1, the administrator also needs to follow the steps in the **For Windows Server 2012, 2012 R2 and 2016** section.

For Windows Server 2012, 2012 R2 and 2016

1. Copy the following text in a text file and save the file in the **.ps1** file format:

```
#Note that you should be running PowerShell as an Administrator
[System.Reflection.Assembly]::Load("System.EnterpriseServices, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")
$publish = New-Object System.EnterpriseServices.Internal.Publish
$publish.GacInstall("C:\Program Files (x86)\MySQL\MySQL Connector Net
6.10.7\Assemblies\v4.5.2\MySql.Data.Entity.EF6.dll")
# If installing into the GAC on a server hosting web applications in IIS, you
need to restart IIS for the #applications to pick up the change.
Iisreset
```

2. Run the **.ps1** file, as an Administrator in the PowerShell.
3. Reset IIS.

Points to Remember

- Default location:

```
System Directory:\Program Files (x86)\MySQL\MySQL Connector Net
6.9.9\Assemblies\v4.5
```

- If someone changes the directory location while installing the MySQL Connector, the above path also needs to be updated in the script.
- Open the PowerShell script and change the path to where your DLL resides.

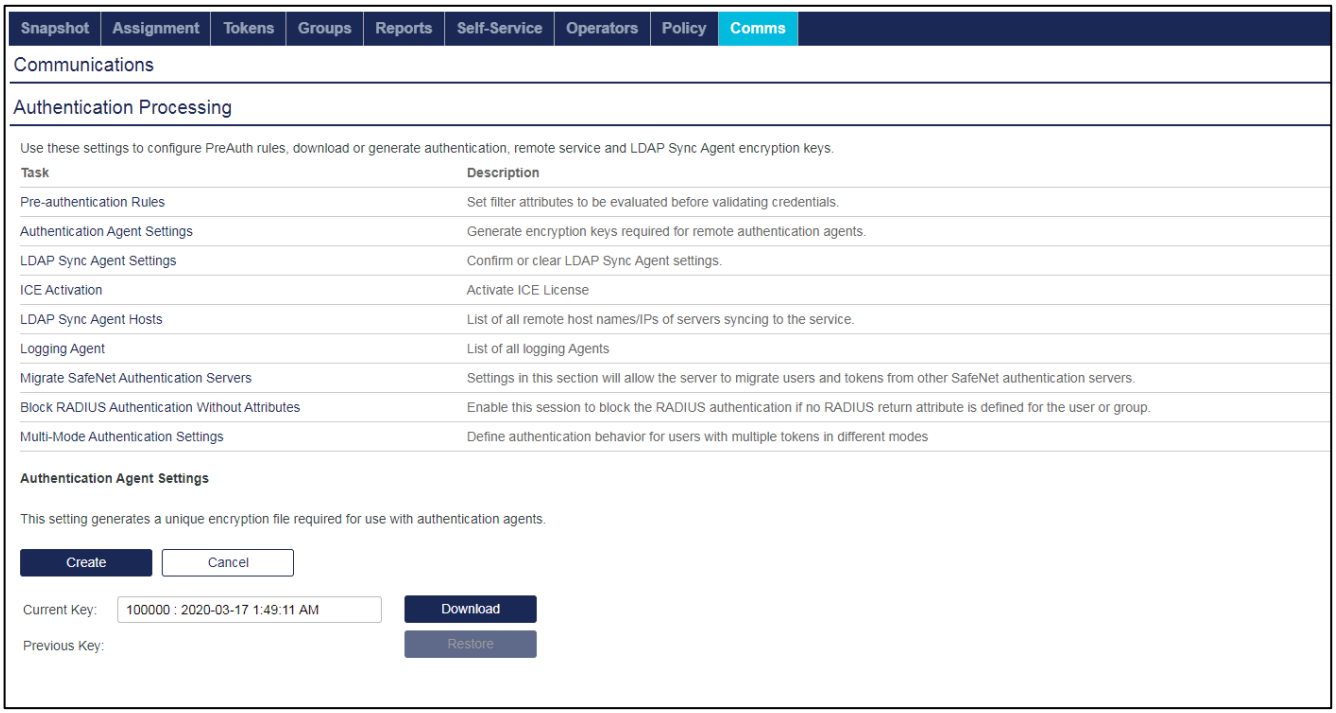
CHAPTER 2: Download BSID and JWT Keys

Downloading BSID Key File

The Agent.bsid key file is the encryption key file that is used to encrypt/decrypt the data. This file is required to manage communication with Token Validator.

Perform the following steps to download BSID Key file:

1. Login to your SafeNet server account, and navigate to **COMMS** > **Authentication Processing**.
2. Under **Task** list, click **Authentication Agent Settings** link and download the Agent.bsidkey key file.



NOTE: The key file must be kept at a directory on your FreeRADIUS host, accessible to all the authorized users.

Downloading JWT Key File

NOTE: JWT Key file is required if FreeRADIUS Synchronization is used with SAS PCE/SPE.

