Red Hat Enterprise Linux 8

管理智能卡验证

在 RHEL 中设置和管理智能卡验证

Last Updated: 2021-06-07

Red Hat Enterprise Linux 8 管理智能卡验证

在 RHEL 中设置和管理智能卡验证

法律通告

摘要

本文档集合提供了如何在 RHEL 中管理智能卡验证的说明。

目录

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始： master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。如需了解更多详细信息，请参阅 CTO Chris Wright 信息。

在身份管理中,计划使用的术语替换包括：

- ***块列表*** 替换 *黑名单*
- ***允许列表*** 替换 *白名单*
- ***从属(secondary)***替换 *slave*
- 根据上下文, *master* 将被更精确地替换为更精确的语言：
  - ***IdM 服务器*** 替换 *IdM master*
  - ***CA 续订服务器*** 替换 *CA 续订 master*
  - ***CRL publisher 服务器*** 替换 *CRL master*
  - ***多供应商*** 替换了 *multi-master*

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。要做到这一点：

- 关于特定内容的简单评论：
  1. 请确定您使用 *Multi-page HTML* 格式查看文档。另外，确定 **Feedback** 按钮出现在文档页的右上方。
  2. 用鼠标指针高亮显示您想评论的文本部分。
  3. 点在高亮文本上弹出的 **Add Feedback**。
  4. 按照显示的步骤操作。
- 要提交更复杂的反馈，请创建一个 Bugzilla ticket：
  1. 进入 Bugzilla 网站。
  2. 在 Component 中选择 **Documentation**。
  3. 在 **Description** 中输入您要提供的信息。包括文档相关部分的链接。
  4. 点 **Submit Bug**。

第 1 章 为智能卡验证配置身份管理

使用基于智能卡的验证是使用密码进行验证的替代选择。您可以以私钥和证书的形式在智能卡中存储用户凭证,并使用特殊软件和硬件访问它们。将智能卡放在读卡器或者 USB 端口中,并为智能卡提供 PIN 代码,而不是提供您的密码。

身份管理(IdM)支持智能卡验证：

- IdM 证书颁发机构发布的用户证书
- 外部证书颁发机构发布的用户证书

这个用户用例演示了如何在 IdM 中为两种类型的证书设置智能卡验证。在用户问题中,**smartcard_ca.pem** CA 证书是包含可信外部证书颁发机构证书的文件。

用户会包括以下模块：

- 第 1.1 节 “为智能卡验证配置 IdM 服务器”
- 第 1.2 节 “为智能卡验证配置 IdM 客户端”
- 第 1.3 节 “在 IdM 中的用户条目中添加证书”
- 第 1.4 节 “安装用来管理和使用智能卡的工具”
- 第 1.5 节 “在智能卡中存储证书”
- 第 1.6 节 “使用智能卡登录到 IdM”
- 第 1.7 节 “使用智能卡验证配置 GDM 访问”
- 第 1.8 节 “使用智能卡验证配置 su 访问”

1.1. 为智能卡验证配置 IDM 服务器

如果要为由 **EXAMPLE.ORG** 域的证书颁发机构发布的证书颁发机构(DN)为 **CN=Certificate Authority,DC=EXAMPLE,DC=ORG** 启用智能卡验证,那么您需要获取颁发机构的证书,以便您可以使用配置 IdM 服务器的脚本运行它。例如，您可以从认证机构发布的证书的网页下载证书。详情请查看 配置浏览器以启用证书验证的步骤 1 到 4a。

要为 IdM 证书颁发机构发布的 IdM 用户启用智能卡验证,请从运行 IdM CA 的 IdM 服务器上的 **/etc/ipa/ca.crt** 文件中获取 CA 证书。

这部分论述了如何为智能卡验证配置 IdM 服务器。首先,以 PEM 格式获取带有 CA 证书的文件,然后运行内置 **ipa-advise** 脚本。最后，重新载入系统配置。

先决条件

- 有到 IdM 服务器的 root 访问权限。
- 您有 root CA 证书和任何子 CA 证书。

流程

1. 创建要进行配置的目录：

```
[root@server]# mkdir ~/SmartCard/
```

2. 进入该目录：

```
[root@server]# cd ~/SmartCard/
```

3. 获取存储在 PEM 格式文件中的相关 CA 证书。如果您的 CA 证书存储在不同格式的文件,如 DER,将其转换为 PEM 格式。IdM 证书颁发机构证书位于 **/etc/ipa/ca.crt** 文件中。
将 DER 文件转换为 PEM 文件：

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

4. 为方便起见，将证书复制到您要进行配置的目录中：

```
[root@server SmartCard]# cp /etc/ipa/ca.crt ~/SmartCard/
[root@server SmartCard]# cp /tmp/smartcard_ca.pem ~/SmartCard/
```

5. 另外,如果您使用外部证书颁发机构证书,使用 **openssl x509** 实用程序以 **PEM** 格式查看文件的内容,以检查 **Issuer** 和 **Subject** 值是否正确：

```
[root@server SmartCard]# openssl x509 -noout -text -in smartcard_ca.pem | more
```

6. 使用管理员的权限生成带有内置 **ipa-advise** 工具的配置脚本：

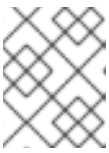
```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# sudo ipa-advise config-server-for-smart-card-auth > config-server-for-smart-card-auth.sh
```

config-server-for-smart-card-auth.sh 脚本执行以下操作：

- 它配置 IdM Apache HTTP 服务器。
- 它在 KDC（Key Distribution Center）中启用 PKINIT（Public Key Cryptography for Initial Authentication in Kerberos）。
- 它将 IdM Web UI 配置为接受智能卡授权请求。

7. 执行脚本,添加包含 root CA 和子 CA 证书的 PEM 文件,作为参数：

```
[root@server SmartCard]# chmod +x config-server-for-smart-card-auth.sh
[root@server SmartCard]# ./config-server-for-smart-card-auth.sh smartcard_ca.pem ca.crt
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



注意

确保您在任何子 CA 证书前添加 root CA 证书作为参数,并确保 CA 或子 CA 证书没有过期。

8. 另外,如果发布用户证书的证书颁发机构不提供任何在线证书状态协议(OCSP)响应程序,您可能需要禁用 OCSP 检查向 IdM Web UI 进行身份验证：

   a. 在 **/etc/httpd/conf.d/ssl.conf** 文件中将 **SSLOCSPEnable** 参数设置为 **off**:

   ```
   SSLOCSPEnable off
   ```

   b. 重启 Apache 守护进程(httpd)以使更改立即生效：

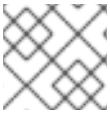
   ```
   [root@server SmartCard]# sudo systemctl restart httpd
   ```

警告

如果您只使用 IdM CA 发出的用户证书，不要禁用 OCSP 检查。OCSP 响应器是 IdM 的一部分。

有关如何使 OCSP 检查启用,并在不包含发布用户证书的 CA 侦听 OCSP 服务请求的位置的信息,请参阅 Apache mod_ssl 配置选项 中的 **SSLOCSPDefaultResponder** 指令。

该服务器现在被配置为智能卡验证。



注意

要在整个拓扑中启用智能卡验证，请在每个 IdM 服务器中运行操作过程。

1.2. 为智能卡验证配置 IDM 客户端

这部分论述了如何为智能卡验证配置 IdM 客户端。这个步骤需要在您要连接的每个 IdM 系统、客户端或服务器中运行,同时使用智能卡进行验证。例如：要启用从主机 A 到主机 B 的 **ssh** 连接,该脚本需要在主机 B 上运行。

作为管理员,使用这个步骤启用智能卡验证

- **ssh** 协议
  详情请查看 使用智能卡验证配置 SSH 访问。
- 控制台登录
- Gnome Display Manager(GDM)
- **su** 命令

在向 IdM Web UI 进行身份验证时不需要此步骤。对 IdM Web UI 进行身份验证涉及两个主机,不需要是 IdM 客户端：

- 浏览器运行的机器（可能不在 IdM 域之外）
- **httpd** 在其中运行的 IdM 服务器

以下流程假设您要在 IdM 客户端而不是 IdM 服务器中配置智能卡验证。因此,您需要两台计算机：一个 IdM 服务器来生成配置脚本,另一个是运行该脚本的 IdM 客户端。

先决条件

- 为智能卡验证配置了您的 IdM 服务器，如 第 1.1 节 “为智能卡验证配置 IdM 服务器” 所述。
- 有对 IdM 服务器和 IdM 客户端的 root 访问权限。
- 您有 root CA 证书和任何子 CA 证书。

流程

1. 在 IdM 服务器中,使用管理员的权限生成带有 **ipa-advise** 的配置脚本：

