SafeNet Authentication Client

Integration Guide

Using SafeNet Authentication Client CBA for Red Hat Enterprise Linux Workstation

security to be free

All information herein is either public information or is the property of and owned solely by Gemalto and/or its subsidiaries who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Gemalto's information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

- The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided “AS IS” without any warranty of any kind. Unless otherwise expressly agreed in writing, Gemalto makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Gemalto reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Gemalto 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 Gemalto 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.

Gemalto 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 Gemalto be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Gemalto products. Gemalto 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.

Document Number: 007-000117-001, Rev. A

Release Date: July 2018

Contents

Third-Party Software Acknowledgement

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

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. Gemalto's certificate-based tokens and smart cards 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.

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 Workstation using Gemalto's tokens and smart cards.

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

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

Applicability

The information in this document applies to:

- **SafeNet Authentication Client (SAC) Typical installation mode**— SafeNet Authentication Client is public key infrastructure (PKI) middleware that manages Gemalto's tokens and smart cards.

  For more details about different SAC installation modes, refer to the Customization section in the SafeNet Authentication Client Administrator Guide.
- **Red Hat Enterprise Linux Workstation**

Environment

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

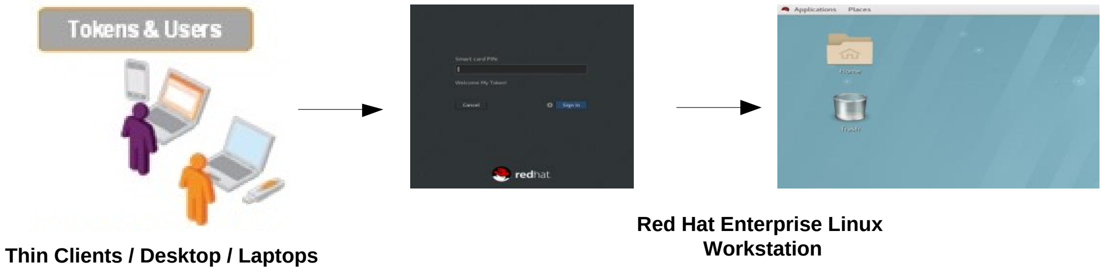
- **SafeNet Authentication Client (SAC)** -10.0.60
- **Red Hat Enterprise Linux Workstation -** 7.5

Audience

This document is intended for system administrators who are familiar with Red Hat Enterprise Linux Workstation, and are interested in adding multi-factor authentication capabilities using SafeNet tokens.

CBA Flow using SafeNet Authentication Client

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


Tokens & Users
Thin Clients / Desktop / Laptops
redhat
Applications
Places
Red Hat Enterprise Linux Workstation

1. A user attempts to log on to Red Hat Enterprise Linux Workstation using the logon manager.
2. The user inserts the SafeNet token on which his certificate resides, and, when prompted, enters the token password.
3. A successful smart card logon 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 Workstation using Gemalto tokens and smart cards:

- To use CBA, the Microsoft Enterprise Certificate Authority must be installed and configured. In general, any CA can be used. However, in this guide, integration is demonstrated using Microsoft CA.
- If SAM is used to manage the tokens, Token Policy Object (TPO) should be configured with MS CA Connector. For further details, refer to the section “Connector for Microsoft CA” in the *SafeNet Authentication Manager Administrator’s Guide*.
- Users must have a Gemalto token or smart card enrolled with an appropriate certificate.
- RHEL 7.5 Workstation x64 computer (not in the domain) Installed with Smart Card Support Add on.
- SafeNet Authentication Client (SAC) 10.0.60 Post GA installed on the RHEL 7.5 x64 computer.
  For more details see “Appendix B: Installation of SafeNet Authentication Client (SAC) on Linux” on page 16.
- A PAM_PKCS11 module installed on the RHEL computer (pam_pkcs11.i686 0:0.6.2-28.el7).
- Authconfig-gtk.x86_64 installed on the RHEL computer (authconfig-gtk.x86_64 0:6.2.8-30.el7).
- An X.509 Smart Card User certificate Enrolled to SafeNet eToken/IDPrime and a Microsoft root CA certificate installed locally on the RHEL computer.
- Configuration of the user(s) to be authenticated (on the RHEL 7.5 x64 computer)