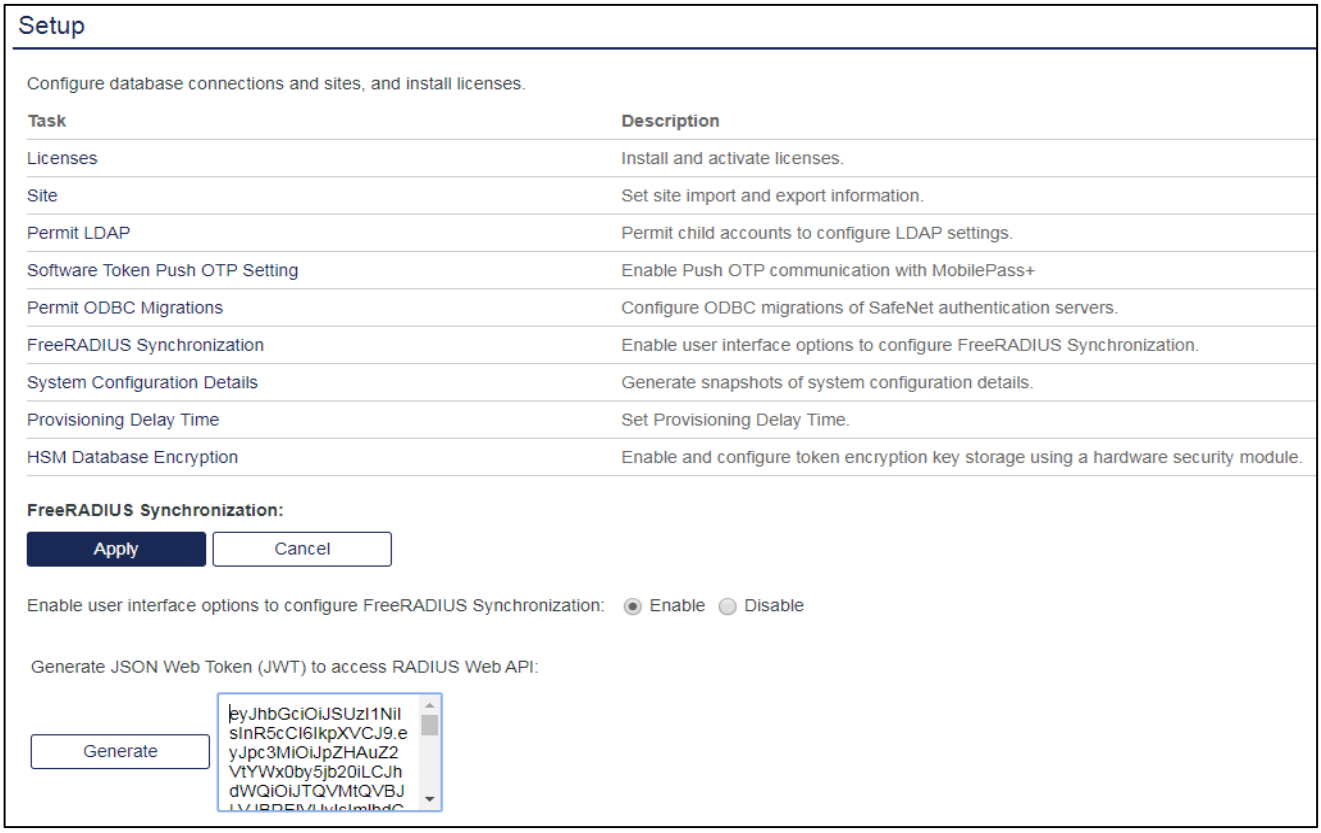
Perform the following steps to download JWT Key file:

1. Login to your SAS PCE/SPE account, and navigate to **SYSTEM** > **FreeRADIUS Synchronization**.
2. Click **Generate**.
3. Copy the JWT key token and place it in a file, named *jwtAuthToken.key*.

NOTE:

- Ensure that the file name is jwtAuthToken.key

  (with .key being the extension).
- Ensure that the FreeRADIUS Synchronization radio button is enabled.
- The key file must be kept at a directory on your FreeRADIUS host, accessible to all the authorized users.



CHAPTER 3: Installation

Copy the FreeRADIUS package to your local Linux system. The package should consists of the following components:

> **FreeRADIUS package**

- FreeRADIUSv3.sh (Agent Deployment Shell Script)
- Client_Updater.sh (Manual RADIUS Client Updater Script)
- Startup.sh (Startup Shell Script)

> **Image tar File**

- FreeRADIUS Image: Load the image into your local system using the following command:

```
docker load –i <image name>.tar
```

NOTE: Ensure that all the scripts are in executable mode for the user running the shell scripts. To confirm, execute the following command:

```
chmod +x ./<name of shell script>
```

Installing the Solution

Execute the `FreeRADIUSv3.sh` shell script from the package by running the command where the shell script is present: `./FreeRADIUSv3.sh`

NOTE: To upgrade, execute the `FreeRADIUSv3.sh` shell script. The script will handle all upgrade scenarios, automatically.

The following steps detail the flow for component check, and prompt for inputs as necessary before completing the agent installation:

1. The first component, the script checks for, is if the Docker is installed. If the Docker is present, the following message is displayed:

```
Docker setup is a prerequisite for this installation.
Checking...

Found Docker. Continuing...
```

   If the Docker is not installed, follow the steps to **Install Docker**.