```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# ipa-advise config-client-for-smart-card-auth > config-client-for-smart-card-auth.sh
```

config-client-for-smart-card-auth.sh 脚本执行以下操作：

- 它配置智能卡守护进程。
- 它设置系统范围信任存储。
- 它配置系统安全性服务守护进程(SSSD),以允许智能卡登录到桌面。

2. 从 IdM 服务器中，将脚本复制到 IdM 客户端机器中选择的目录中：

```
[root@server SmartCard]# scp config-client-for-smart-card-auth.sh root@client.idm.example.com:/root/SmartCard/
Password:
config-client-for-smart-card-auth.sh        100%  2419     3.5MB/s   00:00
```

3. 从 IdM 服务器中,将 PEM 格式的 CA 证书文件复制到 IdM 客户端机器上的同一目录中,如上一步中使用的：

```
[root@server SmartCard]# scp {smartcard_ca.pem,ca.crt} root@client.idm.example.com:/root/SmartCard/
Password:
smartcard_ca.pem                 100%  1237     9.6KB/s   00:00
ca.crt                           100%  2514    19.6KB/s   00:00
```

4. 在客户端机器上执行脚本,将包含 CA 证书的 PEM 文件添加为参数：

```
[root@client SmartCard]# kinit admin
[root@client SmartCard]# chmod +x config-client-for-smart-card-auth.sh
[root@client SmartCard]# ./config-client-for-smart-card-auth.sh smartcard_ca.pem ca.crt
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```

注意

确保您在任何子 CA 证书前添加 root CA 证书作为参数,并确保 CA 或子 CA 证书没有过期。

现在为智能卡验证配置了客户端。

1.3. 在 IDM 中的用户条目中添加证书

这个步骤描述了如何在 IdM 的用户条目中添加外部证书。

也可以将证书映射数据上传到 IdM 中的用户条目，而不必上传整个证书。包含完整证书或证书映射数据的用户条目可与对应的证书映射规则一同使用,以便为系统管理员配置智能卡验证。详情请参阅有关在 智能卡上配置身份验证的证书映射规则。

注意

如果用户的证书由 IdM 证书颁发机构发布,该证书已经存储在用户条目中,您可以跳过本节。

1.3.1. 在 IdM Web UI 的用户条目中添加证书

先决条件

- 您有要添加到用户条目的证书。

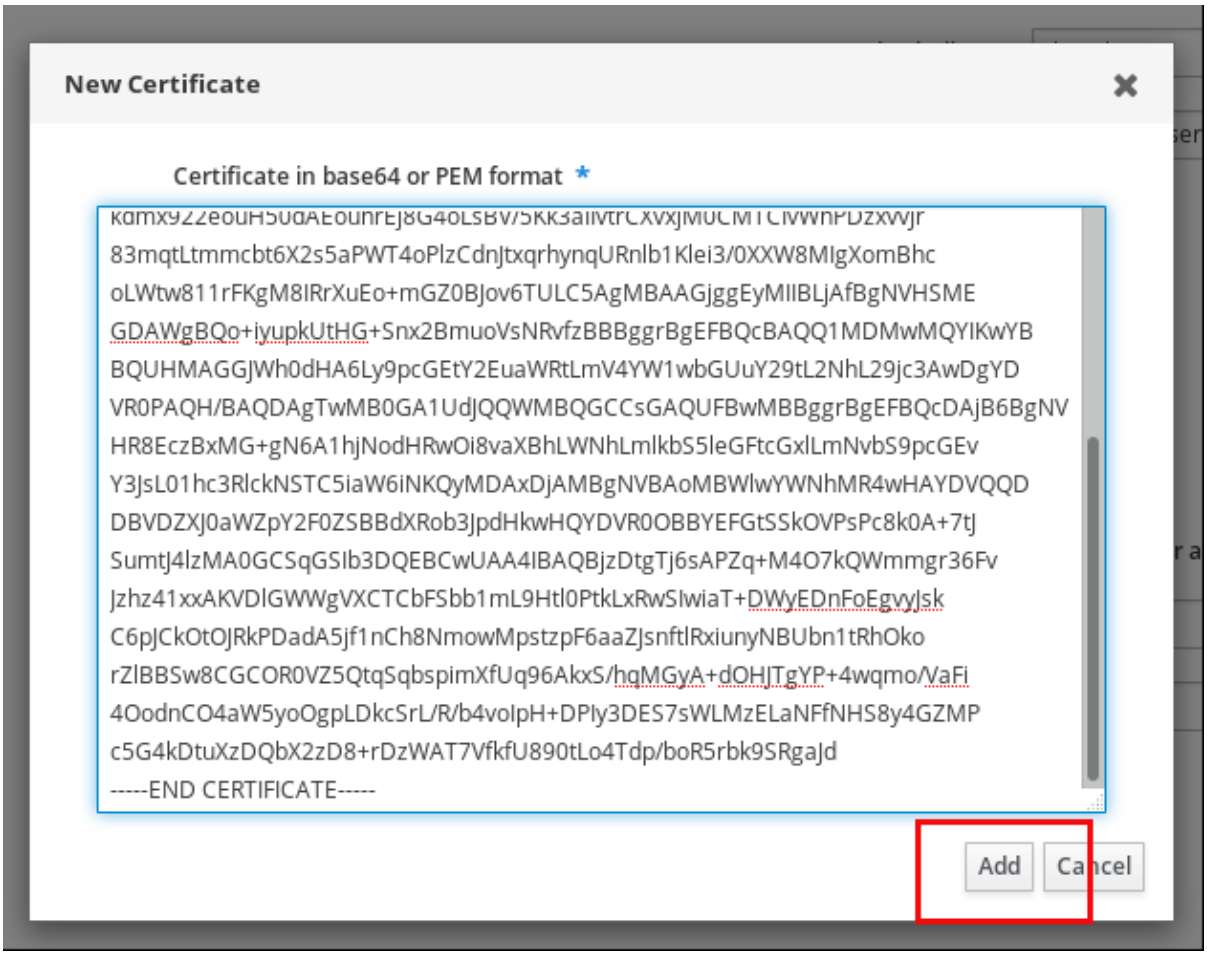
流程

1. 如果要向另一个用户添加证书,请以管理员身份登录到 IdM Web UI。要在您自己的配置集中添加证书,您不需要管理员的凭证。
2. 导航到 **Users → Active users → sc_user**。
3. 找到 **Certificate** 选项并点击 **Add**。
4. 在 **Command-Line Interface** 中,使用 **cat** 实用程序或文本编辑器以 **PEM** 格式显示证书：

```
[user@client SmartCard]$ cat testuser.crt
```

5. 将证书从 CLI 复制并粘贴到 Web UI 中打开的窗口。
6. 点 **Add**。

图 1.1. 在 IdM Web UI 中添加新证书



sc_user 条目现在包含外部证书。

1.3.2. 在 IdM CLI 的用户条目中添加证书

先决条件

- 您有要添加到用户条目的证书。

流程

1. 如果要向另一个用户添加证书,请以管理员身份登录到 IdM CLI:

```
[user@client SmartCard]$ kinit admin
```

要在您自己的配置集中添加证书,您不需要管理员的凭证：

```
[user@client SmartCard]$ kinit sc_user
```

2. 创建包含带有标头和内存器的证书的环境变量,并将其合并成一行,这是 **ipa user-add-cert** 命令期望的格式：

```
[user@client SmartCard]$ export CERT=`openssl x509 -outform der -in testuser.crt | base64 -w0 -`
```

请注意 **testuser.crt** 文件中的证书必须采用 **PEM** 格式。

3. 使用 **ipa user-add-cert** 命令将证书添加到 sc_user 的配置集中：

```
[user@client SmartCard]$ ipa user-add-cert sc_user --certificate=$CERT
```

sc_user 条目现在包含外部证书。

1.4. 安装用来管理和使用智能卡的工具

要配置智能卡，您需要一些工具来生成证书并将其保存在智能卡中。

您必须：

- 安装可帮助您管理证书的 **gnutls-utils** 软件包。
- 安装 **opensc** 软件包，它提供一组库和工具来使用智能卡。
- 启动与智能卡读取器沟通的 **pcscd** 服务。

