Red Hat Enterprise Linux 9

在 RHEL 中配置身份验证和授权

使用 SSSD、authselect 和 sssctl 配置身份验证和授权

Last Updated: 2024-07-04

Red Hat Enterprise Linux 9 在 RHEL 中配置身份验证和授权

使用 SSSD、authselect 和 sssctl 配置身份验证和授权

法律通告

摘要

您可以配置 Red Hat Enterprise Linux (RHEL)来对服务进行身份验证并授权用户使用服务，如 Red Hat Identity Management (IdM)、活动目录(AD)和 LDAP 目录。为此，RHEL 使用系统安全服务守护进程(SSSD)与这些服务进行通信。authselect 和 sssctl 等工具支持您配置 SSSD、可插拔验证模块(PAM)和名称服务交换(NSS)。

目录

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 Jira 提交反馈（需要帐户）

1. 登录到 Jira 网站。
2. 在顶部导航栏中点 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 系统身份验证简介

建立安全网络环境的基点之一可确保访问权限仅限于授权用户。允许访问时，用户可以向系统进行身份验证，验证其身份。

在任何 Red Hat Enterprise Linux 系统上，可以使用各种服务来创建和管理用户身份。这包括本地系统文件、连接到 Kerberos 或 Samba 等大型身份域的服务，或者用于创建这些域的工具。

1.1. 确认用户身份

身份验证是确认身份的过程。对于网络交互，身份验证涉及由另一方识别方。可以通过许多方式通过网络使用身份验证，如简单密码、证书、免密码方法、一次性密码(OTP)令牌或 biometric 扫描。

授权定义了经过身份验证的用户允许或访问的内容。

身份验证要求用户提供一些凭据来验证其身份。所需的凭证类型由正在使用的身份验证机制定义。系统中本地用户有几个验证：

基于密码的身份验证

几乎所有软件都允许用户通过提供可识别的用户名和密码来验证。这也称为简单身份验证。

基于证书的验证

基于证书的客户端身份验证是安全套接字层(SSL)协议的一部分。客户端以数字方式签署随机生成的数据，并通过网络发送证书和签名数据。服务器验证签名并确认证书的有效性。

Kerberos 身份验证

Kerberos 建立一个简短凭证的系统，称为票据授予票(TGT)。用户提供凭证，即用户名和密码，用于标识用户并向系统指明用户可以发出票据。然后，TGT 可以反复用来请求对其他服务（如网站和电子邮件）的访问票据。使用 Kerberos 进行身份验证可让用户以这种方式处理单个身份验证过程。

基于智能卡的验证

这是基于证书的验证变体。智能卡（或令牌）存储用户证书；当用户将令牌插入系统时，系统会读取证书并授予访问权限。使用智能卡的单点登录通过三个步骤：

1. 用户在卡阅读器中插入智能卡。Red Hat Enterprise Linux 上的可插拔验证模块(PAM)会检测插入的智能卡。
2. 系统将证书映射到用户条目，然后将智能卡上提供的证书（其使用基于证书的身份验证中的私钥加密）与用户条目中存储的证书进行比较。
3. 如果证书针对密钥分发中心(KDC)成功验证，则允许用户登录。

基于智能卡的身份验证构建在 Kerberos 构建的简单验证层上，方法是添加证书作为额外的识别机制，以及添加物理访问要求。如需更多信息，请参阅管理智能卡验证。

一次性密码身份验证

一次性密码可为您的身份验证安全性增加一步。身份验证将您的密码与自动生成的一次性密码结合使用。如需更多信息，请参阅 身份管理中的一次性密码(OTP)身份验证。

Passkey 身份验证

passkey 是一个 FIDO2 身份验证设备，由 libfido2 库支持，如 Yubikey 5 和 Nitrokey。它允许免密码和多因素身份验证。如果您的系统已注册并连接到 IdM 环境，这个验证方法会自动发出 Kerberos 票据，该票据为身份管理(IdM)用户启用单点登录(SSO)。如需更多信息，请参阅在 IdM 环境中启用 passkey 身份验证。

外部身份提供程序

您可以将用户与支持 OAuth 2 设备授权流的外部身份提供者(IdP)关联。当这些用户使用 RHEL 9.1 或更高版本中提供的 SSSD 版本进行身份验证时，它们会在外部 IdP 执行身份验证和授权后收到带有 Kerberos 票据的 RHEL Identity Management (IdM)单点登录功能。如需更多信息，请参阅使用外部身份提供程序向 IdM 进行身份验证。

1.2. 规划单点登录

如果没有中央身份存储和维护自己一组用户和凭证的应用程序，用户必须为其打开的每个服务或应用程序输入密码。

通过配置单点登录，管理员会创建一个密码存储，以便用户可以使用单个密码登录一次，并对所有网络资源进行身份验证。

Red Hat Enterprise Linux 支持对多个资源进行单点登录，包括登录工作站、解锁屏幕保存器以及使用 Mozilla Firefox 访问安全网页。通过其他可用的系统服务，如 Privileged Access Management (PAM)、名称服务交换机(NSS)和 Kerberos，可以将其他系统应用程序配置为使用这些身份源。

单点登录是方便用户和服务器和网络的另一层安全性。单点登录隐藏安全有效的身份验证。Red Hat Enterprise Linux 提供两种身份验证机制，可用于启用单点登录：

- 通过 Kerberos 域和 Active Directory 域基于 Kerberos 的身份验证
- 基于智能卡的验证

这两种方法都创建一个集中身份存储（通过 Kerberos 域或公钥基础架构中的证书颁发机构），以及本地系统服务，然后使用这些身份域而不是维护多个本地存储。

1.3. 适用于本地用户身份验证的服务

所有 Red Hat Enterprise Linux 系统均有一些服务可用于为本地系统上的本地用户配置身份验证。它们是：

身份验证设置

- Authentication Configuration 工具 **authselect** 设置不同的身份后端，并为系统设置身份验证方法（如密码、指纹或智能卡）。

身份后端设置

- 安全系统服务守护进程(SSSD)设置多个身份提供程序（通常基于 LDAP 的目录，如 Microsoft Active Directory 或 Red Hat Enterprise Linux IdM），然后可供本地系统和应用程序用于用户。密码和票据被缓存，允许通过重新使用凭证进行离线身份验证和单点登录。
- **realmd** 服务是一个命令行工具，其允许您配置身份验证后端，即用于 IdM 的 ssd。 **realmd** 服务根据 DNS 记录检测可用的 IdM 域，配置 SSSD，然后将系统作为一个帐户加入到域中。
- 名称服务切换(NSS)是低级系统调用的机制，用于返回有关用户、组或主机的信息。NSS 确定应使用哪个模块来获取所需信息的源（即哪个模块）。例如，用户信息可以位于 traditioal UNIX 文件中，如 **/etc/passwd** 文件，或者位于基于 LDAP 的目录，而主机地址可以从文件读取，如 **/etc/hosts** 文件或 DNS 记录；NSS 查找信息所在的位置。

