SafeNet Authentication Client

Integration Guide

Using SAC CBA for Red Hat Enterprise Linux

Document Information

Document Part Number	007-012945-001, Rev. B
Release Date	May 2015

Trademarks

Disclaimer

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

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

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

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

Contents

Third-Party Software Acknowledgement

This document is intended to help users of SafeNet products when working with third-party software, such as Red Hat Enterprise Linux.

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

Remote access poses both a security and a compliance challenge to IT organizations.The ability to positively identify users – often remote users – requesting access to resources is a critical consideration in achieving a secure remote access solution. Deploying remote access solution without strong authentication is like putting your sensitive data in a vault (the datacenter), and leaving the key (user password) under the door mat.

A robust user authentication solution is required to screen access and provide proof-positive assurance that only authorized users are allowed access.

PKI is and effective strong authentication solution to the functional, security, and compliance requirements.

SafeNet Authentication Client (SAC) is a public key infrastructure (PKI) middleware that provides a secure method for exchanging information based on public key cryptography, enabling trusted third-party verification of user identities. SafeNet's certificate-based tokens provide secure remote access, as well as other advanced functions, in a single token, including digital signing, password management, network logon, and combined physical/logical access.

The tokens come in different form factors, including USB tokens, smart cards, and software tokens. All of these form factors are interfaced using a single middleware client, SafeNet Authentication Client (SAC). The SAC generic integration with CAPI, CNG, and PKCS#11 security interfaces enables out-of-the-box interoperability with a variety of security applications, offering secure web access, secure network logon, PC and data security, and secure email. PKI keys and certificates can be created, stored, and used securely with the hardware or software tokens.

SafeNet Authentication Manager (SAM) provides your organization with a comprehensive platform to manage all of your authentication requirements, across the enterprise and the cloud, in a single, integrated system. SAM enables management of the complete user authentication life cycle. SAM links tokens with users, organizational rules, and security applications to allow streamlined handling of your organization's authentication infrastructure with a flexible, extensible, and scalable management platform.

SAM is a comprehensive token management system. It is an out-of-the-box solution for Public Certificate Authorities (CA) and enterprises to ease the administration of SafeNet's hardware or software tokens devices. SAM is designed and developed based on the best practices of managing PKI devices in common PKI implementations. It offers robust yet easy to customize frameworks that meets different organizations' PKI devices management workflows and policies. Using SAM to manage tokens is not mandatory, but it is recommended for enterprise organizations.

For more information, refer to the *SafeNet Authentication Manager Administrator Guide*.

Red Hat Enterprise Linux (RHEL) is an American multinational software company providing open-source software products to the enterprise community. Red Hat has become associated to a large extent with its enterprise operating system Red Hat Enterprise Linux.

This document provides guidelines for deploying certificate-based authentication (CBA) for user authentication to Red Hat Enterprise Linux (RHEL) using SafeNet Tokens managed by SafeNet Authentication Manager.

It is assumed that theRed Hat Enterprise Linux environment is already configured and working with static passwords prior to implementing SafeNet multi-factor authentication.

Red Hat Enterprise Linux can be configured to support multi-factor authentication in several modes. CBA will be used for the purpose of working with SafeNet products.

This document provides guidelines for deploying certificate-based authentication (CBA) for user authentication to Red Hat Enterprise Linux using SafeNet tokens.

It is assumed that the Red Hat Enterprise Linux environment is already configured and working with static passwords prior to implementing SafeNet multi-factor authentication.