2. The script then checks if any of the following two FreeRADIUS components are present:
   - FreeRADIUS Agent

- FreeRADIUS Updater

It also checks, if there is a FreeRADIUS container already present. If all three components are present, visit the **Upgrade** section.

```
FreeRADIUS Agent for RADIUS Server 2.x not found. Continuing...

FreeRADIUS Updater for RADIUS Server 2.x not found.
Continuing...

Docker container for RADIUS Server 3.x not found. Continuing...
```

> **NOTE:** The above steps check for prerequisites. The following steps confirm details from the user, and help in the agent's installation.

Press [ENTER] to continue.

3. If the default protocol for creating a container (by using the HTTP protocol) needs to be changed, type **Y** and press [ENTER]. If no change is required, type **N** and press [ENTER].

```
The default protocol is HTTP. Change to HTTPS (recommended)? Y/N
If using SafeNet Trusted Access (STA) ensure that Y is entered.

Y
```

> **NOTE:** The default protocol must be changed for **SafeNet Trusted Accees (STA)**, since STA only support HTTPS protocol.

4. Enter SafeNet Server IP or FQDN, and press [ENTER].
   a. Please provide the input as mentioned below.

      a.1) For SAS PCE/SPE, please enter primary FQDN or Server IP.

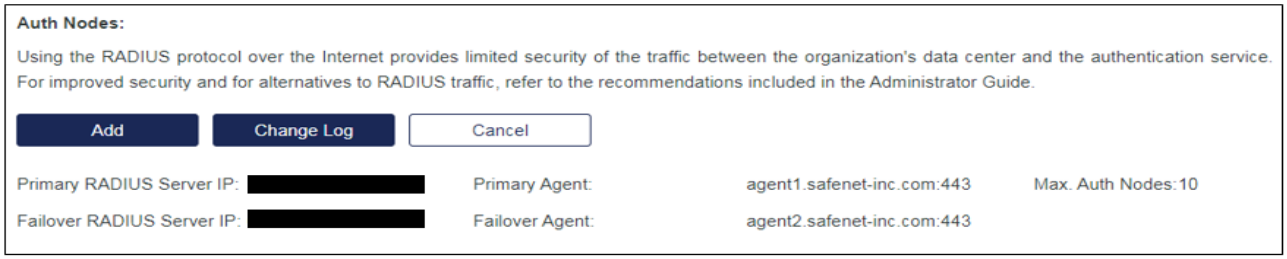
      a.2) For SafeNet Trusted Access (STA), please enter primary FQDN. Server IP is not permitted.

```
Please enter the Fully Qualified Domain Name (FQDN) of the
authentication service (SAS/STA).

NOTE: If using SafeNet Authentication Service (SAS-PCE) IP address
is optionally permitted.

xxxxxxxxx
```

   b. For **STA**, enter the FQDN of PrimaryAgent depending on your service zone. To locate the details, navigate to your **STA Console** > **COMMS** > **Auth Nodes**.



The script validates if the SafeNet server Token Validator is accessible.

```
Making sure the authentication endpoint is accessible...
The authentication endpoint is accessible.
```

NOTE: While providing the FQDN or hostname, ensure that the FQDN or hostname is accessible from RADIUS server machine. The hostname entry of the SafeNet server is provided under `/etc/hosts` file.

The user input needs to be in lowercase letters.

5. Check if the RADIUS Client API is accessible or not. The RADIUS Client API is responsible for updating RADIUS Client configurations from the SafeNet server to the FreeRADIUS server. This is applicable to SAS PCE/SPE only.
   a. For **SAS PCE/SPE**, the RADIUS Client API is accessible. Press **Y** and press [ENTER].

```
Is the SAS RADIUS Client API URL accessible? Y/N
If using SafeNet Trusted Access (STA) ensure that N is entered.
Y
```

   i. The script validates if the RADIUS Client API is accessible.

```
Making sure the SAS RADIUS Client API URL is accessible...
The SAS RADIUS Client API URL is accessible.
```

   ii. Enter the complete path, including the file name, of the JWT key file. The JWT authentication token file can be downloaded from the SafeNet server. To download the key from the SafeNet server, follow the steps at the Downloading JWT Key File section.

```
Please enter the complete path of JWT key file.
/etc/docker/FreeRadiusAgent/keys/jwtAuthToken.key

Making sure the agent BSID key file exists at the provided path...
```

The script validates if the JWT key file is available at the given path, and displays an appropriate message. The file is copied at the following location: `/usr/local/FreeRADIUS/Files/jwtAuthToken.key`

The copied file is mounted on FreeRADIUS Container.

b. For **STA Cloud**, the RADIUS Client API is not accessible. Press **N** and press [ENTER].

```
Is the SAS RADIUS Client API URL accessible? Y/N

If using SafeNet Trusted Access (STA) ensure that N is entered.

N
```

6. Enter the complete path, including the file name, of the Agent BSID key file.

The key file can be downloaded from the SafeNet server, and is used to encrypt/decrypt the authentication string with the server. To download the key from the SafeNet server, follow the steps explained under **Downloading BSID Key File** section.

```
Please enter the complete path of the agent BSID key file (Agent.bsidkey).

/etc/docker/FreeRadiusAgent/keys/Agent.bsidkey


Making sure the agent BSID key file exists at the provided path...
```

The script validates if the Agent BSID key file is available at the given path, and displays an appropriate message. The file is copied at the following location: `usr/local/FreeRADIUS/Files/Agent.bsidkey` The copied file is used to mount onto the FreeRADIUS Container.