身份验证机制

- 可插拔验证模块(PAM)提供了一个系统来设置验证策略。使用 PAM 进行身份验证的应用程序会加载控制不同身份验证方面的不同模块；应用程序使用的 PAM 模块取决于应用的配置方式。可用的 PAM 模块包括 Kerberos、Winbind、SSSD 或本地 UNIX 文件的身份验证。

其他服务和应用程序也可用，但它们是常见的服务和应用程序。

第 2 章 使用 AUTHSELECT 配置用户身份验证

authselect 是一个实用程序，允许您通过选择特定的配置集来配置系统身份和身份验证源。配置集（profile）是一组文件，描述生成的可插拔验证模块 (PAM) 和网络安全服务 (NSS) 配置。您可以选择默认的配置集或创建自定义配置集。

2.1. AUTHSELECT 的作用

您可以使用 **authselect** 工具在 Red Hat Enterprise Linux 9 主机上配置用户身份验证。

您可以通过选择一个可用的配置集来配置身份信息和验证源和供应商：

- 默认 **sssd** 配置集为使用 LDAP 身份验证的系统启用系统安全服务守护进程 (SSSD)。
- **winbind** 配置集为直接与 Microsoft Active Directory 集成的系统启用 Winbind 实用程序。
- **minimal** 配置集只用于直接来自系统文件中的本地用户和组，它允许管理员删除不再需要的网络身份验证服务。

在为给定主机选择了一个 **authselect** 配置集后，配置集将应用于登录到主机的每个用户。

红帽建议在半集中式身份管理环境中使用 **authselect**，例如，如果您的组织使用 LDAP 或 Winbind 数据库来验证用户，以便在您的域中使用服务。

警告

如果出现以下情况，您不需要使用 **authselect** ：

- 您的主机是 Red Hat Enterprise Linux Identity Management(IdM)的一部分。使用 **ipa-client-install** 命令将您的主机加入 IdM 域会自动在主机上配置 SSSD 身份验证。
- 您的主机通过 SSSD 作为 Active Directory 的一部分。调用 **realm join** 命令将您的主机加入 Active Directory 域会自动在您的主机上配置 SSSD 身份验证。

红帽建议不要更改 **ipa-client-install** 或 **realm join** 配置的 **authselect** 配置集。如果您需要修改它们，请在进行任何修改前显示当前的设置，以便在需要时将其恢复：

```
$ authselect current
Profile ID: sssd
Enabled features:
- with-sudo
- with-mkhomedir
- with-smartcard
```

2.1.1. authselect 修改的文件和目录

在以前的 Red Hat Enterprise Linux 版本中使用的 **authconfig** 实用程序会创建并修改许多不同的配置文件，从而使故障排除变得更加困难。**authselect** 简化了测试和故障排除过程，因为它仅修改以下文件和目录：

/etc/nsswitch.conf	GNU C 库和其他应用使用此名称服务交换机 (NSS) 配置文件来确定从中获取一系列类别中的名称服务信息的来源，以及顺序。每个类别的信息都由一个数据库名来标识。
/etc/pam.d/* 文件	Linux-PAM（可插拔验证模块）是处理系统中应用程序（服务）验证任务的模块系统。验证的特性是动态可配置的：系统管理员可以选择如何单独提供服务提供应用程序验证用户。 **/etc/pam.d/** 目录中的配置文件列出了将执行服务所需身份验证任务的 PAM，以及在单个 PAM 失败时 PAM-API 相应行为。 这些文件还包含以下信息： • 用户密码锁定条件 • 使用智能卡验证的能力 • 使用指纹读取器验证的能力
/etc/dconf/db/distro.d/* 文件	此目录包含 **dconf** 实用程序的配置集，可用于管理 GNOME 桌面图形用户界面 (GUI) 的设置。

2.1.2. /etc/nsswitch.conf 中的数据提供程序

默认 **sssd** 配置集通过在 **/etc/nsswitch.conf** 中创建 **sss** 条目将 SSSD 设置为信息源：

```
passwd:     sss files
group:      sss files
netgroup:   sss files
automount:  sss files
services:   sss files
...
```

这意味着，如果请求了有关这些项目之一的信息，系统首先会查找 SSSD：

- **passwd** 用于用户信息
- **group** 用户组群信息
- **netgroup** 用于 NIS **netgroup** 信息
- **automount** 用于 NFS 自动挂载信息
- **services** 用于有关服务的信息

只有在 **sssd** 缓存和提供身份验证的服务器上找不到请求的信息，或者 **sssd** 没有运行时，系统才会查看本地文件，即 **/etc/***。

例如，如果请求有关用户 ID 的信息，则首先在 **sssd** 缓存中搜索用户 ID。如果未在此处找到，则会查阅 **/etc/passwd** 文件。类似地，如果请求用户的组从属关系，则首先在 **sssd** 缓存中搜索它，并且仅在未找到时搜索 **/etc/group** 文件。

实际上，本地**文件**数据库通常不会被查阅。最重要的例外是 **root** 用户，它永远不会由 **sssd** 处理，而是由**文件**处理。

2.2. 选择 AUTHSELECT 配置集

作为系统管理员，您可以为特定主机选择 **authselect** 工具的配置集。该配置集将应用于登录到主机的每个用户。

先决条件

- 运行 **authselect** 命令需要 **root** 凭证

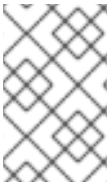
步骤

- 选择适合您的身份验证供应商的 **authselect** 配置集。例如，若要登录到使用 LDAP 的公司网络，请选择 **sssd**。

```
# authselect select sssd
```

  - （可选）您可以在 **authselect select sssd** 或 **authselect select winbind** 命令中添加以下选项来修改默认配置集设置，例如：
    - **with-faillock**
    - **with-smartcard**
    - **with-fingerprint**

  要查看可用选项的完整列表，请参阅将脚本从 authconfig 转换到 authselect 或 **authselect-migration(7)** man page。