Applicability

The information in this document applies to:

- **SafeNet Authentication Client (SAC) —** SafeNet Authentication Client is the middleware that manages SafeNet's tokens.
- **Red Hat Enterprise Linux**

Environment

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

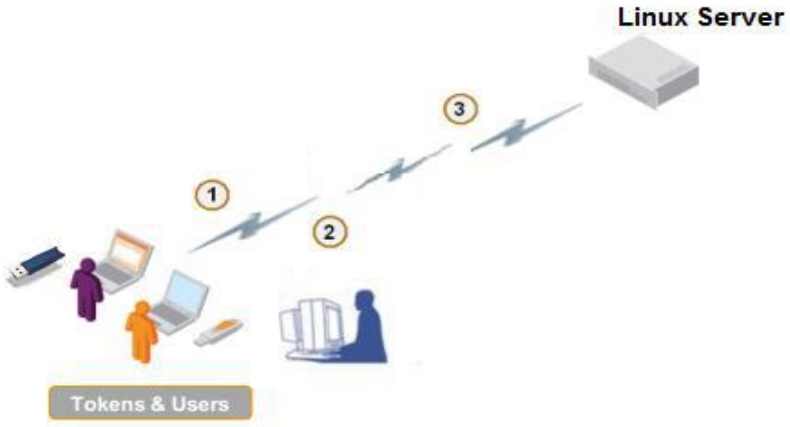
- **SafeNet Authentication Client (SAC) —** 9.0
- **Red Hat Enterprise Linux** - 6.3

Audience

This document is targeted to system administrators who are familiar with Red Hat Enterprise Linux and are interested in adding certificate-based authentication capabilities using SafeNet tokens.

CBA Flow using SAC

The diagram below illustrates the flow of certificate-based authentication:



1. The user attempts to log on to the RHEL computer using a logon manager or through a console.
2. The RHEL computer prompts the user for an eToken PIN (instead of a Linux user password).
3. A successful authentication is performed using the certificate on the token.

   The user is now logged in to the Linux computer without having provided a password.

Prerequisites

This section describes the prerequisites that must be installed and configured before implementing certificate-based authentication for Red Hat Enterprise Linux using SafeNet tokens:

- RHEL 6.3 x64 computer (not in the domain)
- SafeNet Authentication Client (SAC) 9 installed on the RHEL 6.3 x64 computer
- For more details see "Appendix B: Installation of SafeNet Authentication Client (SAC) on Linux" on page 17
- A PAM_PKCS11 module installed on the RHEL computer (pam_pkcs11-0.6.2-12.1.e16)
- An X.509 user certificate on SafeNet eToken 5100 and a Microsoft root CA certificate installed locally on the RHEL computer
- Configuration of the user(s) to be authenticated (on the RHEL 6.3 x64 computer)

Supported Tokens in SAC

SAC supports a number of tokens that can be used as second authentication factor for users who authenticate to Red Hat Enterprise Linux.

SafeNet Authentication Client 9.0 (GA) supports the following tokens:

Certificate-based USB tokens

- SafeNet eToken PRO Java 72K
- SafeNet eToken PRO Anywhere
- SafeNet eToken 5100/5105
- SafeNet eToken 5200/5205
- SafeNet eToken 5200/5205 HID and VSR

Smart Cards

- SafeNet eToken PRO Smartcard 72K
- SafeNet eToken 4100

Certificate-based Hybrid USB Tokens

- SafeNet eToken 7300
- SafeNet eToken 7300-HID
- SafeNet eToken 7000 (SafeNet eToken NG-OTP)

Software Tokens

- SafeNet eToken Virtual
- SafeNet eToken Rescue

Configuring Red Hat Enterprise Linux

Install CA Certificates

Download the root CA and sub-CA certificates in base 64 formats, and add them to the certificate database on the Linux computer.

The certificates are installed in the appropriate system database using the **Certutil** command.

Certutil can import a CA certificate into another directory server certificate database using following arguments:

-A -n certname -t trustargs [-h tokenname] [-d certdir] [-a]
[-i cert-request-file]

Example:

```
# certutil -A -d /etc/pki/nssdb -n "RootCA" -t "CT,C,C" -i /tmp/ca_cert.crt
```

Add the eToken Module to the NSS Database

Add the SafeNet eToken library to the NSS database using the **modutil** command using the following arguments:

-add <moduleName> -libfile <library File> -dbdir <dbFolder>

The **modutil** tool is a command-line utility for managing PKCS #11 module information stored in **secmod.db** files or hardware tokens. You can use the tool to add and delete PKCS #11 modules, change passwords, set defaults, list module contents, enable or disable slots, enable or disable FIPS 140-2 compliance, and assign default providers for cryptographic operations. This tool can also create **key3.db**, **cert8.db**, and **secmod.db** security database files.