7. If you wish to use "*FreeRADIUS Agent support for on-prem password validation*", enable concatenated credentials whereby an LDAP Password and an OTP is sent in a single password field. To use credential concatenation, enter **Y** as input and go to step **7.a**

Please ensure that you type **N** if you want to use the default behavior, where only OTP validation works and move to point **8**.

```
Do you want to enable validation of concatenated LDAP Password+OTP as a
single field? Y/N

N
```

a. To configure the Agent with credential concatenation, please ensure to Type **Y** at step 7, and provide the LDAP details as mentioned below. After completing these inputs, go to point 8 to continue configuration.

i. Enter the LDAP Server IP or Hostname.

```
Enter LDAP Server IP or Hostname

10.0.1.25
```

ii. Enter the LDAP port which is configured with your LDAP Server.

```
Enter LDAP Port to connect on. Example : 389
389
```

iii. Enter the LDAP service account Details.

```
Enter LDAP Identity Common Name. Example :cn=Administrator,cn=Users


cn=Administrator,cn=Users
```

iv. Enter the LDAP service account Password.

```
Enter LDAP Administrator Password
```

v. Enter the Base DN of your LDAP Server.

```
Enter LDAP Domain Controller Base DN. Example:dc=example,dc=com

dc=testdomain,dc=com
```

vi. Enter the delimiter to split the password for authentication.

```
Enter the Delimiter which will be used to split the LDAP AD password
and OTP while authenticating
NOTE: Delimiters like '%', '('Left Bracket, '.', '?', and '*' will
not work with the Agent. Please use any other Delimiter
,
```

Note: Make sure the LDAP Server details are entered correctly. The FreeRADIUS container may crash if the details provided are not correct.

Note: With complex LDAP password and OTP it is recommended to use string as a delimiter to avoid issues with user passwords having the same character in delimiter

8. To determine FreeRADIUS agent response in case of the authentication service being unavailable, Please provide '**Y**' as input to fail silently, else go with '**N**' as your input (returns access-reject).

```
Do you want the service to be silent (do_not_respond) when SAS/STA is
unavailable? Y/N
NOTE: Not responding may help customer controlled failover.

Y
Setting Value Accordingly
```

For Input as '**N**'

```
Do you want the service to be silent (do_not_respond) when SAS/STA is
unavailable? Y/N
NOTE: Not responding may help customer controlled failover.

N
```

NOTE: By default the FreeRADIUS agent would respond as an Access-Reject in case the SafeNet server is not reachable, if '**Y**' is provided as an input then the RADIUS Client would respond as "No response from the server" this will help to control failover.

9. This step is Optional. For PEAP support with a strong EAP type, such as TLS with certificates, both the client and the server use certificates to verify their identities to each other. Certificates must meet specific requirements, both on the server and on the client machine for successful authentication.

By default, the FreeRADIUS image contains these certificates at the following path: `/opt/gemalto/certs`

a. For using default certificates, press **Y** and press [ENTER].

```
Do you wish to use default certificates for PEAP support? Y/N
Y
```

b. For using proprietary/own certificates, press **N** and press [ENTER].

```
Do you wish to use default certificates for PEAP support? Y/N
N
```

i. Certificate Authority Certificate is the first required certificate. Enter the path of the certificate.

```
Enter complete path of the Certificate Authority certificate.

/etc/docker/certsFRv3/cacert.pem
```

The script will validate if the certificate is present at the provided path. The Certificate Authority Certificate will then be copied to the following location, `usr/local/FreeRADIUS/Certs`, and renamed as *ca.pem*.

ii. Key Certificate is the second required certificate. Enter the path of the certificate.

```
Enter complete path of the Key Certificate.

/etc/docker/certsFRv3/keycert.pem
```

- The script will validate if the certificate is present at the provided path. The Key Certificate will then be copied to the following location: `usr/local/FreeRADIUS/Certs`, and renamed as *ca.pem*.
- The script will then prompt the user to enter passphrase if the Key Certificate is passphrase protected. If the certificate is not passphrase protected, press [ENTER] to continue.

```
Enter Passphrase for Key Certificate. If no Passphrase exists
Press [Enter] to continue.
```

iii. Client Certificate is the third required certificate. Enter the path of the certificate.

```
Enter complete path of the Client Certificate.
/etc/docker/certsFRv3/servercert.pem
```

The script will validate if the certificate is present at the provided path. The Client Certificate will then be copied to the following location, `usr/local/FreeRADIUS/Certs`, and renamed as *server.pem*

10. Enter external port number of the FreeRADIUS container.

```
Please enter the Port Number FreeRadius will listen to (1812).
1812

The Port is accessible.
```

The script validates if the port is accessible or not, and displays an appropriate message. If the port is not accessible, the script **does not exit**.