注意

在完成 **authselect select** 过程前,请确定正确配置了与您的配置集相关的配置文件。例如，如果 **sssd** 守护进程没有正确配置并处于活动状态，则运行 **authselect select** 会导致只有本地用户可以使用 **pam_unix** 进行身份验证。

验证步骤

1. 验证 **/etc/nsswitch.conf** 中是否存在 SSSD 的 **sss** 条目：

```
passwd:     sss files
group:      sss files
netgroup:   sss files
automount:  sss files
services:   sss files
...
```

2. 在 **pam_sss.so** 条目中查看 **/etc/pam.d/system-auth** 文件的内容：

```
# Generated by authselect on Tue Sep 11 22:59:06 2018
# Do not modify this file manually.

auth        required      pam_env.so
auth        required      pam_faildelay.so delay=2000000
auth        [default=1 ignore=ignore success=ok]    pam_succeed_if.so uid >= 1000 quiet
auth        [default=1 ignore=ignore success=ok]    pam_localuser.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 1000 quiet_success
auth        sufficient    pam_sss.so forward_pass
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
...
```

其它资源

- authselect 的作用
- 修改可用的 authselect 配置集
- 创建并部署您自己的 authselect 配置集

2.3. 修改可用的 AUTHSELECT 配置集

作为系统管理员，您可以修改一个默认配置集使其适合您的需要。

您可以修改 **/etc/authselect/user-nsswitch.conf** 文件中的任何项目，但以下除外：

- **passwd**
- **group**
- **netgroup**
- **automount**
- **services**

随后运行 **authselect select profile_name** 会导致将允许的更改从 **/etc/authselect/user-nsswitch.conf** 传输到 **/etc/nsswitch.conf** 文件。不可接受的更改会被默认配置集的配置覆盖。

重要

不要直接修改 **/etc/nsswitch.conf** 文件。

步骤

1. 选择一个 **authselect** 配置集，例如：

```
# authselect select sssd
```

2. 按照您所需的更改编辑 **/etc/authselect/user-nsswitch.conf** 文件。

3. 应用 **/etc/authselect/user-nsswitch.conf** 文件中的更改：

```
# authselect apply-changes
```

验证步骤

- 查看 **/etc/nsswitch.conf** 文件，以验证 **/etc/authselect/user-nsswitch.conf** 中的更改是否已在此传播。

其它资源

- authselect 的作用

2.4. 创建并部署您自己的 AUTHSELECT 配置集

作为系统管理员，您可以通过生成一个默认配置集的自定义副本来创建和部署自定义配置集。

这在修改一个现成的 authselect 配置集不足以满足您的需要时特别有用。当您部署自定义配置集时，配置集将应用于记录到给定主机上的每个用户。

步骤

1. 使用 **authselect create-profile** 命令创建自定义配置集。例如，基于可用的 **sssd** 配置集创建一个名为 **user-profile** 的自定义配置集，您可以自行在 **/etc/nsswitch.conf** 文件中配置项目：

```
# authselect create-profile user-profile -b sssd --symlink-meta --symlink-pam
New profile was created at /etc/authselect/custom/user-profile
```

警告

如果您计划修改 **/etc/authselect/custom/user-profile/{password-auth,system-auth,fingerprint-auth,smartcard-auth,postlogin}**，然后输入没有 **--symlink-pam** 选项的上述命令。这是为了在升级 **authselect-libs** 过程中确保修改持久。

在命令中包含 **--symlink-pam** 选项意味着 PAM 模板将是原始配置集文件的符号链接，而不是其副本；包括 **--symlink-meta** 选项意味着元文件（如 README 和 REQUIREMENTS）将是原始配置文件文件的符号链接，而不是其副本。这样可确保以后对原始配置集中的 PAM 模板和 meta 文件的所有更新都会反映在您的自定义配置集中。

这个命令会在 **/etc/authselect/custom/user-profile/** 目录中创建 **/etc/nsswitch.conf** 文件的副本。

2. 配置 **/etc/authselect/custom/user-profile/nsswitch.conf** 文件。

3. 运行 **authselect select** 命令选择自定义配置集，并添加 **custom/*name_of_the_profile*** 作为一个参数。例如，要选择 **user-profile** 配置集：

```
# authselect select custom/user-profile
```

为您的机器选择 **user-profile** 配置集意味着，如果以后红帽更新了 **sssd** 配置集，您就可以受益于这些更新（对 **/etc/nsswitch.conf** 文件的更新除外）。

例 2.1. 创建配置集

以下步骤演示了如何基于 **sssd** 配置集创建一个配置集，它仅在 **/etc/hosts** 文件中的本地静态表中查找主机名，而不在 **dns** 或 **myhostname** 数据库中查找。

1. 编辑 **/etc/nsswitch.conf** 文件，修改以下行：

```
hosts:      files
```

2. 基于 **sssd** 创建自定义配置集，它排除了对 **/etc/nsswitch.conf** 的更改：

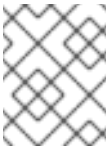
```
# authselect create-profile user-profile -b sssd --symlink-meta --symlink-pam
```

3. 选择配置集：

```
# authselect select custom/user-profile
```

4. （可选）检查是否已选择自定义配置集
   - 根据所选的 **sssd** 配置集创建 **/etc/pam.d/system-auth** 文件
   - 原封不动保留 **/etc/nsswitch.conf** 中的配置：

```
hosts:      files
```



注意

运行 **authselect select sssd** 将会产生 **hosts: files dns myhostname**

其它资源

- authselect 的作用

2.5. 将脚本从 AUTHCONFIG 转换为 AUTHSELECT

如果您使用 **ipa-client-install** 或 **realm join** 加入域，您可以在脚本中安全地删除任何 **authconfig** 调用。如果不可能，将每个 **authconfig** 调用替换为其等同的 **authselect** 调用。要做到这一点请选择正确的配置集和适当的选项。另外，请编辑必要的配置文件：

- **/etc/krb5.conf**
- **/etc/sssd/sssd.conf**（用于 **sssd** 配置文件）或 **/etc/samba/smb.conf**（用于 **winbind** 配置集）

authconfig 选项和 authselect 配置集的关系和authconfig 选项对应的 Authselect 配置集选项显示了与 **authconfig** 选项对应的 **authselect**。

表 2.1. authconfig 选项与 authselect 配置集的关系

authconfig 选项	authselect 配置集
--enableldap --enableldapauth	**sssd**
--enablesssd --enablesssdauth	**sssd**
--enablekrb5	**sssd**
--enablewinbind --enablewinbindauth	**winbind**