Example:

 # modutil –add “SafeNet eToken” –libfile /usr/lib64/libeTPkcs11.so –dbdir /etc/pki/nssdb

The following message verifies that the eToken module was successfully added to the database :

Module “SafeNet eToken“ added to database.

Configuring the PAM-PKCS11 Module

The PAM-PKCS11 module uses the **/etc/pam_pkcs11** directory for configuration.

For details see "**15Appendix A: PAM-pkcs11 Configuration Files (Reference"** on page 15.

1. Pam-pkcs11 needs a list of recognized Certificate Authorities to properly validate user certificates.
   a. Create a folder on the Linux computer:

      # mkdir /etc/pam_pkcs11/cacerts

   b. Copy all CA Certificates to the **cacerts** directory

      # cp < CA certificate directory> < /etc/pam_pkcs11/cacerts>

Create hash links to CA certificates with the provided **cacertdir_rehash**.

 # cacertdir_rehash /etc/pam_pkcs11/cacerts

NOTE: If CRL is used (the **crl_policy** option in **module** is set to ***offline*** or ***auto***), repeat the process described above using the CRL directory (**/etc/pam_pkcs11/crls**).

2. Open the pam-pkcs11 configuration file (**/etc/pam_pkcs11/pam_pkcs11.conf**) on the Linux computer and add the SafeNet eToken pkcs11 library in the **pam_pkcs11** section:

```
pkcs11_module eToken  {
   module = /usr/lib64/libeTPkcs11.so
   description = "eToken"
   slot_num = 0;
   support_threads = true ;
   ca_dir = /etc/pam_pkcs11/cacerts;
   nss_dir = /etc/pki/nssdb;
   cert_policy = ca,signature;
 }
```

3. Update **use_ pkcs11_ module** with the added **SafeNet pkcs11** module named **eToken**.

   Edit **use_pkcs11_module = eToken** in **pam_pkcs11** section in **/etc/pam_pkcs11/pam_pkcs11.conf** file.

4. Select the mapper you want to use for login user-mapping.

   If your selected mapper module uses login mapping, create and set up mapping files.

   Edit the **use_mappers** variable with the mapper information in the **PAM_PKCS11** configuration file (**/etc/pam_pkcs11/pam_pkcs11.conf**).

Configuring Mappers

The following can be used as mappers:

- Certificate Common Name (CN)
- Microsoft Universal Principle Name (UPN)

For details, see "**Appendix A: PAM-pkcs11 Configuration Files (Reference)" on page 15**.

Using Certificate CN as a Mapper

To use Certificate CN as a mapper, make the following modifications to the mapper cn section:

```
use_mappers = cn;
mapper cn {
        debug = false;
        module = internal;
        # module = /usr/$LIB/pam_pkcs11/cn_mapper.so;
        ignorecase = false;
        mapfile = "file:///etc/pam_pkcs11/cn_map";
                }
```

Create a file named **/etc/pam_pkcs11/cn_map** and add a mapping CN to the username as follows:

<Common Name> -> <login>

where **<Common Name>** is the CN field on the user certificate, and **<login>** is the RHEL user login name.

Using Microsoft UPN as a Mapper

To use Microsoft UPN as mapper, make the following modifications to the mapper ms section:

```
use_mappers = ms;
mapper ms {
        debug = false;
        module = internal;
        #module = /usr/lib/pam_pkcs11/ms_mapper.so;
        ignorecase = false;
        ignoredomain = false;
        domainname = "domain.com";
                }
```

Configuring Application-specific Configuration Files (pam.d files)

The PAM module enables configuration of the PAM-aware application you want to implement. All of the applications listed in the **/etc/pam.d** directory are PAM aware. The SafeNet token is expected to work for most applications in the **pam.d** directory.