11. Select whether you want to use the default Encoding Format iso8859 or utf8. This configuration option will determine how to interpret username/password. By default the agent supports iso-8859-1 encoding which means that there is no support for non-European languages.

a. To select the default option of **'iso8859'.** Type **N**, and press [ENTER].

b. To change the option to '**utf8'.** Type **Y**, and press [ENTER].

```
By default the FreeRADIUS agent is configured for ISO-8859-1 encoding.
Change to UTF8? Y/N
NOTE: Changing to UTF8 may help support national characters such as
å,ä,ö (but these MUST be supported in the full architecture).

N
```

12. Select whether you want to use 'SYSLOG' or a JSON-FILE as the default log driver for the FreeRADIUS container. By default, the script will deploy 'SYSLOG' as the default log driver.

```
By default the log driver for the FreeRADIUS container is set to 'SYSLOG'.
Change to JSON-FILE? Y/N
```

a. To select 'SYSLOG' as the default log driver: Type **N**, and press [ENTER].

```
By default the log driver for the FreeRADIUS container is set to
'SYSLOG'. Change to JSON-FILE? Y/N

N
```

i. To select local host as SYSLOG Server: Type **N**, and press [ENTER].

```
Do you want to use an external Syslog server? Y/N
N

Checking if syslog daemon is running on host machine..

Redirecting to /bin/systemctl status  rsyslog.service
Syslog daemon is running on host machine
```

```
```

ii. To select external host as a SYSLOG Server: Type **Y**, and press [ENTER].

```
Do you want to use an external Syslog server? Y/N
Y
```

```
Provide the Syslog server address in this format:
<Protocol>://<Syslog_Server_IP>:<Port>, example:
udp://127.0.0.1:514"
```

```
Checking if syslog daemon is running on host machine..

Redirecting to /bin/systemctl status  rsyslog.service
Syslog daemon is running on host machine
```

NOTES:

- Enter SYSLOG Server address in the following format:
  `<Protocol>://<SYSLOG_Server_IP>:<Port>`

  For example, `udp://127.0.0.1:514 udp://10.164.45.44:514`
- SYSLOG log drivers are saved under: /var/log/messages
- The FreeRADIUS Container logs can be distinguished from system logs. The former will have the FreeRADIUS tag before them.

b. To select a JSON-FILE as the default log driver: Type **Y**, and press [ENTER]. Configure the JSON file as below:

i. To set **'JSON-FILE'** as the log driver and create the FreeRADIUS container, the container values need to be set. If you type **N** (and press [ENTER]), the script, by default, creates a container with **max-size=50m** and **maxfile=5** values. If you want to change these values, type **Y**, and press [ENTER].

```
Setting log driver to 'JSON-FILE' with default values: max-size=50m
and maxfile=5. Do you want to change these values? Y/N

Y
```

ii. The script will prompt for the values required for max-size and maxfile fields, and creates the container with the supplied values.

```
Enter value for maz-size (followed by 'm', 'k', or 'g').
1000k
Enter value for max-file
7
```

NOTE:

- **max-size** field: Maximum log size before it is rolled.
- Accepted Values: Positive integer followed by the unit of measure

  (k for kilobytes, m for megabytes or g for gigabytes).
- **maxfile** field: Maximum number of log files that can exist in a container. This field is effective only when the max-size field is set. If rolling the logs create excess files, the oldest file is removed.

  Accepted Values: Positive integer.

  The Json file logs can be located by executing the following command:

  ```
  docker logs   -f FreeRADIUSv3
  ```

13. Using the given values, the script creates a FreeRADIUS container.
14. The script validates if the container starts, using the provided inputs.

```
Making sure the FreeRADIUS Container has been deployed successfully...
FreeRadius container FreeRADIUSv3 has been deployed successfully.
```

15. The FreeRADIUS container is ready to use.

```
FreeRADIUS container 'FreeRADIUSv3' is ready to process requests.
```

NOTES:

- In case of fresh installations, no uninstallations are required. In case of upgrades, old installations need to be uninstalled
- To ensure uninterrupted access to the FreeRADIUS Container (after installation), execute the Startup Script.

Test Authentication.

The passcode required for Authentication will be LDAP password followed by OTP code (split by delimiter). For example, if you are using NTRadping, then the password will be concatenation of LDAP password followed by delimiter followed by OTP code.