Supported Tokens and Smart Cards in SafeNet Authentication Client

SafeNet Authentication Client 10.0.60 supports the following tokens and smart cards:

Certificate-based USB tokens

- SafeNet eToken 5110 GA
- SafeNet eToken 5110 FIPS
- SafeNet eToken 5110 CC

Smart Cards

- Gemalto IDPrime MD 830
- Gemalto IDPrime MD 840
- Gemalto IDCore 30B eToken (not supported in Linux with CT-40 reader)

For a full list of supported devices, refer to SafeNet Authentication Client Customer Release Notes.

Configuring Red Hat Enterprise Linux Workstation

Installing 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/RootCA.cer

NOTE: 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

Adding the eToken Module to the NSS Database

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

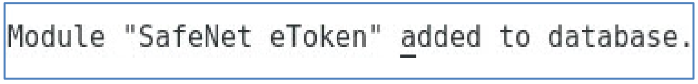
-add <moduleName> -libfile <library File> -dbdir <dbFolder>

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.
```

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

Configuring the PAM-PKCS11 Module

The PAM-PKCS11 module uses the /etc/pam_pkcs11 directory for configuration.
For more information see “Appendix A: PAM-pkcs11 Configuration Files (Reference)” on page 14**.**

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>
    ```

2. Create hash links to the 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**).

3. 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;
    }
    ```

4. Open the pam-pkcs11 configuration file in **pam_pkcs11** section **/etc/pam_pkcs11/pam_pkcs11.conf** file.

    In **use_ pkcs11_ module** row update the added **SafeNet pkcs11** module named **eToken**.

    Example: **use_pkcs11_module = eToken**

Configuring Mappers Using Certificate CN as a Mapper

1. 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**).

   In this example Certificate Common Name (CN) is demonstrated to use Certificate CN as a mapper, make the following modifications to the mapper cn section:

   Scroll down in the **pam_pkcs11.conf** file until you see:

   ```
   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";
   }
   ```

   Verify that default is: **mapfile = file:///etc/pam_pkcs11/cn_map;**

   For more information see “Appendix A: PAM-pkcs11 Configuration Files (Reference)” on page 14.

2. Set up and create “**cn_map**” mapping file, **Create the map file** “**cn_map**” in the default path 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.