Configuring Console and Graphical Login

Login

The **LOGIN** PAM file controls local console login sessions. The following **LOGIN** PAM configuration file enables PAM_PKCS11 authentication.

LOGIN		
#%PAM-1.0		
auth	required	pam_pkcs11.so
account	required	pam_unix.so
password	required	pam_unix.so
session	required	pam_unix.so

GDM (Graphical Desktop Logon)

PAM_PKCS11 authentication can be enabled for users who log on to Gnome. The following changes are required for the GDM login manager to enable SafeNet eToken authentication.

/etc/pam.d/gdm-passwd

#%PAM-1.0		
auth	required	pam_pkcs11.so
account	required	pam_unix.so
password	required	pam_unix.so
session	required	pam_unix.so

Running the Solution

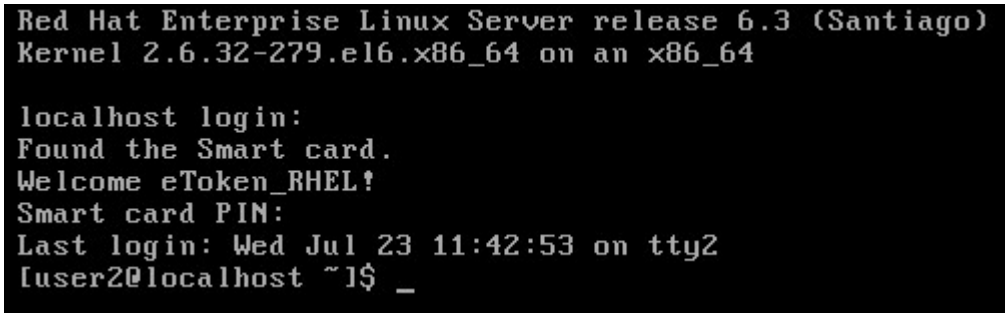
Configure the PAM_PKCS11 module as described previously (see “Configuring the PAM-PKCS11 Module” on page 9) and then restart the computer. You can now use the SafeNet token or smart card to authenticate to configured PAM-aware Linux services without requiring a static password.

Logging In Using the Console

5. Restart the computer, or press or **CTRL+ALT+F2**. The console login window is displayed.
6. When prompted for the login, type a space and then press **Enter**.

   You are prompted to enter the smart card PIN.

7. Upon entry of a valid PIN, the user is successfully logged in.



Logging In Using the Graphical Login Manager

8. Open the **/etc/inittab** file for editing.
9. Change the following: **id:3:initdefault:** to **id:5:initdefault:**
10. Restart the computer. The Gnome login window is displayed.
11. Enter the smart card PIN, and then click **Log In.**

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

The user is logged in.

Troubleshooting Tips

- **To enable graphical logon globally on startup:**

  a. Open the **/etc/ inittab** file for editing.

  b. Change the following: **id:3:initdefault** to **id:5:initdefault**

- **To install the PAM-PKCS11 module on Linux:**

  Run the following command: **# yum install pam-pkcs11**

- **To add a user on Linux:**

  Run the following command: **# useradd <username>**

- **To set a password for the new user:**

  Run the following command: **# passwd <username>**

- **To map certificates to a user:**

  Insert a smart card into the smart card reader, and then run the following command:

  # pklogin_finder debug

  The command tries to find a map between installed certificates and a user login.

  If successful, **pklogin_finder** prints the login name on **stdout**.

- **To check that the RootCA certificate is correctly installed in the PKI store of the Linux computer:**

  Run the following command: **# certutil –L –d /etc/pki/nssdb**

- **To see what certificates are present:**

  Insert the smart card into smart card reader, and then type the following command: **pkcs11_listcerts**

  After entering the smart card PIN, all available certificates are listed in the following format:

  Certificate #

  -Subject: ... /CN=<Name> ...