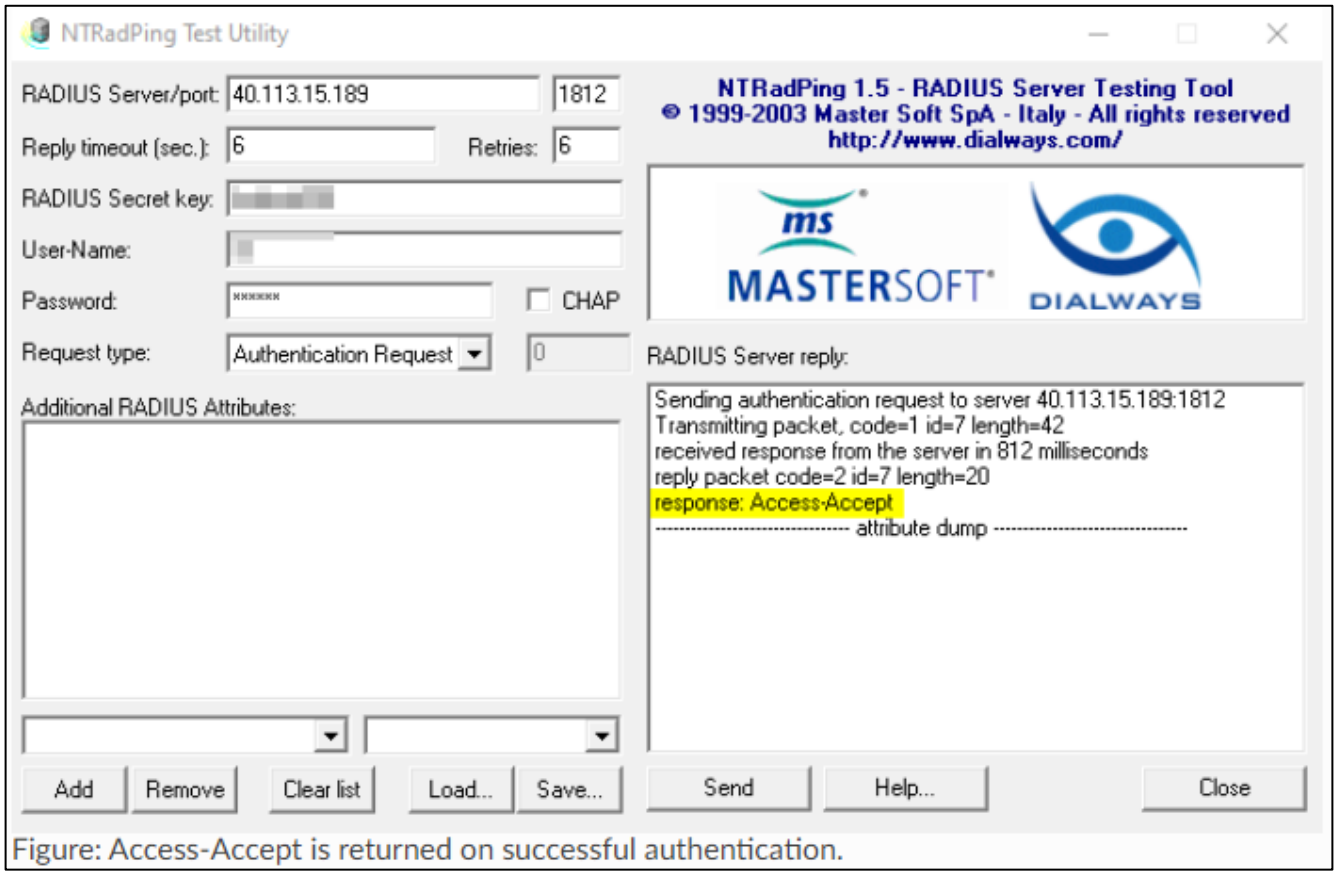


Figure: Access-Accept is returned on successful authentication.

JSON and SYSLOG Drivers

JSON Logging Driver

JSON logging driver allows to capture the standard output (and standard error) in JSON format, in files which annotate each line with its origin (`stdout` or `stderr`) and timestamp.

SYSLOG Logging Driver

SYSLOG logging driver allows you to route logs to a SYSLOG server. The SYSLOG message must be formatted in a specific way, to enable valid extraction of the information.

CHAPTER 4: Upgrade

The upgrade scenarios will be handled by the deployment shell script `FreeRADIUSv3.sh` by itself. No input is required.

Earlier FreeRADIUS releases before v3.0 had following three components:

> FreeRADIUS Agent

> FreeRADIUS Updater

> FreeRADIUS Server

For the agent solution to work, all three components needed to be installed and configured. Now, the agent is deployed using a simple shell script. Executing the script will confirm some details from the end user, and will help in quick, hassle-free agent upgradation.

Upgrading SafeNet Agent for FreeRADIUS 1.X with FreeRADIUS Server 2.X

Follow the steps to upgrade the agent with FreeRADIUS Server 2.X installations:

1. Execute the `FreeRADIUSv3.sh` shell script from the package.
2. The shell script will stop the RADIUS Daemon, if it finds the agent.

```
FreeRadius Agent for Radius Server 2.X found.
Stopping the Radius daemon…
```

3. If found, the shell script will then stop the FreeRADIUS Updater service.

```
FreeRadius Updater for Radius Server 2.X found.
Stopping the FreeRadius Updater Service…
Stopping the freerad_updaterservice (via systemctl):                    [ OK ]
```

> **NOTE:** The above two steps check for prerequisites. If the prerequisites are met, the FreeRADIUS container will be deployed.

4. After the container has been deployed successfully, the script will remove the previous installations.

```
New container deployed successfully. Checking old installations…
```

5. The script also removes the installed RPMs packages of **FreeRADIUS Agent** and **FreeRADIUS Updater**.

```
Uninstalling the FreeRadius Agent and Free Radius Updater RPMs…
Uninstallation completed.
```

Upgrading SafeNet Agent for FreeRADIUS with FreeRADIUS Server 3.X

Follow the steps to upgrade the agent with FreeRADIUS Server 3.X installations, if in addition to **FreeRADIUS Agent** and **FreeRADIUS Updater**, a FreeRADIUS container is already present.

1. Execute the `FreeRADIUSv3.sh` shell script from the package.
2. The shell script will check if a FreeRADIUS container already exists. If it exists, the script stops and renames the existing container to *FreeRADIUSv3_Old*. The renaming is done to deploy a new FreeRADIUS container (*FreeRADIUSv3*) as containers with same names cannot be deployed.

```
Docker container for RADIUS Server 3.x found.
Stopping old container to deploy new FreeRADIUS container…
FreeRADIUSv3

Renaming old container to 'FreeRADIUS_Old'…
```

> **NOTE:** The above step checks for the additional prerequisite. If the prerequisite is met, the FreeRADIUS container will be deployed.