   Example: Users -> bob

NOTE:

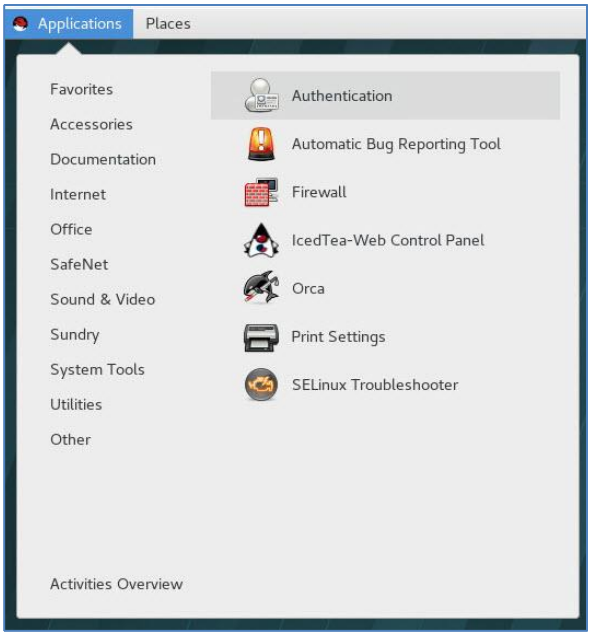
To check the map certificates to a user:
Insert a smart card into the smart card reader, and then run the following command:
# pklogin_finder debug

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

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

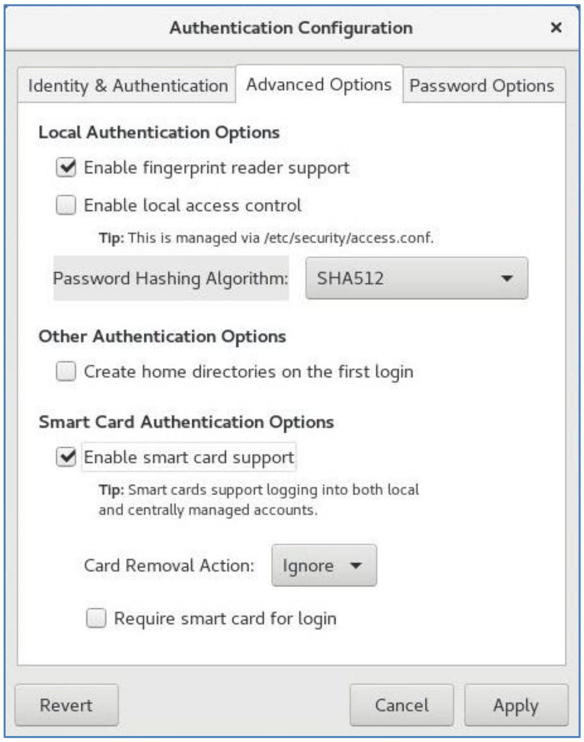
Configuring Graphical Login

1. Click **Applications > Sundry > Authentication**.



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

2. In the **Advanced Options** tab, select **Enable smart card support** and click **Apply**.



Authentication Configuration ×

Identity & Authentication | Advanced Options | Password Options

Local Authentication Options

☑ Enable fingerprint reader support

☐ Enable local access control

Tip: This is managed via /etc/security/access.conf.

Password Hashing Algorithm: SHA512

Other Authentication Options

☐ Create home directories on the first login

Smart Card Authentication Options

☑ Enable smart card support

Tip: Smart cards support logging into both local and centrally managed accounts.