流程

1. 安装 **opensc** 和 **gnutls-utils** 软件包：

```
# dnf -y install opensc gnutls-utils
```

2. 启动 **pcscd** 服务。

```
# systemctl start pcscd
```

验证 **pcscd** 服务是否正在运行。

1.5. 在智能卡中存储证书

本节论述了使用 **pkcs15-init** 工具配置智能卡，帮助您配置：

- 擦除智能卡
- 设置新的 PIN 和可选的 PIN Unblocking Keys（PUKs）
- 在智能卡上创建新插槽
- 在插槽存储证书、私钥和公钥
- 锁定智能卡设置（有些智能卡需要这种类型）

先决条件

- 已安装 **opensc** 软件包，其中包括 **pkcs15-init** 工具。
  详情请查看安装用于管理和使用智能卡的工具。
- 该卡插入读卡器并连接到计算机。
- 您有可保存在智能卡中的私钥、公钥和证书。在此过程中，**testuser.key**、**testuserpublic.key** 和 **testuser.crt** 是用于私钥、公钥和证书的名称。

- 您当前的智能卡用户 PIN 和 Security Officer PIN（SO-PIN）

流程

1. 擦除智能卡并使用您的 PIN 验证自己：

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

这个卡已经被清除。

2. 初始化智能卡，设置您的用户 PIN 和 PUK，以及您的安全响应 PIN 和 PUK：

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
   --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

pcks15-init 工具在智能卡上创建新插槽。

3. 为插槽设置标签和验证 ID：

```
$ pkcs15-init --store-pin --label testuser \
   --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

该标签设置为人类可读值，本例中为 **testuser**。**auth-id** 必须是两个十六进制值，在本例中将其设置为 **01**。

4. 在智能卡的新插槽中存储并标记私钥：

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
   --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```

注意

在存储您的私钥和证书时，您为 **--id** 指定的值必须相同。如果没有为 **--id** 指定值，则更复杂的值由工具计算，因此更易于定义您自己的值。

5. 在智能卡上的新插槽中存储并标记该证书：

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \
   --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6. （可选）在智能卡上新插槽中保存并标记公钥：

```
$ pkcs15-init --store-public-key testuserpublic.key
   --label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```

注意

如果公钥与私钥和/或证书对应，您应该指定与私钥和/或证书相同的 ID。

7. （可选）有些智能卡要求您通过锁定设置来完善卡：

```
$ pkcs15-init -F
```

此时您的智能卡在新创建的插槽中包含证书、私钥和公钥。您还创建了您的用户 PIN 和 PUK，以及安全响应 PIN 和 PUK。

1.6. 使用智能卡登录到 IDM

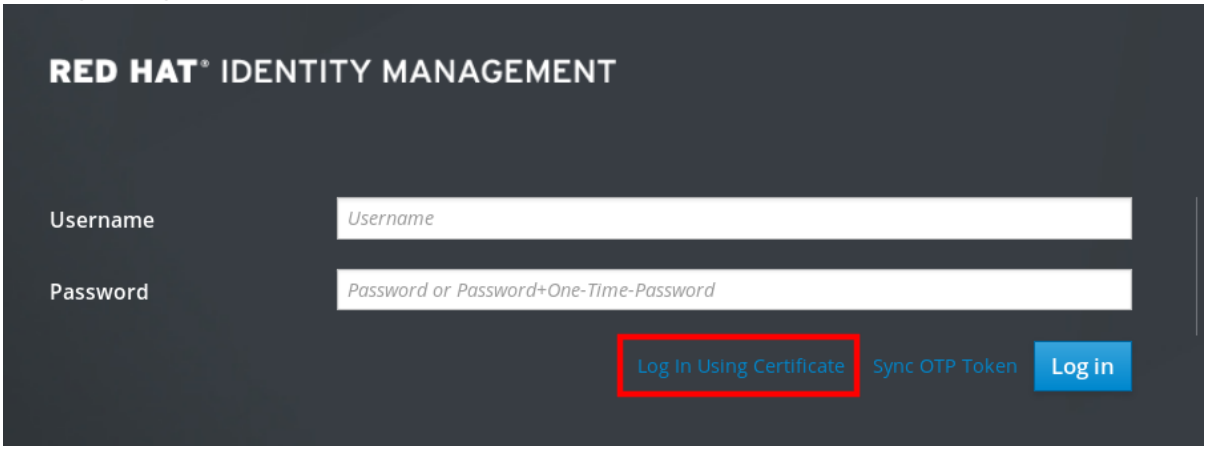
本节提供有关使用智能卡登录到 IdM Web UI 的信息。

先决条件

- web 浏览器被配置为使用智能卡验证。
- IdM 服务器已被配置为智能卡验证。
- IdM 服务器知道在智能卡中安装的证书。
- 您需要 PIN 解锁智能卡。
- 智能卡已插入到读取器中。

流程

1. 在浏览器中打开 IdM Web UI。
2. 点 **Log Insing Certificate**。

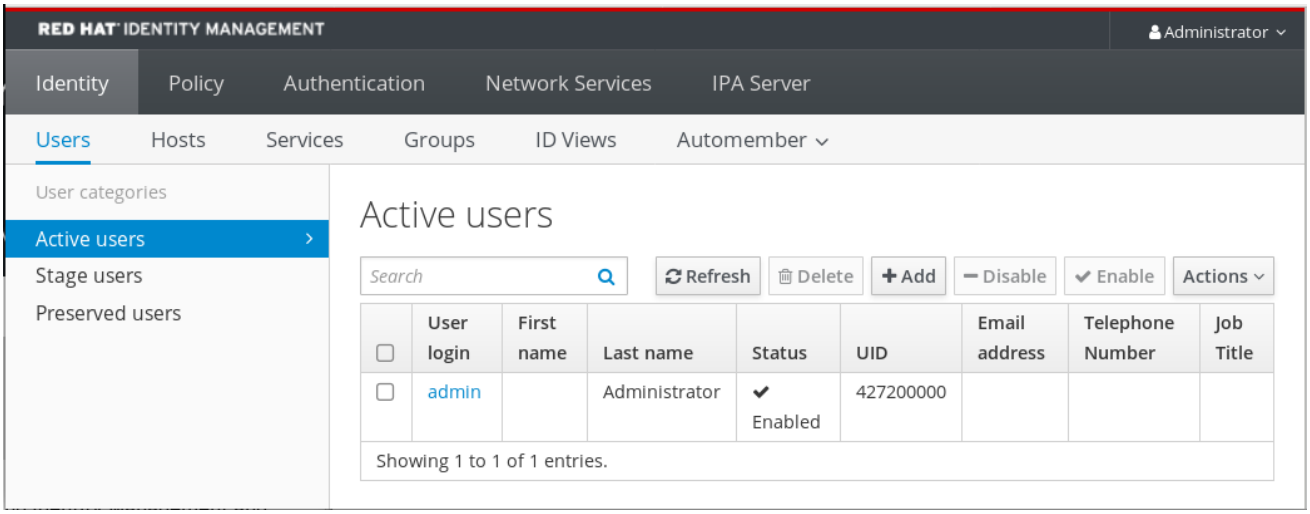


3. 如果打开了 **Password Required** 对话框,添加 PIN 来解锁智能卡并点击 **确定** 按钮。
此时会打开 **User Identification Request** 对话框。

   如果智能卡包含多个证书, **请在下面的下拉列表中选择要用于验证的证书。**

4. 点**确定**按钮。

现在,您可以成功登录到 IdM Web UI。



1.7. 使用智能卡验证配置 GDM 访问

Gnome Desktop Manager(GDM)需要身份验证。您可以使用您的密码,也可以使用智能卡进行验证。

这部分论述了访问 GDM 的智能卡验证。

使用智能卡验证的优点是,如果用户帐户是身份管理域的一部分,您也会获得一个 ticket-granting ticket(TGT)。

先决条件

- 该智能卡包含您的证书和私钥。
- 该用户帐户是 IdM 域的成员。
- 智能卡上的证书通过以下方式映射到用户条目：
  - 为特定用户条目分配证书。详情请查看, 第 1.3 节 “在 IdM 中的用户条目中添加证书”
  - 应用到该帐户的证书映射数据。详情请参阅有关在 智能卡上配置身份验证的证书映射规则。