  -Issuer:

  -Algorithm:

Appendix A: PAM-pkcs11 Configuration Files (Reference)

```
#
# Configuration file for pam_pkcs11 module
#
# Version 0.4
# Author: Juan Antonio Martinez <jonsito@teleline.es>
#
pam_pkcs11 {
  nullok = true;
  debug = true;
  card_only = true;
  use_first_pass = false;
  try_first_pass = false;
  use_authtok = false;
  use_pkcs11_module = eToken;
  screen_savers = gnome-screensaver,xscreensaver,kscreensaver

  pkcs11_module eToken  {
    module = /usr/lib64/libeTPkcs11.so
    description = "eToken"
    slot_num = 0;
    support_threads = true ;
    ca_dir = /etc/pam_pkcs11/cacerts;
    nss_dir = /etc/pki/nssdb;
    crl_dir = /etc/pam_pkcs11/crls;
    cert_policy = ca,signature;
  }

  # Which mappers ( Cert to login ) to use?
  # you can use several mappers:
  #
  # subject - Cert Subject to login file based mapper
  # pwent   - CN to getpwent() login or gecos fields mapper
  # ldap    - LDAP mapper
  # opensc  - Search certificate in ${HOME}/.eid/authorized_certificates
  # openssh - Search certificate public key in ${HOME}/.ssh/authorized_keys
  # mail    - Compare email fields from certificate
  # ms      - Use Microsoft Universal Principal Name extension
  # krb     - Compare againts Kerberos Principal Name
  # cn      - Compare Common Name (CN)
  # uid     - Compare Unique Identifier
  # digest  - Certificate digest to login (mapfile based) mapper
  # generic - User defined certificate contents mapped
  # null    - blind access/deny mapper
```

```
#
  # You can select a comma-separated mapper list.
  # If used null mapper should be the last in the list :-)
  # Also you should select at least one mapper, otherwise
  # certificate will not match :-)

  use_mappers = ms;


# When no absolute path or module info is provided, use this
  # value as module search path
  # TODO:
  # This is not still functional: use absolute pathnames or LD_LIBRARY_PATH
  mapper_search_path = /usr/$LIB/pam_pkcs11;


  # Search public keys from $HOME/.ssh/authorized_keys to match users
  mapper openssh {
        debug = false;
        module = /usr/$LIB/pam_pkcs11/openssh_mapper.so;
  }

  # Assume MS UPN to be the login
  mapper ms {
         debug = false;
         module = internal;
         # module = /usr/lib/pam_pkcs11/ms_mapper.so;
         ignorecase = false;
         ignoredomain = false;
         domainname = "SafenetDemos.com";
 }
```

Appendix B: Installation of SafeNet Authentication Client (SAC) on Linux

To install SAC 9 Client on RHEL:

12. In the Linux Terminal window, run the following command:

    # rpm –Uvh SafenetAuthenticationClient-9-0.x86_64.rpm

13. If during installation, pcsc-lite dependency is required, install the pcsc-lite package on the RHEL 6.3 computer.

To install pcsc-lite on RHEL:

Run the following command: **# yum install pcsc-lite**

Appendix C: SELinux Policy Update

If Security-Enhanced Linux (SELinux) is enabled, you must update the policy module to enable login with a smart card:

To update the policy module:

14. Copy the **safenet.te** file to the **/tmp** folder in the Linux box.

    The **safenet.te file** can be found at - https://kb.safenet-inc.com/resources/sites/SAFENET/content/staging/TECH_NOTES/1000/TE1821/en_US/1.1/safenet.te.

15. Log in as a root user.

16. To compile the policy file (**safenet.te**), run the following commands:

    checkmodule -M -m -o /tmp/safenet.mod /tmp/safenet.te

    semodule_package -m /tmp/safenet.mod -o /tmp/safenet.pp

17. To install the policy module, run the following command:

    semodule -i /tmp/safenet.pp

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.	