Card Removal Action: Ignore

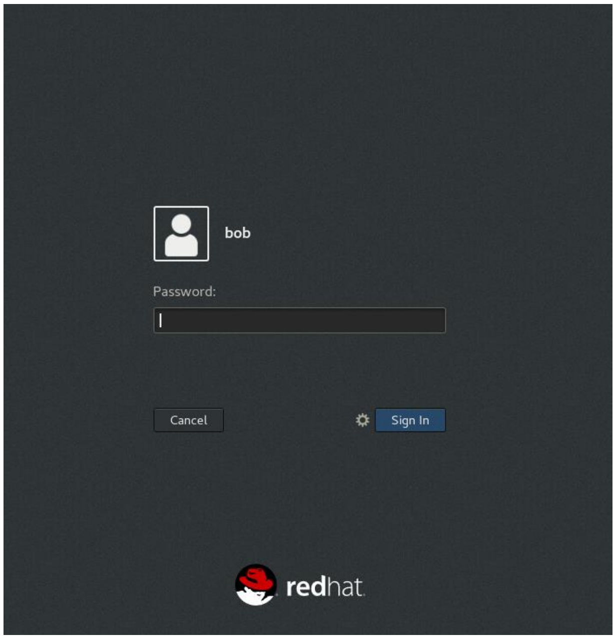
☐ Require smart card for login

Revert | Cancel | Apply

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

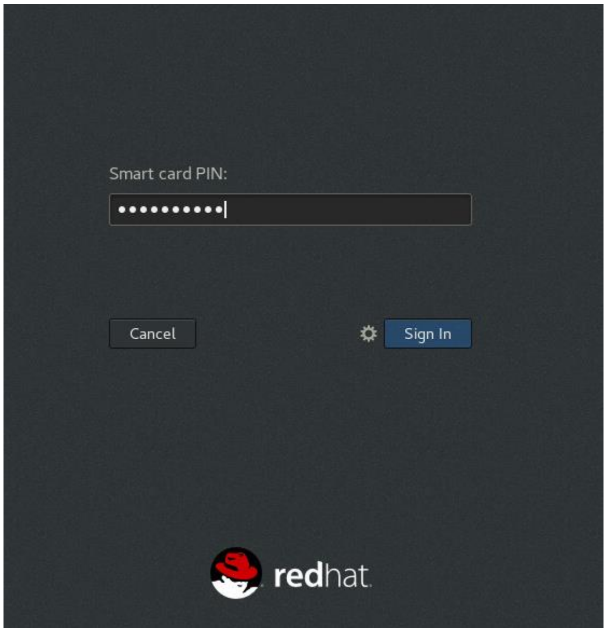
Running the Solution

1. The Gnome Classic logon window is displayed



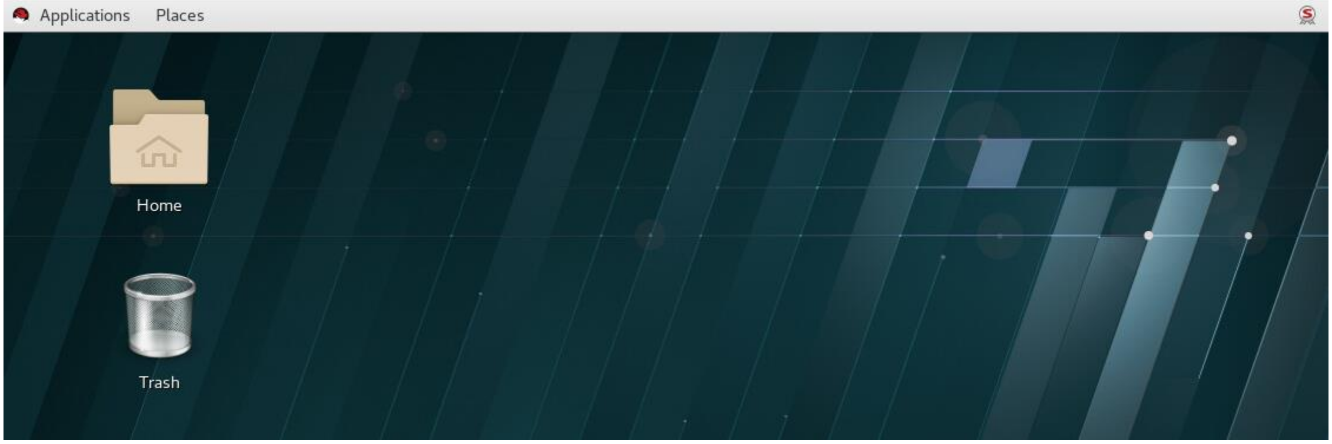
(The screen image above is from Red Hat. Trademarks are the property of their respective owners.)

2. Connect the Smart Card/eToken on which the certificate resides, and, when prompted, enter the smart card/eToken PIN and click **Sign in.**



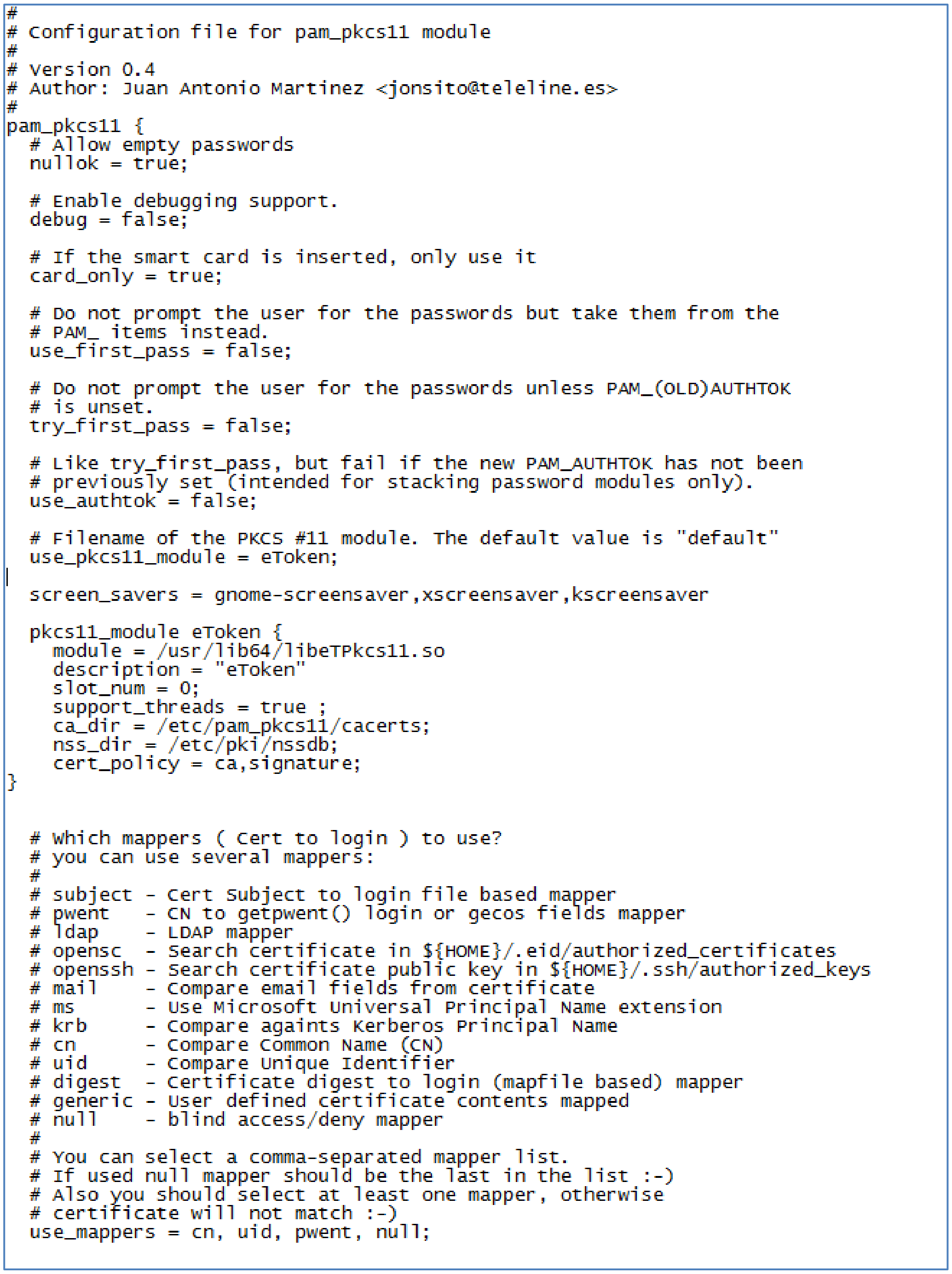
(The screen image above is from Red Hat. Trademarks are the property of their respective owners.)

The user is successfully logged in.



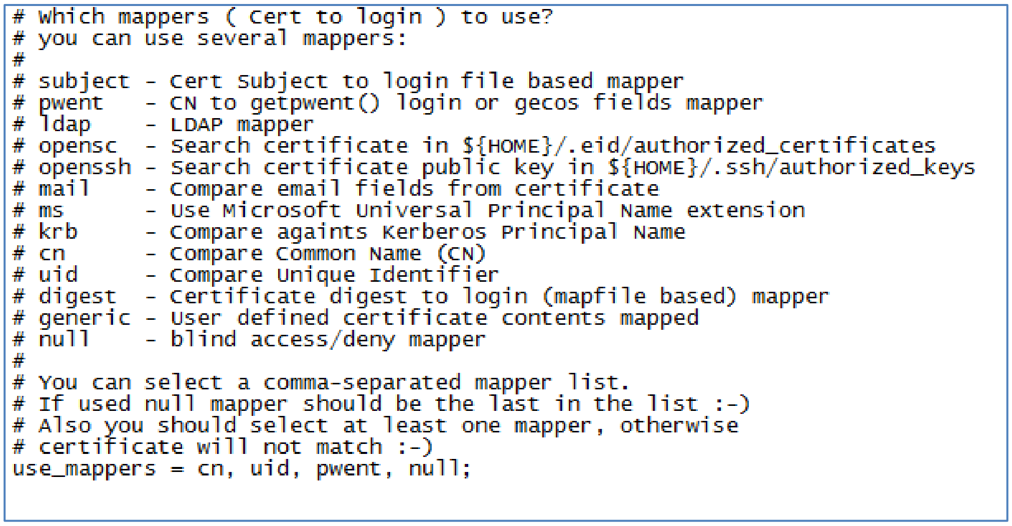
(The screen image above is from Red Hat. Trademarks are the property of their respective owners.)

Appendix A: PAM-pkcs11 Configuration Files (Reference)



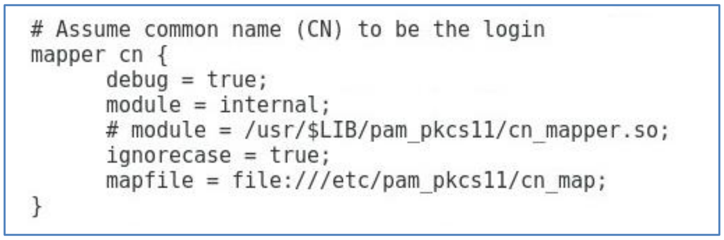
```
#
# Configuration file for pam_pkcs11 module
#
# Version 0.4
# Author: Juan Antonio Martinez <jonsito@teleline.es>
#
pam_pkcs11 {
  # Allow empty passwords
  nullok = true;