流程

1. 在读取器中插入智能卡。
2. 输入智能卡 PIN。
3. 点 **Sign In**。

您已成功登录到 RHEL 系统,且您有由 IdM 服务器提供的 TGT。

验证步骤

- 在 **Terminal** 窗口中输入 **klist** 并检查结果：

```
$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: example.user@REDHAT.COM
```

```
Valid starting       Expires              Service principal
04/20/2020 13:58:24  04/20/2020 23:58:24  krbtgt/EXAMPLE.COM@EXAMPLE.COM
 renew until 04/27/2020 08:58:15
```

1.8. 使用智能卡验证配置 SU 访问

切换到其他用户需要身份验证。您可以使用密码或证书。这部分论述了使用您的智能卡来使用 **su** 命令。它表示,在输入 **su** 命令后会提示您输入智能卡 PIN。

先决条件

- 该智能卡包含您的证书和私钥。
- 该卡插入读卡器并连接到计算机。

流程

- 在终端窗口中,使用 **su** 命令切换到其他用户：

```
$ su - example.user
PIN for smart_card
```

如果配置成功，会提示您输入智能卡 PIN。

第 2 章 为 IDM 中智能卡验证配置 ADCS 发布的证书

这种情境描述了以下情况：

- 您的部署基于身份管理(IdM)和 Active Directory(AD)间的跨林信任。
- 您需要允许将其帐户存储在 AD 中的用户使用智能卡验证。
- 证书创建并存储在 Active Directory 证书服务(ADCS)中。

配置将按以下步骤完成：

- 将 CA 和用户证书从 Active Directory 复制到 IdM 服务器和客户端
- 使用 ADCS 证书为智能卡验证配置 IdM 服务器和客户端
- 转换 PFX(PKCS#12)文件,以便将证书和私钥保存到智能卡中
- 在 sssd.conf 文件中配置超时
- 为智能卡验证创建证书映射规则

先决条件

- 已安装身份管理(IdM)和 Active Directory(AD)信任
  详情请参阅在 IdM 和 AD 间安装信任。
- 已安装活跃目录证书服务(ADCS),并生成用户的证书

2.1. 智能卡验证

智能卡是一个物理设备，它可使用保存在卡中的证书提供个人验证。个人验证意味着，您可以象使用用户密码一样使用智能卡。

您可以以私钥和证书的形式在智能卡中存储用户凭证,并使用特殊软件和硬件访问它们。您将智能卡放在读卡器或者 USB 套接字中,并为智能卡提供 PIN 代码,而不是提供您的密码。

您可以配置在特定 IdM 客户端中希望智能卡验证如何工作：

- 用户可以使用用户名和密码或者智能卡进行身份验证
- 用户可以使用智能卡进行验证，并不允许使用密码进行验证
- 用户可以使用智能卡在删除时使用功能锁定来注销,且不允许密码

身份管理(IdM)支持智能卡验证：

- IdM 证书颁发机构发布的用户证书。详情请参阅 为智能卡验证配置身份管理。
- ADCS 证书颁发机构发布的用户证书。详情请参阅为 IdM 中智能卡验证配置 ADCS 发布的证书。
- RHEL 系统中生成的本地认证认证机构发布的用户证书。详情请参阅 配置和导入本地证书到智能卡。
- 由外部证书颁发机构发布的用户证书。

注意

如果要开始使用智能卡验证，请参阅RHEL8 中的硬件要求：智能卡支持。

2.2. 信任配置和证书使用量所需的 WINDOWS 服务器设置

本节总结在 Windows 服务器上必须配置的内容：

- 已安装 Active Directory 证书服务(ADCS)
- 创建证书颁发机构
- [可选] 如果您使用证书颁发机构 Web 注册,则必须配置互联网信息服务(IIS)

导出证书：

- 键必须有 **2048** 位或更多位
- 包括一个私钥
- 您需要一个以下格式的证书：个人信息交换 - **PKCS #12(.PFX)**
  - 启用证书隐私

2.3. 使用 SFTP 从 ACTIVE DIRECTORY 复制证书

为了可以使用智能卡验证功能,您需要复制以下证书文件：

- **CER** 格式的 root CA 证书：**adcs-winserver-ca.cer** 在您的 IdM 服务器中。
- 带有 **PFX** 格式的一个私有密钥的用户证书：**aduser1.pfx** 在 IdM 客户端中。

注意

这个过程预期 SSH 访问是允许的。如果 SSH 不可用,用户必须将文件从 AD 服务器复制到 IdM 服务器和客户端。

流程

1. 从 **IdM 服务器** 连接并将 **adcs-winserver-ca.cer** 根证书复制到 IdM 服务器中：

```
root@idmserver ~]# sftp Administrator@winserver.ad.example.com
Administrator@winserver.ad.example.com's password:
Connected to Administrator@winserver.ad.example.com.
sftp> cd <Path to certificates>
sftp> ls
adcs-winserver-ca.cer    aduser1.pfx
sftp>
sftp> get adcs-winserver-ca.cer
Fetching <Path to certificates>/adcs-winserver-ca.cer to adcs-winserver-ca.cer
<Path to certificates>/adcs-winserver-ca.cer                100%  1254    15KB/s 00:00
sftp quit
```

2. 从 **IdM 客户端** 连接并将 **aduser1.pfx** 用户证书复制到客户端：

```
[root@client1 ~]# sftp Administrator@winserver.ad.example.com
Administrator@winserver.ad.example.com's password:
Connected to Administrator@winserver.ad.example.com.
sftp> cd /<Path to certificates>
sftp> get aduser1.pfx
Fetching <Path to certificates>/aduser1.pfx to aduser1.pfx
<Path to certificates>/aduser1.pfx                100%  1254    15KB/s 00:00
sftp quit
```

现在,CA 证书存储在 IdM 服务器中,用户证书保存在客户端机器上。

2.4. 使用 ADCS 证书为智能卡验证配置 IDM 服务器和客户端

您必须配置 IdM(Identity Management)服务器和客户端以便在 IdM 环境中使用智能卡验证。IdM 包括进行所有必要的更改的 **ipa-advise** 脚本：

- 安装所需的软件包
- 它配置 IdM 服务器和客户端
- 将 CA 证书复制到预期的位置

您可以在 IdM 服务器中运行 **ipa-advise**。

这个步骤描述了：

- 在 IdM 服务器中：准备 **ipa-advise** 脚本来配置您的 IdM 服务器进行智能卡验证。
- 在 IdM 服务器中：准备 **ipa-advise** 脚本来配置您的 IdM 客户端进行智能卡验证。
- 在 IdM 服务器中：使用 AD 证书应用 IdM 服务器上的 **ipa-advise** 服务器脚本。
- 将客户端脚本移动到 IdM 客户端机器中。
- 在 IdM 客户端中：使用 AD 证书应用 IdM 客户端 **ipa-advise** 客户端脚本。

先决条件

- 证书已复制到 IdM 服务器。
- 获取 Kerberos 票据。
- 以具有管理权限的用户身份登录。

流程

1. 在 IdM 服务器中，使用 **ipa-advise** 脚本配置客户端：

```
[root@idmserver ~]# ipa-advise config-client-for-smart-card-auth > sc_client.sh
```

2. 在 IdM 服务器中，使用 **ipa-advise** 脚本配置服务器：

```
[root@idmserver ~]# ipa-advise config-server-for-smart-card-auth > sc_server.sh
```

3. 在 IdM 服务器中执行脚本：

```
[root@idmserver ~]# sh -x sc_server.sh adcs-winserver-ca.cer
```

- 它配置 IdM Apache HTTP 服务器。
- 它在 KDC（Key Distribution Center）中启用 PKINIT（Public Key Cryptography for Initial Authentication in Kerberos）。
- 它将 IdM Web UI 配置为接受智能卡授权请求。

4. 将 **sc_client.sh** 脚本复制到客户端系统中：

```
[root@idmserver ~]# scp sc_client.sh root@client1.idm.example.com:/root
Password:
sc_client.sh                  100%  2857   1.6MB/s   00:00
```

5. 将 Windows 证书复制到客户端系统中：

```
[root@idmserver ~]# scp adcs-winserver-ca.cer root@client1.idm.example.com:/root
Password:
adcs-winserver-ca.cer                 100%  1254   952.0KB/s   00:00
```