表 2.2. authselect profile 选项等同于 authconfig 选项

authconfig 选项	authselect 配置集特性
--enablesmartcard	**with-smartcard**
--enablefingerprint	**with-fingerprint**
--enableecryptfs	**with-ecryptfs**
--enablemkhomedir	**with-mkhomedir**
--enablefaillock	**with-faillock**
--enablepamaccess	**with-pamaccess**
--enablewinbindkrb5	**with-krb5**

与 authconfig 命令等效的 authselect 命令示例 显示了，Kickstart 对 **authconfig** 的调用转换为 Kickstart 对 **authselect** 的调用的示例。

表 2.3. 与 authconfig 命令等同的 authselect 命令示例

authconfig 命令	authselect 等同的命令
authconfig --enableldap --enableldapauth --enablefaillock --updateall	**authselect select sssd with-faillock**
authconfig --enablesssd --enablesssdauth --enablesmartcard --smartcardmodule=sssd --updateall	**authselect select sssd with-smartcard**
authconfig --enableecryptfs --enablepamaccess --updateall	**authselect select sssd with-ecryptfs with-pamaccess**

authconfig 命令	authselect 等同的命令
authconfig --enablewinbind --enablewinbindauth --winbindjoin=Administrator --updateall	**realm join -U Administrator --client-software=winbind WINBINDDOMAIN**

2.6. 其他资源

- 什么是 pam_faillock 以及如何在 Red Hat Enterprise Linux 8 和 9 中使用它？
- 在 Red Hat Enterprise Linux 8 中设置密码策略/复杂度

第 3 章 了解 SSSD 及其优势

系统安全服务后台程序 (SSSD) 是一种用于访问远程目录和身份验证机制的系统服务。以下章节概述了 SSSD 的工作原理、使用它的益处、如何处理配置文件，以及您可以配置的身份和身份验证提供程序。

3.1. SSSD 如何工作

系统安全服务后台程序 (SSSD) 是一种系统服务，可让您访问远程目录和身份验证机制。您可以把一个本地系统（一个 SSSD *客户端*）连接到外部后端系统（一个 *provider*）。

例如：

- 一个 LDAP 目录
- 一个 Identity Management（IdM）域
- 一个 Active Directory（AD）域
- 一个 Kerberos realm

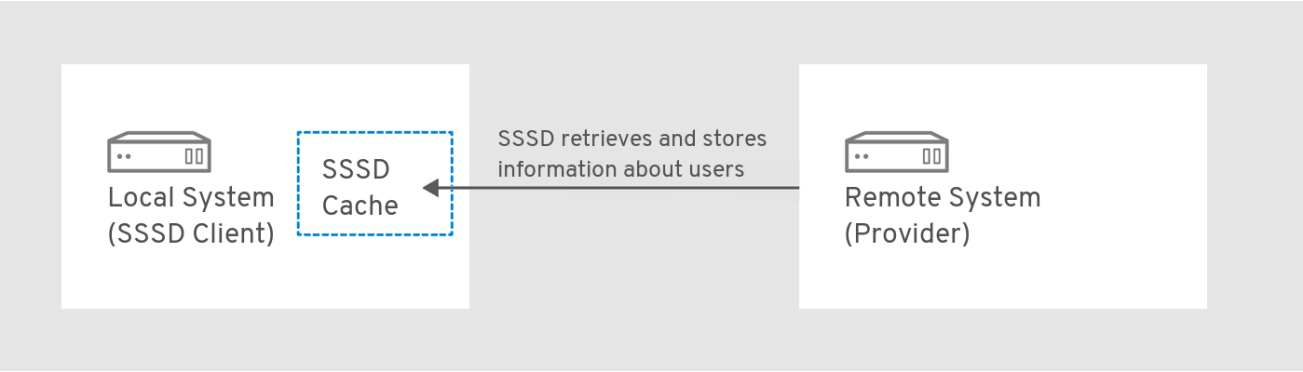
SSSD 分为两个阶段：

1. 它将客户端连接到远程供应商以检索身份和验证信息。
2. 它使用获得的验证信息来创建客户端用户和凭证的本地缓存。

然后，本地系统中的用户可以使用保存在远程供应商的用户帐户进行身份验证。

SSSD 不会在本地系统上创建用户帐户。但是，可将 SSSD 配置为为 IdM 用户创建主目录。创建后，当用户注销时，IdM 用户主目录及其在客户端中的内容不会被删除。

图 3.1. SSSD 如何工作



SSSD 还可以为多个系统服务提供缓存，如名称服务交换机 (NSS) 或可插拔验证模块 (PAM)。

注意

仅使用 SSSD 服务来缓存用户信息。运行名称服务缓存守护进程(NSCD)和 SSSD 在同一系统上进行缓存可能会导致性能问题和冲突。

3.2. 使用 SSSD 的好处

使用系统安全服务后台程序 (SSSD) 在用户身份检索和用户身份验证方面具有多个益处。

离线验证

SSSD 可选保留一个从远程供应商获取的用户身份和凭证缓存。在此设置中，如果用户已在会话开始时对远程提供程序进行身份验证一次 - 即使远程提供程序或客户端脱机，也可以成功验证资源。

单一用户帐户：提高身份验证过程的一致性

使用 SSSD 时，不需要同时维护中央帐户和本地用户帐户进行离线身份验证。条件为：

- 在特定的会话中，用户必须至少登录一次：当用户第一次登录时,客户端必须连接到远程供应商。
- SSSD 中必须启用缓存。
  在没有 SSSD 时，远程用户通常会有多个用户帐户。例如，要连接到虚拟专用网络（VPN），远程用户需要有一个本地系统帐户，以及另外一个 VPN 帐户。在这种情况下，您必须首先在私有网络中进行身份验证，以便从远程服务器获取用户,并在本地缓存用户凭证。

  使用 SSSD 时，利用缓存和离线身份验证，远程用户只需向本地机器验证即可连接到网络资源。然后，SSSD 维护其网络凭证。

这可以减少身份和验证提供程序上的负载

在请求信息时，客户端首先检查本地 SSSD 缓存。只有在缓存中没有这些信息时，SSSD 才会联系远程供应商。

3.3. 基于每个客户端有多个 SSSD 配置文件

SSSD 的默认配置文件为 **/etc/sssd/sssd.conf**。除了这个文件外，SSSD 还可以从 **/etc/sssd/conf.d/** 目录中的所有 ***.conf** 文件中读取其配置。