3. After the container has been deployed successfully, the script will remove the previous installations.

```
New container deployed successfully. Checking old installations…
```

4. The script also removes the old FreeRADIUS container.

```
New container deployed successfully. Checking old installations…

Uninstalling the old container…
FreeRADIUSv3_Old
```

> **NOTE:** To ensure uninterrupted access to the FreeRADIUS container (after installation), execute the **Startup Script**.

CHAPTER 5: Startup Script

Docker provides restart policies to restart your containers automatically when they exit or when Docker restarts. Restart policies ensure that linked containers are started in the correct order. Docker recommends the use of restart policies and avoid using process managers to start containers.

For more information, refer to restart policies.

CHAPTER 6: Manual Client Updater

CRUD Script

This section describes how to manually create, update, delete and view the RADIUS Client using the C – Create, U – Update, D – Delete, V – View (Client_Updater script).

Execute the `./Client_Updater` from the package. The following steps detail the flow for the components it checks, and the inputs it prompts for, before completing a CRUD operation.

NOTE:

- In case of SAS PCE/SPE, if the RADIUS Client API is accessible and FreeRADIUS Synchronization is enabled, the Auth Nodes from SAS PCE/SPE server get updated to the FreeRADIUS server.
- On the other hand, for SafeNet Trusted Access (STA), or if FreeRADIUS Synchronization is not used with SAS PCE/SPE, , the Auth Nodes from STA do not get updated to the FreeRADIUS server. All entries have to be done manually using the Client_Updater script.

C – Create

1. After the following prompt, type **C**, and press [ENTER].

```
Choose appropriate actions as C-Create, View-V, U-Update or D-Delete
(C/V/U/D) and press [ENTER]:
C
```

The screen will display the current list of clients:

```
Existing FreeRADIUS Client(s):
 id |    nasname      |    shortname     |  secret   | description
----+-----------------+------------------+-----------+-------------
  1 | 10.164.44.211   | NtRadping        | 63048     |
  2 | 10.164.108.173  | Ntradping_local  | 1234      |
  3 | 10.164.45.11    | 10.164.45.11     | Temp123#  |
  4 | 10.164.44.62    | 10.164.44.62     | Temp123#  |
  5 | 127.0.0.1       | 127.0.0.1        | Temp123#  |
  6 | 10.164.44.133   | 10.164.44.133    | 1111      |
  7 | 10.1.1.1        | test             | 1234      |
(7 rows)
```

2. Enter IP address of the RADIUS Client.

```
Enter RADIUS Client IP and press [ENTER].
10.6.2.1
```

3. Enter a name for the RADIUS Client.

```
Enter RADIUS Client IP and press [ENTER].
10.6.2.1
```

4. Enter a Secret key.

```
Enter secret key and press [ENTER]:
1234
```

5. Enter a description.

```
Enter description and press [ENTER].
Test Client Added for RADIUS Auth
```

6. The RADIUS Client is now added. The container will run with the new changes.

```
Client added successfully.
```

U – Update

1. After the following prompt, type **U**, and press [ENTER

```
Choose appropriate action as C-Create, U-Update or D-Delete (C/U/D) and
press [ENTER]:
```

2. The records that can be updated will be listed. The records are identified by the ID value (the number displayed against them).

```
Existing FreeRADIUS Client (s):
Id    |     nasname          |     shortname  |     secret     |description
----------+-------------------------------+-------------------------------
   1  |     10.164.44.211    |     NtRadping  |     12356      |
   2  |     10.164.44.211    |     NtRadping2 |     Temp123#   |
   3  |     10.164.44.211    |     NtRadping3 |     1234@Temp  |
   4  |     10.164.44.211    |     NtRadping4 |     1234       |
(4 rows)
```

> **NOTE:** As an example, we will illustrate how to update some fields of the third (3) record.

3. Enter the required ID value.

```
Enter the ID value and press [ENTER].
3
```

4. Enter a short name in the field that you want to update and press [ENTER]

```
Enter the field name you want to update and press [ENTER].
shortname
```

> **NOTE:** As an example, we will illustrate how to update shortname and secret fields.

5. Enter a new value and press [ENTER].

```
Enter the new value to update for shortname
Newtest
```

6. The script then prompts the user if any other field needs to be updated. Enter **Y** to update else, enter **N** to exit the script.

```
Do you want to Update any other field, press [Y/N]

Y
```

7. If you have entered **Y**, the script prompts to update the field name.

```
Type field name to update

secret
```

8. Enter the new value to update the secret field.

```
Enter new value to update for secret

1234
```

9. If there is no other field to update, enter **N**.

```
Do you want to update any  other field, press [Y/N]

N
```

10. On exit, the following message is displayed.

```
Exiting, please re-run the script to update any other field or Id
```

The script will show the updated ID with the fields:

```
Id    |     nasname     |     shortname   |     secret      |     description
----------+---------------------------------+---------------------------------
   3  |   10.164.45.11 |     Newtest     |     1234        |
(1 row)
```

D – Delete

1. After the following prompt, type **D**, and press [ENTER].

```
Choose appropriate action as C-Create, U-Update, or D-Delete (C/U/D) and
press [ENTER]:
D
```

2. The records that can be deleted will be listed. The records are identified by the ID value (the number displayed against them).

```
Existing FreeRADIUS client(s):
Id  |    nasname     |   shortname     | type | ports  |  secret | server |  community | description
-----+---------------------+----------------------+-------+---------+----------+----------+----------------+-----------------
  1 | 10.164.44.211   | NtRadping       | other |       |63048  |       |            |
  2 | 10.164.100.173  | NtRadping_local | other |1234 |1234    |       |            |
```

NOTE: As an example, we will illustrate how to delete the second (2) record.