6. 在客户端系统中运行客户端脚本：

```
[root@idmclient1 ~]# sh -x sc_client.sh adcs-winserver-ca.cer
```

CA 证书以正确格式安装在 IdM 服务器和客户端系统中，下一步是将用户证书复制到智能卡本身。

2.5. 转换 PFX 文件

在将 PFX(PKCS#12)文件保存到智能卡前,您必须：

- 将文件转换为 PEM 格式
- 将私钥和证书提取到两个不同的文件

先决条件

- PFX 文件被复制到 IdM 客户端机器中。

流程

1. 在 IdM 客户端中，采用 PEM 格式：

```
[root@idmclient1 ~]# openssl pkcs12 -in aduser1.pfx -out aduser1_cert_only.pem -clcerts -
nodes
Enter Import Password:
```

2. 将密钥提取到单独的文件中：

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -nocerts -out adduser1.pem >
aduser1.key
```

3. 将公共证书提取到单独的文件中：

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -clcerts -nokeys -out
aduser1_cert_only.pem > aduser1.crt
```

此时，您可以将 **aduser1.key** 和 **aduser1.crt** 保存到智能卡中。

2.6. 安装用来管理和使用智能卡的工具

要配置智能卡，您需要一些工具来生成证书并将其保存在智能卡中。

您必须：

- 安装可帮助您管理证书的 **gnutls-utils** 软件包。
- 安装 **opensc** 软件包，它提供一组库和工具来使用智能卡。
- 启动与智能卡读取器沟通的 **pcscd** 服务。

流程

1. 安装 **opensc** 和 **gnutls-utils** 软件包：

```
# dnf -y install opensc gnutls-utils
```

2. 启动 **pcscd** 服务。

```
# systemctl start pcscd
```

验证 **pcscd** 服务是否正在运行。

2.7. 在智能卡中存储证书

本节论述了使用 **pkcs15-init** 工具配置智能卡，帮助您配置：

- 擦除智能卡
- 设置新的 PIN 和可选的 PIN Unblocking Keys（PUKs）
- 在智能卡上创建新插槽
- 在插槽存储证书、私钥和公钥
- 锁定智能卡设置（有些智能卡需要这种类型）

先决条件

- 已安装 **opensc** 软件包，其中包括 **pkcs15-init** 工具。
  详情请查看安装用于管理和使用智能卡的工具。
- 该卡插入读卡器并连接到计算机。
- 您有可保存在智能卡中的私钥、公钥和证书。在此过程中，**testuser.key**、**testuserpublic.key** 和 **testuser.crt** 是用于私钥、公钥和证书的名称。
- 您当前的智能卡用户 PIN 和 Security Officer PIN（SO-PIN）

流程

1. 擦除智能卡并使用您的 PIN 验证自己：

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

这个卡已经被清除。

2. 初始化智能卡，设置您的用户 PIN 和 PUK，以及您的安全响应 PIN 和 PUK：

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
   --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

pcks15-init 工具在智能卡上创建新插槽。

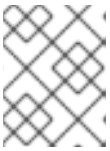
3. 为插槽设置标签和验证 ID：

```
$ pkcs15-init --store-pin --label testuser \
   --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

该标签设置为人类可读值，本例中为 **testuser**。**auth-id** 必须是两个十六进制值，在本例中将其设置为 **01**。

4. 在智能卡的新插槽中存储并标记私钥：

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
   --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



注意

在存储您的私钥和证书时，您为 **--id** 指定的值必须相同。如果没有为 **--id** 指定值，则更复杂的值由工具计算，因此更易于定义您自己的值。

5. 在智能卡上的新插槽中存储并标记该证书：

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \
   --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6. （可选）在智能卡上新插槽中保存并标记公钥：

```
$ pkcs15-init --store-public-key testuserpublic.key
   --label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```

注意

如果公钥与私钥和/或证书对应，您应该指定与私钥和/或证书相同的 ID。

7. （可选）有些智能卡要求您通过锁定设置来完善卡：

```
$ pkcs15-init -F
```

此时您的智能卡在新创建的插槽中包含证书、私钥和公钥。您还创建了您的用户 PIN 和 PUK，以及安全响应 PIN 和 PUK。

2.8. 在 SSSD.CONF 中配置超时

使用智能卡证书进行身份验证的时间可能比 SSSD 使用的默认超时时间更长。超时时间可能是由以下原因造成的：

- 阅读速度较慢
- 将物理设备转发到虚拟环境中
- 保存在智能卡中的证书太多
- 如果使用 OCSP 验证证书,则 OCSP（无线证书状态协议）响应程序响应较慢

在这种情况下，您可以在 **sssd.conf** 文件中延长以下超时时间，例如延长到 60 秒：

- **p11_child_timeout**
- **krb5_auth_timeout**

先决条件

- 您必须以 root 身份登录。

流程

1. 打开 **sssd.conf** 文件：

```
[root@idmclient1 ~]# vim /etc/sssd/sssd.conf
```

2. 更改 **p11_child_timeout** 的值：

```
[pam]
p11_child_timeout = 60
```

3. 更改 **krb5_auth_timeout** 的值：

```
[domain/IDM.EXAMPLE.COM]
krb5_auth_timeout = 60
```

4. 保存设置。

现在,与智能卡的交互可以在有超时验证失败前运行 1 分钟（60 秒）。

2.9. 为智能卡验证创建证书映射规则

如果要为在 AD(Active Directory)和 IdM(Identity Management)中有帐户的用户使用一个证书,您可以在 IdM 服务器上创建证书映射规则。

创建这样的规则后,用户就可以在两个域中使用智能卡进行验证。

有关证书映射规则的详情，请参阅用于在智能卡上配置身份验证的证书映射规则。

第 3 章 用于在智能卡上配置验证的证书映射规则

当身份管理(IdM)管理员无法访问某些用户的证书时,证书映射规则是方便的,允许用户使用证书进行身份验证。这种缺少访问权限通常是由于证书是由外部证书颁发机构发布的。一个特殊的用例由 IdM 域相互信任关系的 Active Directory(AD)证书系统发布的证书系统来表示。

如果 IdM 环境较大且有大量使用智能卡的用户，使用证书映射规则就会比较方便。在这种情况下，添加完整证书可能会比较复杂。在多数情况下,主体和签发者是可预测的,因此比完整证书更易于在时间之前添加。作为系统管理员,您可以创建证书映射规则,并将证书映射数据添加到用户条目中,即使在向特定用户签发证书前。证书发布后,即使完整证书还没有上传到用户条目,用户也可以使用证书登录。

另外,因为证书必须定期续订,证书映射规则会降低管理开销。当用户的证书更新时,管理员不必更新用户条目。例如：如果映射基于 **Subject** 和 **Issuer** 值,且新证书的主体和签发者与旧证书有相同的主题和签发者,则映射仍然被应用。如果使用完整证书，则管理员必须将新证书上传到用户条目以替换旧证书。

设置证书映射：

1. 管理员必须将证书映射数据（通常为签发者及主体）或者完整证书加载到用户帐户中。
2. 管理员必须创建证书映射规则，允许用户成功登录到 IdM
   a. 其帐户包含证书映射数据条目
   b. 哪个证书映射数据条目与证书的信息匹配

   有关组成映射规则以及如何获取和使用它们的各个组件的详情,请参阅 IdM 中的身份映射规则组件,并 从证书获取签发者以便用于匹配规则。

之后,当最终用户显示证书并保存在 文件系统 或 智能卡 中时,验证将成功。

3.1. 使用 ACTIVE DIRECTORY 域信任的证书映射规则

本节概述了当 IdM 部署与 Active Directory(AD)域处于信任关系时,可以使用的不同证书映射用例。

证书映射规则是为拥有由可信 AD 证书系统发布的智能卡证书的用户启用对 IdM 资源的访问的便捷方式。根据 AD 配置，可能会出现以下情况：

- 如果证书由 AD 发布,但用户和证书存储在 IdM 中,则验证请求的映射和整个处理都会在 IdM 端进行。有关配置此情境的详情，请参阅为存储在 IdM 中的用户配置证书映射
- 如果用户存储在 AD 中,身份验证请求的处理会在 AD 中进行。有三个不同的子案例：
  - AD 用户条目包含整个证书。有关在这种情况下配置 IdM 的详情，请参考为 AD 用户条目包含整个证书的用户配置证书映射。
  - AD 配置为将用户证书映射到用户帐户。在这种情况下,AD 用户条目不包含整个证书。而是包含名为 **altSecurityIdentities** 的属性。有关如何在这种场景中配置 IdM 的详情，请参阅在将 AD 配置为将用户证书映射到用户帐户时配置证书 映射。
  - AD 用户条目不包含整个证书和映射数据。在这种情况下,唯一解决方案是使用 **ipa idoverrideuser-add** 命令将整个证书添加到 IdM 中的 AD 用户的 ID 覆盖中。详情请参阅 如果 AD 用户条目没有证书或映射数据,则需要配置证书映射。