这个组合允许您在所有客户端中使用默认 **/etc/sssd/sssd.conf** 文件，并在以后的配置文件中添加附加设置，以针对每个客户端单独扩展功能。

SSSD 如何处理配置文件
SSSD 按以下顺序读取配置文件：

1. 主 **/etc/sssd/sssd.conf** 文件
2. **/etc/sssd/conf.d/** 中的其他 ***.conf** 文件，按字母顺序排列

如果同一参数出现在多个配置文件中，SSSD 将使用最后一个读取的参数。

注意

SSSD 不读取 **conf.d** 目录中的隐藏文件（以 **.** 开头的文件）。

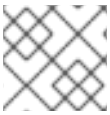
3.4. SSSD 的身份和验证供应商

您可以将 SSSD 客户端连接到外部身份和身份验证供应商，如 LDAP 目录、身份管理 (IdM)、Active Directory(AD)域或 Kerberos 域。然后，SSSD 客户端使用 SSSD 供应商访问身份和身份验证远程服务。您可以将 SSSD 配置为使用不同的身份和身份验证供应商或它们的组合。

身份识别和身份验证提供程序作为 SSSD 域
身份和身份验证提供程序在 SSSD 配置文件 **/etc/sssd/sssd.conf** 中配置为 *domains（域）*。提供程序在文件的 **[domain/*name of the domain*]** 或 **[domain/default]** 部分中列出。

可将单个域配置为以下供应商之一：

- 一个*身份供应商*，它提供用户信息，如 UID 和 GID。
  - 使用 **/etc/sssd/sssd.conf** 文件的 **[domain/*name of the domain*]** 部分中的 **id_provider** 选项将域指定为 *身份提供程序*。
- 一个*身份验证供应商*，用于处理身份验证请求。
  - 使用 **/etc/sssd/sssd.conf** 的 **[domain/*name of the domain*]**部分中的 **auth_provider** 选项将域指定为 *身份验证提供程序*。
- *访问控制提供程序*，负责处理授权请求。
  - 使用 **/etc/sssd/sssd.conf** 的 **[domain/*name of the domain*]**部分中的 **access_provider** 选项将域指定为 *访问控制提供程序*。默认情况下，选项设置为 **permit**，这将始终允许所有访问。详情请查看 **sssd.conf**(5)man page。
- 组合这些供应商，例如，所有对应的操作都是在单一服务器中执行的。
  - 在本例中，**id_provider**、**auth_provider** 和 **access_provider** 选项都列在 **/etc/sssd/sssd.conf** 的相同的 **[domain/*name of the domain*]** 或 **[domain/default]** 部分。



注意

您可以为 SSSD 配置多个域。您必须至少配置一个域，否则 SSSD 不会启动。

代理供应商
代理供应商充当 SSSD 和 SSSD 资源之间的中间中继。使用代理供应商时，SSSD 会连接到代理服务，代理会加载指定的库。

您可以将 SSSD 配置为使用代理提供商，以启用：

- 其他验证方法，如指纹扫描仪
- 传统系统，如 NIS
- 在 **/etc/passwd** 文件中定义的本地系统帐户作为身份提供程序和远程身份验证提供程序，如 Kerberos
- 使用智能卡验证本地用户

身份供应商可以和认证服务商组合使用
您可以将 SSSD 配置为使用以下身份和验证供应商的组合。

表 3.1. 身份供应商可以和认证服务商组合使用

身份供应商	验证供应商
身份管理 [a]	身份管理
Active Directory	Active Directory
LDAP	LDAP
LDAP	Kerberos

身份供应商	验证供应商
Proxy	Proxy
Proxy	LDAP
Proxy	Kerberos

[a] LDAP 供应商类型的扩展。

其他资源

- 使用 authselect 配置用户身份验证
- 使用 SSSD 查询域信息 [1]
- 使用 SSSD 报告主机的用户访问权限

[1] 要使用 **sssctl** 实用程序列出并验证域的状态，您的主机应注册为与 Active Directory (AD) 林信任协议中的身份管理 (IdM)。

第 4 章 配置 SSSD 以使用 LDAP 并需要 TLS 身份验证

系统安全服务守护进程(SSSD)是一个在 Red Hat Enterprise Linux 主机上管理身份数据检索和身份验证的守护进程。系统管理员可以将主机配置为使用独立 LDAP 服务器作为用户帐户数据库。管理员还可以指定与 LDAP 服务器的连接必须使用 TLS 证书加密的要求。

注意

强制 TLS 的 SSSD 配置选项 **ldap_id_use_start_tls**，默认为 **false**。当使用 **ldap://** 而不是 TLS 进行身份查找时，可能会导致攻击向量的风险，即一个中间人(MITM)攻击，例如，它允许您通过更改 LDAP 搜索中返回对象的 UID 或 GID 用户来冒充用户。

确保您的设置在可信环境中可以操作，并决定是否可以安全地对 **id_provider = ldap** 使用未加密的通信。注意 **id_provider = ad** 和 **id_provider = ipa** 不受影响，因为它们使用 SASL 和 GSSAPI 保护的加密连接。

如果不能安全地使用未加密的通信，您应该通过在 **/etc/sssd/sssd.conf** 文件中将 **ldap_id_use_start_tls** 选项设为 **true** 来强制实施 TLS。

4.1. 使用 SSSD 的 OPENLDAP 客户端以加密的方式从 LDAP 检索数据

LDAP 对象的验证方法可以是 Kerberos 密码，也可以是 LDAP 密码。请注意，此处没有解决 LDAP 对象的身份验证和授权问题。

重要

使用 LDAP 配置 SSSD 是一个复杂的流程，需要对 SSSD 和 LDAP 有非常专业的知识。考虑改为使用集成和自动化解决方案，如 Active Directory 或 Red Hat Identity Management (IdM)。有关 IdM 的详情，请参阅规划身份管理。

identity :leveloffset: +1

第 5 章 配置 SSSD 以使用 LDAP 并需要 TLS 身份验证

完成这个步骤，将 Red Hat Enterprise Linux (RHEL) 系统配置为 OpenLDAP 客户端。

使用以下客户端配置：

- RHEL 系统验证存储在 OpenLDAP 用户帐户数据库中的用户。
- RHEL 系统使用系统安全服务守护进程 (SSSD) 服务检索用户数据。
- RHEL 系统通过 TLS 加密的连接与 OpenLDAP 服务器通信。