  # Enable debugging support.
  debug = false;

  # If the smart card is inserted, only use it
  card_only = true;

  # Do not prompt the user for the passwords but take them from the
  # PAM_ items instead.
  use_first_pass = false;

  # Do not prompt the user for the passwords unless PAM_(OLD)AUTHTOK
  # is unset.
  try_first_pass = false;

  # Like try_first_pass, but fail if the new PAM_AUTHTOK has not been
  # previously set (intended for stacking password modules only).
  use_authtok = false;

  # Filename of the PKCS #11 module. The default value is "default"
  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;
    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 = cn, uid, pwent, null;
```

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



```
# 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 = cn, uid, pwent, null;
```

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



```
# Assume common name (CN) to be the login
mapper cn {
      debug = true;
      module = internal;
      # module = /usr/$LIB/pam_pkcs11/cn_mapper.so;
      ignorecase = true;
      mapfile = file:///etc/pam_pkcs11/cn_map;
}
```

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

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

To install SAC Client on RHEL:

In the Linux Terminal window, run the following command:

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

If during installation, pcsc-lite dependency is required, install the pcsc-lite package on the RHEL 7.5 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:

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

   The **safenet.te file** can be found at **KB KB0017111** in - https://gemalto.service-now.com/csm?id=kb_article&sys_id=494c9974db44d344d298728dae9619ad

2. Log in as a root user.

3. 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

4. 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 Gemalto Customer Support. Gemalto 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 Gemalto 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	Gemalto 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 Gemalto Knowledge Base.	