3.2. IDM 中身份映射规则的组件

本节论述了 IdM 中 *身份映射规则* 的组件以及如何配置它们。每个组件都有一个可覆盖的默认值。您可以在 Web UI 或 CLI 中定义这些组件。在 CLI 中，身份映射规则是使用 **ipa certmaprule-add** 命令创建的。

映射规则

映射规则组件将证书与一个或多个用户帐户关联（或 *映射*）。该规则定义了一个 LDAP 搜索过滤器,该过滤器将证书与预期的用户帐户相关联。
不同证书颁发机构(CA)发布的证书可能具有不同的属性,并可能会在不同域中使用。因此,IdM 不会无条件应用映射规则,而只适用于适当的证书。适当的证书使用 *匹配的规则* 定义。

请注意,如果您将映射规则选项留空,则会将证书作为 DER 编码二进制文件在 **userCertificate** 属性中搜索。

在 CLI 中使用 **--maprule** 选项定义映射规则。

匹配规则

匹配的规则组件选择您要应用映射规则的证书。默认匹配规则与证书与 **digitalSignature key** 使用量和 **clientAuth extended key** 使用量匹配。
在 CLI 中使用 **--matchrule** 选项定义匹配的规则。

域列表

域列表指定您希望 IdM 在处理身份映射规则时搜索用户的身份域。如果您没有指定该选项,IdM 只会在 IdM 客户端所属的本地域中搜索用户。
使用 **--domain** 选项在 CLI 中定义域。

优先级

当多个规则适用于某个证书时,具有最高优先级的规则将具有优先权。所有其他规则将被忽略。

- 数字值越低，身份映射规则的优先级越高。例如，具有优先级 1 的规则的优先级高于优先级 2 的规则。
- 如果规则没有定义优先级值，它具有最低的优先级。

使用 **--priority** 选项在 CLI 中定义映射规则优先级。

证书映射规则示例

要使用 CLI 定义名为 **simple_rule** 的证书映射规则,该规则允许对 **Smart Card CA** En 的 **EXAMPLE.ORG** 发布的证书进行身份验证,只要该证书中的 **Subject** 上的 **certmapdata** 与 IdM 中用户帐户中的 条目匹配：

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'
```

3.3. 从匹配规则中使用的证书获取签发者

这个步骤描述了如何从证书获取签发者信息,以便您可以复制这些信息并将其粘贴到证书映射规则的匹配规则中。要获得匹配规则所需的签发者格式,请使用 **openssl x509** 实用程序。

先决条件

- 您有 **.pem** 或 **.crt** 格式的用户证书

流程

流程

1. 从证书获取用户信息。使用 **openssl x509** 证书显示和签名实用程序：

   - 用于防止编码请求版本输出的 **-noout** 选项
   - 输出签发者名称的 **-issuer** 选项
   - **-in** 选项指定要从中读取证书的输入文件名称
   - 带有 **RFC2253** 值的 **-nameopt** 选项首先显示带有最特定相对可分辨名称(RDN)的输出

   如果输入文件包含 Identity Management 证书,命令的输出显示签发者使用 **Organisation** 信息定义：

   ```
   # openssl x509 -noout -issuer -in idm_user.crt -nameopt RFC2253
   issuer=CN=Certificate Authority,O=REALM.EXAMPLE.COM
   ```

   如果输入文件包含 Active Directory 证书,命令输出显示签发者使用 **Domain Component** 信息定义：

   ```
   # openssl x509 -noout -issuer -in ad_user.crt -nameopt RFC2253
   issuer=CN=AD-WIN2012R2-CA,DC=AD,DC=EXAMPLE,DC=COM
   ```

2. 另外,要根据一个匹配规则在 CLI 中创建新的映射规则,该规则指定证书签发者必须是 **ad.example.com** 域的 **AD-WIN2012R2-CA**,证书上的内容必须与 IdM 中用户帐户中的 **certmapdata** 条目匹配：

   ```
   # ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=AD-WIN2012R2-
   CA,DC=AD,DC=EXAMPLE,DC=COM' --maprule '(ipacertmapdata=X509:<I>
   {issuer_dn!nss_x500}<S>{subject_dn!nss_x500})'
   ```

附加信息

有关匹配规则和映射规则支持的格式的详情,以及优先级和域字段的说明,请查看 **sss-certmap(5)** man page。

第 4 章 配置和导入本地证书到智能卡

本章描述了以下情况：

- 主机没有连接到某个域。
- 您需要在这个主机上使用智能卡进行验证。
- 您需要使用智能卡验证配置 SSH 访问。
- 您需要使用 **authselect** 配置智能卡。

使用以下配置来实现这种情况：

- 为希望使用智能卡进行身份验证的用户获取用户证书。证书应该由在域中使用的可信认证认证机构生成。
  如果您无法获得证书，您可以生成由本地证书颁发机构签名的用户证书用于测试。
- 在智能卡中保存证书和私钥。
- 为 SSH 访问配置智能卡验证。

重要

如果主机可以作为域的一部分，将主机添加到域中，并使用活动目录或者身份管理认证机构生成的证书。

有关如何为智能卡创建 IdM 证书的详情，请参考为智能卡验证配置身份管理。

先决条件

- 已安装 authselect
  authselect 工具在 Linux 主机中配置用户验证，您可以使用它配置智能卡验证参数。有关 authselect 的详情，请参考浏览 authselect。
- RHEL 8 支持的智能卡或者 USB 设备
  详情请参阅 RHEL8 中的智能卡支持。

4.1. 创建本地证书

本节论述了如何执行这些任务：

- 生成 OpenSSL 证书颁发机构
- 创建证书签名请求

警告

以下步骤仅用于测试目的。由本地自签名证书颁发机构生成的证书与使用 AD、IdM 或 RHCS 认证认证机构一样安全。即使主机不是域的一部分，您仍应使用企业认证认证机构生成的证书。

流程

1. 创建可生成证书的目录，例如：

```
# mkdir /tmp/ca
# cd /tmp/ca
```

2. 设置证书（将该文件复制到 **ca** 目录中您的命令行）：

```
cat > ca.cnf <<EOF
[ ca ]
default_ca = CA_default

[ CA_default ]
dir             = .
database        = \$dir/index.txt
new_certs_dir   = \$dir/newcerts

certificate     = \$dir/rootCA.crt
serial          = \$dir/serial
private_key     = \$dir/rootCA.key
RANDFILE        = \$dir/rand

default_days    = 365
default_crl_days = 30
default_md      = sha256

policy          = policy_any
email_in_dn     = no

name_opt        = ca_default
cert_opt        = ca_default
copy_extensions  = copy

[ usr_cert ]
authorityKeyIdentifier = keyid, issuer

[ v3_ca ]
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer:always
basicConstraints       = CA:true
keyUsage               = critical, digitalSignature, cRLSign, keyCertSign

[ policy_any ]
organizationName       = supplied
```

```
organizationalUnitName = supplied
commonName             = supplied
emailAddress           = optional

[ req ]
distinguished_name = req_distinguished_name
prompt             = no

[ req_distinguished_name ]
O  = Example
OU = Example Test
CN = Example Test CA
EOF
```

3. 创建以下目录：

```
# mkdir certs crl newcerts
```

4. 创建以下文件：

```
# touch index.txt crlnumber index.txt.attr
```

5. 在串行文件中写入数字 01:

```
# echo 01 > serial
```

该命令在串行文件中写入数字 01。它是证书的序列号。当这个 CA 发布一个新证书时这个数字会加一。

6. 创建一个 OpenSSL root CA 密钥：