注意

您还可以使用此流程将 RHEL 系统配置为 Red Hat Directory Server 的客户端。

先决条件

- OpenLDAP 服务器安装并配置了用户信息。
- 您在要配置为 LDAP 客户端的主机上具有 root 权限。
- 在您要配置为 LDAP 客户端的主机上，已创建并配置了 **/etc/sssd/sssd.conf** 文件，以将 **ldap** 指定为 **autofs_provider** 和 **id_provider**。
- 您有来自发布 OpenLDAP 服务器的证书颁发机构的 root CA 签名证书链的 PEM 格式副本，存储在名为 **core-dirsrv.ca.pem** 的本地文件中。

步骤

1. 安装必要的软件包：

```
# dnf -y install openldap-clients sssd sssd-ldap oddjob-mkhomedir
```

2. 将身份验证供应商切换到 **sssd**：

```
# authselect select sssd with-mkhomedir
```

3. 将包含 root CA 签名证书链的 **core-dirsrv.ca.pem** 文件从颁发 OpenLDAP 服务器的 SSL/TLS 证书的证书颁发机构链复制到 **/etc/openldap/certs** 文件夹。

```
# cp core-dirsrv.ca.pem /etc/openldap/certs
```

4. 将 LDAP 服务器的 URL 和后缀添加到 **/etc/openldap/ldap.conf** 文件中：

```
URI ldap://ldap-server.example.com/
BASE dc=example,dc=com
```

5. 在 **/etc/openldap/ldap.conf** 文件中，向 **/etc/openldap/certs/core-dirsrv.ca.pem** 添加指向 **TLS_CACERT** 参数的行：

```
# When no CA certificates are specified the Shared System Certificates
# are in use. In order to have these available along with the ones specified
# by TLS_CACERTDIR one has to include them explicitly:
```

```
TLS_CACERT /etc/openldap/certs/core-dirsrv.ca.pem
```

6. 在 **/etc/sssd/sssd.conf** 文件中，将您的环境值添加到 **ldap_uri** 和 **ldap_search_base** 参数中，并将 **ldap_id_use_start_tls** 设为 **True** ：

```
[domain/default]
id_provider = ldap
autofs_provider = ldap
auth_provider = ldap
chpass_provider = ldap
ldap_uri = ldap://ldap-server.example.com/
ldap_search_base = dc=example,dc=com
ldap_id_use_start_tls = True
cache_credentials = True
ldap_tls_cacertdir = /etc/openldap/certs
ldap_tls_reqcert = allow

[sssd]
services = nss, pam, autofs
domains = default

[nss]
homedir_substring = /home
…
```

7. 在 **/etc/sssd/sssd.conf** 中，修改 **[domain]** 部分中的 **ldap_tls_cacert** 和 **ldap_tls_reqcert** 值来指定 TLS 身份验证要求：

```
…
cache_credentials = True
ldap_tls_cacert = /etc/openldap/certs/core-dirsrv.ca.pem
ldap_tls_reqcert = hard
…
```

8. 更改 **/etc/sssd/sssd.conf** 文件的权限：

```
# chmod 600 /etc/sssd/sssd.conf
```

9. 重启并启用 SSSD 服务和 **oddjobd** 守护进程：

```
# systemctl restart sssd oddjobd
# systemctl enable sssd oddjobd
```

10. （可选）如果您的 LDAP 服务器使用已弃用的 TLS 1.0 或 TLS 1.1 协议，请将客户端系统上的系统范围加密策略切换到 LEGACY 级别，以允许 RHEL 使用这些协议进行通信：

```
# update-crypto-policies --set LEGACY
```

如需更多信息，请参阅红帽客户门户网站中的 RHEL 8 中的 强加密默认值和弱加密算法 知识库中的文章以及 **update-crypto-policies(8)**手册页。

验证步骤

- 验证您可以使用 **id** 命令和指定 LDAP 用户从 LDAP 服务器检索用户数据：

```
# id ldap_user
uid=17388(ldap_user) gid=45367(sysadmins)
groups=45367(sysadmins),25395(engineers),10(wheel),1202200000(admins)
```

系统管理员现在可以使用 **id** 命令从 LDAP 查询用户。该命令返回一个正确的用户 ID 和组群成员。

第 6 章 其他身份和身份验证供应商配置

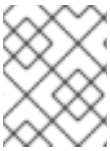
系统安全服务后台程序 (SSSD) 是一种用于访问远程目录和身份验证机制的系统服务。SSSD 的主要配置文件是 **/etc/sssd/sssd.conf**。以下章节概述了如何通过修改 **/etc/sssd/sssd.conf** 文件来配置 SSSD 服务和域：

- 调整 SSSD 如何解析并打印完整用户名，以启用离线身份验证。
- 配置 DNS 服务发现、简单访问提供程序规则和 SSSD 以应用 LDAP 访问过滤器。

6.1. 调整 SSSD 如何解释完整用户名

SSSD 将完整的用户名字符串解析到用户名和域组件中。默认情况下，SSSD 根据 Python 语法中的以下正则表达式，以 **user_name@domain_name** 格式解释完整的用户名：

```
(?P<name>[^@]+)@?(?P<domain>[^@]*$)
```



注意

对于 Identity Management 和 Active Directory 提供程序，默认的用户名格式为 **user_name@domain_name** 或 **NetBIOS_name\user_name**。

您可以通过在 **/etc/sssd/sssd.conf** 文件中添加 **re_expression** 选项并定义自定义正则表达式来调整 SSSD 如何解释完整的用户名。

- 要全局定义正则表达式，请将正则表达式添加到 **sssd.conf** 文件的 **[sssd]** 部分，如在全局范围内定义正则表达式所示。
- 要定义特定域的正则表达式，请将正则表达式添加到 **sssd.conf** 文件的对应域部分（例如 **[domain/LDAP]**），如为特定域定义正则表达式示例所示。

先决条件

- **root** 访问权限

步骤

1. 打开 **/etc/sssd/sssd.conf** 文件：
2. 使用 **re_expression** 选项定义自定义正则表达式。

例 6.1. 在全局范围内定义正则表达式

要全局定义所有域的正则表达式，请将 **re_expression** 添加到 **sssd.conf** 文件的 **[sssd]** 部分：

您可以使用以下全局表达式来定义 **domain\\username** 或 **domain@username** 的格式：

```
[sssd]
[... file truncated ...]
re_expression = (?P<domain>[^\\]*?)\\?(?P<name>[^\\]+$)
```