3. Enter the required ID value.

```
Enter the ID value and press [ENTER].
2
```

4. The RADIUS Client record is now deleted. After you restart, the container will run with the new changes.

```
RADIUS Client deleted successfully.
```

5. If required, perform View (V) to check if the record has been deleted successfully.

V – View

To view the client table, type **V** and press [ENTER].

```
Choose appropriate action as C-Create, View-V, U-Update or D-Delete
(C/V/U/D) and press [ENTER]:
V
```

The following table is displayed to confirm:

```
Id    |      nasname          |    shortname    |   secret    |   description
------+-----------------------+-----------------------+-----------------
   1  |  10.164.44.211    | NtRadping        |  63048     |
(1 row)
```

CHAPTER 7: Troubleshooting

Inaccessible RADIUS Client API

Problem: The Script output displays the message, **The SAS RADIUS Client API is not accessible**.

```
Validating the SAS RADIUS Client API URL is accessible…
The SAS RADIUS Client API is not accessible.
```

Solution: Check if the Client API is accessible on the SAS PCE/SPE server. This can be verified by following the steps:

1. Open **IIS Manager** on the SAS PCE/SPE Server.
2. In the left Pane, navigate to, **localhost website** > **Sites** > **Default Web Site**.
3. Click **/RADIUS**.
4. In the right Pane, Click **Browse *:80 (http)**.

> **NOTE:** If the RADIUS Client API does not allow browsing, the Agent Deployment Script will not be able to access the RADIUS Client API from the RADIUS Server.
>
> Also, check if the RADIUS Client API URL is accessible from the RADIUS Server. The URL for RADIUS Client API is in the following format:
> http://<IP>/RADIUS/api/RADIUS/clients

Inaccessible FreeRADIUS Container Port

Problem: The Script output displays the message, **The Port is not accessible**.

```
The Port is not accessible. Ensure that the Port is not use by other process
or routine.
```

Solution: Check if the port is in use by any other process or routine. Execute the following command to check if the port is in use.

```
netstat –tuplen | grep <Port number>
```

Unsuccessful Deployment of FreeRADIUS Container

Problem: The Script output displays the message, **The FreeRADIUS container is not deployed successfully**.

Solution: Check the logs as per Log Drivers.

- If JSON file driver is used, check using the following command:

  ```
  docker logs –f FreeRADIUSv3
  ```

- If SYSLOG file driver is used, check using the following command against FreeRADIUS tag.

```
cat /var/log/messages
```

Bad Interpreter – Bin / Bash File

Ensure that the user has privileges to execute the `/bin/bash` file.

To verify, open your script with *vi* or *vim*. Enter through *vi* command mode (ESC key), and then type the following and save it: `:set fileformat=unix`

To save, use the following text: `:x! or :wq!`

Hostname Resolution for STA (and SAS PCE/SPE)

Problem: If the logs displays the following error, **Hostname does not resolve**.

Solution: Ensure that the IP and the hostname entries are present under the following path: `/etc/hosts`

STA Hostname and IP are provided under the **Auth Node** section.

APPENDIX A: About Protected Extensible Authentication Protocol

The Protected Extensible Authentication Protocol (PEAP) is a common authentication protocol for communication between a VPN server and mobile devices. The protocol works by covering the Extensible Authentication Protocol (EAP) within an encrypted and authenticated Transport Layer Security (TLS) tunnel.

EAP-PEAP uses TLS to authenticate only the Server-to-Client communication (and not the Client-to-Server communication). This ensures that only the server is required to have a public key certificate. Once the client is satisfied about the credibility of the server's identity, the client and server exchange a sequence of EAP messages encapsulated within TLS.

FreeRADIUS has a built-in feature to locally terminate the TLS outer tunnel, decrypt the PEAP tunnel, and then extract the inner identity to proxy the MS-CHAPv2 authentication mechanism to another RADIUS server.

PEAP creates two concentric tunnels:

- An encrypted and authenticated TLS outer tunnel.
- An inner tunnel that uses an EAP method (such as EAP-MS-CHAPv2) for authentication, and is protected by the TLS outer tunnel.

An example of configuring certificates required for PEAP support is provided in Solution Configuration.

APPENDIX B: Solution Configuration

PEAP adds a TLS layer on top of EAP and uses TLS to authenticate the server to the client. To achieve this, the FreeRADIUS server is required to have a server certificate. As an example, a Microsoft CA is used, but any other CA can be designated to provide a server certificate. Key certificate file and Client certificate is also required along with CA Server certificate.

> **NOTE:** FreeRADIUS agent comes with the default certificates, which may or may not work with PEAP. You may create your own certificates if default certificates do not work.