```
# openssl genrsa -out rootCA.key 2048
```

7. 创建自签名 root 认证认证机构证书：

```
# openssl req -batch -config ca.cnf \
    -x509 -new -nodes -key rootCA.key -sha256 -days 10000 \
    -set_serial 0 -extensions v3_ca -out rootCA.crt
```

8. 为您的用户名创建密钥：

```
# openssl genrsa -out example.user.key 2048
```

这个密钥是在本地系统中生成的，因此当密钥保存在卡中时，从系统中删除密钥。

您还可以直接在智能卡中创建密钥。要做到这一点，请遵循智能卡生产商生成的说明。

9. 创建证书签名请求配置文件（将这个文本复制到 ca 目录中的命令行中）：

```
cat > req.cnf <<EOF
[ req ]
distinguished_name = req_distinguished_name
prompt = no
```

```
[ req_distinguished_name ]
O = Example
OU = Example Test
CN = testuser

[ req_exts ]
basicConstraints = CA:FALSE
nsCertType = client, email
nsComment = "testuser"
subjectKeyIdentifier = hash
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth, emailProtection, msSmartcardLogin
subjectAltName = otherName:msUPN;UTF8:testuser@EXAMPLE.COM,
email:testuser@example.com
EOF
```

10. 为 example.user 证书创建证书签名请求：

```
# openssl req -new -nodes -key example.user.key \
    -reqexts req_exts -config req.cnf -out example.user.csr
```

11. 配置新证书。过期期限设定为 1 年：

```
# openssl ca -config ca.cnf -batch -notext \
    -keyfile rootCA.key -in example.user.csr -days 365 \
    -extensions usr_cert -out example.user.crt
```

此时，认证颁发机构和证书被成功生成并准备好导入到智能卡。

4.2. 将证书复制到 SSSD 目录中

GNOME 桌面管理器(GDM)需要 SSSD。如果使用 GDM，则需要将 PEM 证书复制到 **/etc/sssd/pki** 目录中。

先决条件

- 已生成本地 CA 颁发机构和证书

流程

1. 确保已在系统中安装了 SSSD。

```
# rpm -q sssd
sssd-2.0.0.43.el8_0.3.x86_64
```

2. 创建 **/etc/sssd/pki** 目录：

```
# file /etc/sssd/pki
/etc/sssd/pki/: directory
```

3. 将 **rootCA.crt** 作为 PEM 文件复制到 **/etc/sssd/pki/** 目录中：

```
# cp /tmp/ca/rootCA.crt /etc/sssd/pki/sssd_auth_ca_db.pem
```

现在，您已成功生成了证书颁发机构和证书，并将它们保存到 **/etc/sssd/pki** 目录中。

注意

如果要与另一个应用程序共享证书颁发机构证书，您可以在 sssd.conf 中更改位置：

- SSSD PAM 响应者：**[pam]** 部分中的 **pam_cert_db_path**
- SSSD ssh 响应器：**[ssh]** 部分中的 **ca_db**

详情请查看 **sssd.conf** 的 man page。

红帽建议保留默认路径,并使用 SSSD 的专用证书颁发机构文件来确保这里只列出信任验证的证书颁发机构。

4.3. 安装用来管理和使用智能卡的工具

要配置智能卡，您需要一些工具来生成证书并将其保存在智能卡中。

您必须：

- 安装可帮助您管理证书的 **gnutls-utils** 软件包。
- 安装 **opensc** 软件包，它提供一组库和工具来使用智能卡。
- 启动与智能卡读取器沟通的 **pcscd** 服务。

流程

1. 安装 **opensc** 和 **gnutls-utils** 软件包：

```
# dnf -y install opensc gnutls-utils
```

2. 启动 **pcscd** 服务。

```
# systemctl start pcscd
```

验证 **pcscd** 服务是否正在运行。

4.4. 在智能卡中存储证书

本节论述了使用 **pkcs15-init** 工具配置智能卡，帮助您配置：

- 擦除智能卡
- 设置新的 PIN 和可选的 PIN Unblocking Keys（PUKs）
- 在智能卡上创建新插槽
- 在插槽存储证书、私钥和公钥
- 锁定智能卡设置（有些智能卡需要这种类型）

先决条件

- 已安装 **opensc** 软件包，其中包括 **pkcs15-init** 工具。
  详情请查看安装用于管理和使用智能卡的工具。
- 该卡插入读卡器并连接到计算机。
- 您有可保存在智能卡中的私钥、公钥和证书。在此过程中，**testuser.key**、**testuserpublic.key** 和 **testuser.crt** 是用于私钥、公钥和证书的名称。
- 您当前的智能卡用户 PIN 和 Security Officer PIN（SO-PIN）

流程

1. 擦除智能卡并使用您的 PIN 验证自己：

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

   这个卡已经被清除。

2. 初始化智能卡，设置您的用户 PIN 和 PUK，以及您的安全响应 PIN 和 PUK：

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
    --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

   pcks15-init 工具在智能卡上创建新插槽。

3. 为插槽设置标签和验证 ID：

```
$ pkcs15-init --store-pin --label testuser \
    --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

   该标签设置为人类可读值，本例中为 **testuser**。**auth-id** 必须是两个十六进制值，在本例中将其设置为 **01**。

4. 在智能卡的新插槽中存储并标记私钥：

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
    --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```

注意

在存储您的私钥和证书时，您为 **--id** 指定的值必须相同。如果没有为 **--id** 指定值，则更复杂的值由工具计算，因此更易于定义您自己的值。

5. 在智能卡上的新插槽中存储并标记该证书：

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \
    --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6. （可选）在智能卡上新插槽中保存并标记公钥：

```
$ pkcs15-init --store-public-key testuserpublic.key
   --label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```

注意

如果公钥与私钥和/或证书对应，您应该指定与私钥和/或证书相同的 ID。

7. （可选）有些智能卡要求您通过锁定设置来完善卡：

```
$ pkcs15-init -F
```

此时您的智能卡在新创建的插槽中包含证书、私钥和公钥。您还创建了您的用户 PIN 和 PUK，以及安全响应 PIN 和 PUK。

4.5. 使用智能卡验证配置 SSH 访问

SSH 连接需要身份验证。您可以使用密码或证书。本节描述：

- 使用保存在智能卡中的证书启用验证所需的配置
- 使用 **authselect** 工具设置在取出卡时进行锁定

取出卡时进行锁定会强制在智能卡被取出后注销用户的登陆。

有关使用**authselect** 配置智能卡的详情，请参考使用 authselect 配置智能卡。

先决条件

- 该智能卡包含您的证书和私钥。
- 该卡插入读卡器并连接到计算机。
- 已安装并配置了 SSSD。
- 您的用户名与证书的 SUBJECT 中的 Common Name(CN)或 User ID(UID)匹配。
- **pcscd** 服务正在您的本地机器中运行。
  详情请查看安装用于管理和使用智能卡的工具。

流程

1. 在使用智能卡验证的用户主目录中为 SSH 密钥创建新目录：

```
# mkdir /home/example.user/.ssh
```

2. 使用 **opensc** 库运行 **ssh-keygen -D** 命令检索与智能卡中的私钥配对的现有公钥,并将其添加到用户 SSH 密钥目录的 **authorized_keys** 列表中,以启用智能卡验证的 SSH 访问。

```
# ssh-keygen -D /usr/lib64/pkcs11/opensc-pkcs11.so >>
~example.user/.ssh/authorized_keys
```

3. SSH 需要正确配置 **/.ssh** 目录和 **authorized_keys** 文件。要设置或更改访问权限，请输入：

```
# chown -R example.user:example.user ~example.user/.ssh/
# chmod 700 ~example.user/.ssh/
# chmod 600 ~example.user/.ssh/authorized_keys
```

4. 另外，显示密钥：

```
# cat ~example.user/.ssh/authorized_keys
```

终端会显示密钥。

5. 验证 **/etc/sssd/sssd.conf** 文件中是否启用了智能卡身份验证：
在 **[pam]** 部分，启用 pam 证书验证模块： **pam_cert_auth = True**

如果该 **sssd.conf** 文件还没有创建，您可以通过将以下脚本复制到命令行来创建最小功能配置：

```
# cat > /etc/sssd/sssd.conf <<EOF
[sssd]
services = nss, pam
domains = shadowutils

[nss]