例 6.2. 定义特定域的正则表达式

要单独为特定域定义正则表达式，请将 **re_expression** 添加到 **sssd.conf** 文件的对应域部分：

您可以使用以下全局表达式来定义 LDAP 域的 **domain\\username** 或 **domain@username** 格式的用户名：

```
[domain/LDAP]
[... file truncated ...]
re_expression = (?P<domain>[^\\]*?)\\?(?P<name>[^\\]+$)
```

如需了解更多详细信息，请参阅 **sssd.conf(5)** 手册页中的 **SPECIAL SECTIONS** 和 **DOMAIN SECTIONS** 部分的对 **re_expression** 的描述。

6.2. 调整 SSSD 如何打印完整用户名

如果在 **/etc/sssd/sssd.conf** 文件中启用了 **use_fully_qualified_names** 选项，SSSD 会默认根据以下扩展以 **name@domain** 格式打印完整的用户名：

```
%1$s@%2$s
```

注意

如果未为受信任的域设置 **use_fully_qualified_names**，或者明确设置为 **false**，则仅打印没有域部分的用户名。

您可以通过在 **/etc/sssd/sssd.conf** 文件中添加 **full_name_format** 选项并定义自定义扩展来调整 SSSD 显示完整用户名的格式。

先决条件

- **root** 访问权限

步骤

1. 以 **root** 身份，打开 **/etc/sssd/sssd.conf** 文件。
2. 要为所有域定义全局扩展，请将 **full_name_format** 添加到 **sssd.conf** 的 **[sssd]** 部分：

   ```
   [sssd]
   [... file truncated ...]
   full_name_format = %1$s@%2$s
   ```

   在这种情况下，用户名显示为 **user@domain.test**。
3. 要定义特定域的用户名打印格式，请将 **full_name_format** 添加到 **sssd.conf** 的相应域部分。
   - 要使用 **%2$s\%1$s** 为活动目录(AD)域配置扩展：

```
[domain/ad.domain]
[... file truncated ...]
full_name_format = %2$s\%1$s
```

在本例中，用户名显示为 **ad.domain\user**。

- 要使用 **%3$s\%1$s** 为活动目录(AD)域配置扩展：

```
[domain/ad.domain]
[... file truncated ...]
full_name_format = %3$s\%1$s
```

在这种情况下，如果活动目录域的扁平域名被设置为 **AD**，则用户名显示为 **AD\user**。

如需了解更多详细信息，请参阅 **sssd.conf(5)** 手册页中的 **SPECIAL SECTIONS** 和 **DOMAIN SECTIONS** 部分的 **full_name_format** 的说明。

注意

SSSD 可在某些名称配置中剥离名称的域组件，这可能会导致身份验证错误。如果将 **full_name_format** 设置为非标准值，您会收到警告提示您将其更改为标准格式。

6.3. 启用离线验证

默认情况下，SSSD 不缓存用户凭证。在处理身份验证请求时，SSSD 始终联系身份提供程序。如果提供商不可用，用户身份验证会失败。

为确保在身份提供程序不可用时用户也可以被验证，在 **/etc/sssd/sssd.conf** 文件中将 **cache_credentials** 设置为 **true** 来启用凭证缓存。如果使用双因素身份验证，缓存的凭证引用密码和第一个身份验证因素。请注意，对于 passkey 和智能卡身份验证，您不需要将 **cache_credentials** 设置为 true 或设置任何附加配置；只要在缓存中记录成功在线身份验证，它们应该离线。

重要

SSSD 永远不会以纯文本形式缓存密码。它仅存储密码的哈希。

虽然凭证被存储为 salted SHA-512 哈希，但如果攻击者试图访问缓存文件并使用暴力攻击破解密码，这可能会带来安全风险。访问缓存文件需要特权访问权限，这是 RHEL 中的默认要求。

先决条件

- **root** 访问权限

步骤

1. 打开 **/etc/sssd/sssd.conf** 文件：
2. 在 domain 部分中，添加 **cache_credentials = true** 设置：

```
[domain/your-domain-name]
cache_credentials = true
```

3. *可选，但推荐使用*：为 SSSD 在身份提供程序不可用时允许离线身份验证的时间限制：

a. 配置 PAM 服务以使用 SSSD。
如需了解更多详细信息，请参阅使用 authselect 配置用户身份验证。

b. 使用 **offline_credentials_expiration** 选项来指定时间限制。
请注意，限制以天数为单位。

例如，要指定用户在上一次成功登录后 3 天可以离线验证，请使用：

```
[pam]
offline_credentials_expiration = 3
```

其他资源

- **sssd.conf(5)** 手册页

6.4. 配置 DNS 服务发现

DNS 服务发现使应用程序能够检查给定域中特定类型的特定服务的 SRV 记录，然后返回与所需类型匹配的服务器。如果在 **/etc/sssd/sssd.conf** 文件中未明确定义身份或身份验证服务器，SSSD 可以使用 DNS 服务发现动态发现服务器。

例如，如果 **sssd.conf** 包含 **id_provider = ldap** 设置，但是 **ldap_uri** 选项没有指定任何主机名或 IP 地址，SSSD 会使用 DNS 服务发现来动态发现服务器。

注意

SSSD 无法动态发现备份服务器，只有主服务器。

先决条件

- **root** 访问权限

步骤

1. 打开 **/etc/sssd/sssd.conf** 文件：

2. 将主服务器值设置为 **_srv_**。
对于 LDAP 供应商，使用 **ldap_uri** 选项设置主服务器：

```
[domain/your-domain-name]
id_provider = ldap
ldap_uri = _srv_
```

3. 设置服务类型，在密码更改供应商中启用服务发现：

```
[domain/your-domain-name]
id_provider = ldap
ldap_uri = _srv_

chpass_provider = ldap
ldap_chpass_dns_service_name = ldap
```

4. *可选：*默认情况下，服务发现使用系统主机名的域部分作为域名。要使用不同的 DNS 域，请使用 **dns_discovery_domain** 选项指定域名。

5. *可选：*默认情况下，针对 LDAP 服务类型的服务发现扫描。要使用不同的服务类型，请使用 **ldap_dns_service_name** 选项指定类型。

6. *可选：*默认情况下，SSSD 尝试查找 IPv4 地址。如果尝试失败，SSSD 会尝试查找 IPv6 地址。要自定义此行为，请使用 **lookup_family_order** 选项。

7. 对于您要使用服务发现的每个服务，在 DNS 服务器中添加 DNS 记录：

```
_service._protocol._domain TTL priority weight port host_name
```