[pam]
pam_cert_auth = True

[domain/shadowutils]
id_provider = files
EOF
```

6. 要使用 SSH 密钥，请使用 **authselect** 命令配置身份验证：

```
# authselect select sssd with-smartcard with-smartcard-lock-on-removal --force
```

现在，您可以使用以下命令验证 SSH 访问：

```
# ssh -I /usr/lib64/opensc-pkcs11.so -l example.user localhost hostname
```

如果配置成功，会提示您输入智能卡 PIN。

这个配置现在可以在本地运行。现在，您可以复制公钥并将其发布到您要使用 SSH 的所有服务器中的 **authorized_keys** 文件。

第 5 章 使用 AUTHSELECT 配置智能卡

这部分论述了如何配置智能卡以达到以下目的之一：

- 启用密码和智能卡验证
- 禁用密码并启用智能卡验证
- 在删除时启用锁定

先决条件

- 已安装 authselect
  authselect 工具在 Linux 主机中配置用户验证，您可以使用它配置智能卡验证参数。有关 authselect 的详情，请参考使用 authselect 配置用户身份验证。
- RHEL 8 支持的智能卡或者 USB 设备
  详情请参阅 RHEL8 中的智能卡支持。

5.1. 适用于智能卡的证书

在使用 **authselect** 配置智能卡前，您必须将证书导入您的卡中。您可以使用以下工具生成证书：

- Active Directory(AD)
- 身份管理(IdM)
  有关如何创建 IdM 证书的详情，请参阅请求新用户证书并将其导出到客户端。
- Red Hat Certificate System(RHCS)
  详情请参阅使用企业安全客户端管理智能卡。
- 本地认证认证机构。如果用户不是某个域的一部分或用于测试,您可以使用本地认证认证机构生成的证书。
  有关如何创建并导入本地证书到智能卡的详情，请参阅配置和导入本地证书到智能卡。

5.2. 启用用户密码验证来配置智能卡验证

这部分论述了如何在您的系统中启用智能卡和密码验证。

先决条件

- 智能卡包含您的证书和私钥。
- 在读卡器中插入卡并连接到计算机。
- **authselect** 工具安装在您的系统中。

流程

- 输入以下命令允许智能卡和密码验证：

```
# authselect select sssd with-smartcard --force
```

此时，智能卡验证会被启用。但是如果您忘记携带了智能卡，密码验证仍可以正常工作。

5.3. 配置 AUTHSELECT 以强制智能卡验证

authselect 工具可让您在系统中配置智能卡验证，并禁用默认密码验证。**authselect** 命令必须包括以下选项：

- **with-smartcard** - 启用智能卡验证
- **with-smartcard-required** - 启用专用智能卡验证（禁用密码验证）

先决条件

- 智能卡包含您的证书和私钥。
- 在读卡器中插入卡并连接到计算机。
- **authselect** 工具安装在您的本地系统中。

流程

- 输入以下命令强制智能卡验证：

```
# authselect select sssd with-smartcard  with-smartcard-required --force
```

此时，您只能使用智能卡登录。密码验证将不再工作。

5.4. 配置智能卡认证，使它在取出智能卡时进行锁定

authselect 服务可让您配置智能卡验证，以便在从读取器中删除智能卡后立即锁定屏幕。**authselect** 命令必须包括以下变量：

- **with-smartcard** - 启用智能卡验证
- **with-smartcard-required** - 启用专用智能卡验证（禁用密码验证）
- **with-smartcard-lock-on-removal** - 在删除智能卡后强制注销

先决条件

- 智能卡包含您的证书和私钥。
- 在读卡器中插入卡并连接到计算机。
- **authselect** 工具安装在您的本地系统中。

流程

- 输入以下命令启用智能卡验证、禁用密码验证并在删除时强制锁定：

```
# authselect select sssd  with-smartcard  with-smartcard-required with-smartcard-lock-on-removal --force
```

现在，当您取出卡时，屏幕会锁定。您必须重新插入智能卡来解锁它。

第 6 章 使用智能卡远程验证 SUDO

这部分论述了如何使用智能卡远程验证 sudo。在本地运行 ssh-agent 服务并可将 ssh-agent 套接字转发到远程机器后,您可以使用 sudo PAM 模块中的 SSH 验证协议远程验证用户。

使用智能卡在本地登录后,您可以使用 SSH 到远程机器并在不提示输入密码的情况下运行 sudo 命令,使用 SSH 转发智能卡验证。

在本例中,客户端通过 SSH 连接到 IPA 服务器,并在将凭证保存在智能卡中的 IPA 服务器中运行 sudo 命令。

- 在 IdM 中创建 sudo 规则
- 为 sudo 设置 PAM 模块
- 使用智能卡远程连接到 sudo

6.1. 在 IDM 中创建 SUDO 规则

这个步骤描述了如何在 IdM 中创建 sudo 规则以便为 **ipauser1** 权限在远程主机上运行 sudo。

在本例中, **less** 和 **whoami** 命令作为 sudo 命令添加来测试该过程。

先决条件

- IdM 用户已创建。在本例中,用户是 **ipauser1**。
- 您有远程运行 sudo 系统的主机名。在本例中,主机是 **server.ipa.test**。

流程

1. 创建一个名为 **adminrule** 的 **sudo** 规则,以允许用户运行命令。

```
ipa sudorule-add adminrule
```

2. 添加 **less** 和 **whoami** 作为 **sudo** 命令：

```
ipa sudocmd-add /usr/bin/less
ipa sudocmd-add /usr/bin/whoami
```

3. 在 **adminrule** 中添加 **less** 和 **whoami** 命令：

```
ipa sudorule-add-allow-command adminrule --sudocmds /usr/bin/less
ipa sudorule-add-allow-command adminrule --sudocmds /usr/bin/whoami
```

4. 在 **adminrule** 中添加 **ipauser1** 用户：

```
ipa sudorule-add-user adminrule --users ipauser1
```

5. 将运行 **sudo** 的主机添加到 **adminrule**:

```
ipa sudorule-add-host adminrule --hosts server.ipa.test
```

其它资源

- 如需更多信息,请参阅 **ipa sudorule-add --help**。
- 如需更多信息,请参阅 **ipa sudocmd-add --help**。

6.2. 为 SUDO 设置 PAM 模块

这个步骤描述了如何在运行 sudo 的任何主机上使用智能卡为 sudo 验证安装和设置 **pam_ssh_agent_auth.so** PAM 模块。

流程

1. 安装 PAM SSH 代理：

```
dnf -y install pam_ssh_agent_auth
```

2. 在其它 **auth** 条目前,将 **pam_ssh_agent_auth.so** 的 **authorized_keys_command** 添加到 **/etc/pam.d/sudo** 文件中：

```
#%PAM-1.0
auth sufficient pam_ssh_agent_auth.so
authorized_keys_command=/usr/bin/sss_ssh_authorizedkeys
auth       include      system-auth
account    include      system-auth
password   include      system-auth
session    include      system-auth
```

3. 要启用 SSH 代理转发在运行 sudo 命令时,请将以下内容添加到 **/etc/sudoers** 文件中：

```
Defaults env_keep += "SSH_AUTH_SOCK"
```

这允许存储在 IPA/SSSD 中的智能卡中的公钥用户在不输入密码的情况下与 sudo 验证。

4. 重启 **sssd** 服务：

```
systemctl restart sssd
```

其它资源

- 请查看 **pam** man page。

6.3. 使用智能卡远程连接到 SUDO

这个步骤描述了如何配置 SSH 代理和客户端以便使用智能卡远程连接到 sudo。

先决条件

- 您已在 IdM 中创建了 sudo 规则。
- 您已在要运行 sudo 的远程系统中安装并设置了 **pam_ssh_agent_auth** PAM 模块用于 sudo 验证。

流程

流程

1. 启动 SSH 代理（如果还没有运行）。

```
eval `ssh-agent`
```

2. 在 SSH 代理中添加智能卡。提示时输入您的 PIN:

```
ssh-add -s /usr/lib64/opensc-pkcs11.so
```

3. 通过 SSH 与启用了 ssh-agent 转发（使用 **-A** 选项）连接到要远程运行 **sudo** 的系统：

```
ssh -A ipauser1@server.ipa.test
```

验证步骤

- 使用 **sudo** 运行 **whoami** 命令：

```
sudo /usr/bin/whoami
```

您不应该被提示输入 PIN 或密码。