其他资源

- RFC 2782，DNS 服务发现
- **sssd.conf(5)** 手册页

6.5. 配置简单的访问提供程序规则

simple 访问提供程序会基于用户名或组允许或拒绝访问。它可让您限制对特定机器的访问。

例如，您可以使用 **simple** 访问供应商限制对特定用户或组的访问。即使他们针对配置的身份验证提供程序成功进行身份验证，也不允许其他用户或组登录。

先决条件

- **root** 访问权限

步骤

1. 打开 **/etc/sssd/sssd.conf** 文件：

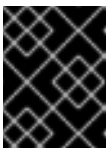
2. 将 **access_provider** 选项设置为 **simple** ：

```
[domain/your-domain-name]
access_provider = simple
```

3. 为用户定义访问控制规则。

   a. 要允许访问用户，请使用 **simple_allow_users** 选项。

   b. 若要拒绝用户访问，可使用 **simple_deny_users** 选项。



重要

如果您拒绝对特定用户的访问，则会自动允许对所有其他用户的访问。因此，允许访问特定用户通常被认为比拒绝对特定用户的访问更安全。

4. 定义组的访问控制规则。选择以下任意一项：

   a. 若要允许访问组，可使用 **simple_allow_groups** 选项。

b. 若要拒绝对组的访问，可使用 **simple_deny_groups** 选项。

重要

如果您拒绝访问特定组，则会自动允许访问其他任何组。因此，允许访问特定组通常被认为比拒绝对特定组的访问更安全。

例 6.3. 允许访问特定用户和组

以下示例允许访问 user1、user2 和 group1 的成员，同时拒绝对所有其他用户的访问：

```
[domain/your-domain-name]
access_provider = simple
simple_allow_users = user1, user2
simple_allow_groups = group1
```

重要

将拒绝列表保留为空可能会导致允许任何人访问。

注意

如果您要将一个可信 AD 用户添加到 **simple_allow_users** 列表中，请确保使用完全限定域名(FQDN)格式，例如 aduser@ad.example.com。由于不同域中的短名称可以相同，因此这防止出现访问控制配置的问题。

其他资源

- **sssd-simple** 手册页

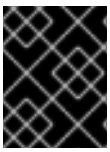
6.6. 配置 SSSD 以应用 LDAP 访问过滤器

如果在 **/etc/sssd/sssd.conf** 中设置 **access_provider** 选项，SSSD 会使用指定的访问提供程序来评估哪些用户被授予系统访问权限。如果您正在使用的访问提供商是 LDAP 提供商类型的扩展，您也可以指定一个用户必须匹配的 LDAP 访问控制过滤器，以允许访问系统。

例如，当使用 Active Directory (AD) 服务器作为访问提供程序时，您可以将 Linux 系统的访问权限限制为指定的 AD 用户。与指定过滤器不匹配的所有其他用户的访问都被拒绝。

注意

访问过滤器仅应用于 LDAP 用户条目。因此，在嵌套组上使用这种类型的访问控制可能无法正常工作。要在嵌套的组中应用访问控制，请参阅配置 **simple** 访问提供程序规则。



重要

在使用脱机缓存时，SSSD 会检查用户最近的在线登录尝试是否成功。在最近一次在线登录期间成功登录的用户仍将能够脱机登录，即使他们与访问过滤器不匹配。

先决条件

- **root** 访问权限

步骤

1. 打开 **/etc/sssd/sssd.conf** 文件：
2. 在 **[domain]** 部分中，指定 LDAP 访问控制过滤器。
   - 对于 LDAP 访问提供程序，请使用 **ldap_access_filter** 选项。详情请查看 **sssd-ldap(5)** 手册页。
   - 对于 AD 访问提供程序，请使用 **ad_access_filter** 选项。详情请查看 **sssd-ad(5)** 手册页。

     例 6.4. 允许访问特定 AD 用户

     例如，要只允许对属于 **admins** 用户组且具有 **unixHomeDirectory** 属性集的 AD 用户进行访问，请使用：

     ```
     [domain/your-AD-domain-name]
     access provider = ad
     [... file truncated ...]
     ad_access_filter = (&(memberOf=cn=admins,ou=groups,dc=example,dc=com)
     (unixHomeDirectory=*))
     ```

SSSD 也可以根据条目中的 **authorizedService** 或 **host** 属性检查结果。实际上，可以根据用户条目和配置评估所有选项 MDASH LDAP 过滤器，**authorizedService** 和 **host** MDASH。**ldap_access_order** 参数列出所有要使用的访问控制方法，按照应如何评估它们进行排序。

```
[domain/example.com]
access_provider = ldap
ldap_access_filter = memberOf=cn=allowedusers,ou=Groups,dc=example,dc=com
ldap_access_order = filter, host, authorized_service
```

其他资源

- **sssd-ldap(5)** 手册页

第 7 章 SSSD 客户端侧的视图

SSSD 提供 **sss_override** 工具，允许您创建一个本地视图，显示特定于本地机器的 POSIX 用户或组属性的值。您可以为所有 **id_provider** 值配置覆盖，但 **ipa** 除外。

如果您使用 **ipa** 提供程序，请在 IPA 中集中定义 ID 视图。如需更多信息，请参阅 使用 ID 视图覆盖 IdM 客户端中的用户属性值。

有关对 SSSD 性能的潜在负面影响的信息，请参阅 对 SSSD 性能 ID 视图的负面影响 。

7.1. 覆盖 LDAP USERNAME 属性

作为管理员，您可以将现有主机配置为使用 LDAP 中的帐户。但是，LDAP 中用户（名称、UID、GID、主目录、shell）的值与本地系统中的值不同。您可以按照以下步骤定义二级 **username** 来覆盖 LDAP **username** 属性。

先决条件

- **root** 访问权限
- 已安装 **sssd-tools**

步骤

1. 显示用户的当前信息：

   ```
   # id username
   ```

   使用用户名替换 *username*。

2. 添加二级 **username**：

   ```
   # sss_override user-add username -n secondary-username
   ```

   使用用户的名称替换 *username*，并使用新 **username** 替换 *secondary-username*。

3. 使用 **sss_override user-add** 命令创建第一次覆盖后，重启 SSSD 以使更改生效：

   ```
   # systemctl restart sssd
   ```

验证步骤

- 验证是否添加了新的 **username**：

  ```
  # id secondary-username
  ```

- *可选*。显示用户的覆盖：

  ```
  # sss_override user-show user-name
  user@ldap.example.com:secondary-username::::::
  ```

  例 7.1. 定义二级用户名

为用户 *sjones* 添加一个二级 **username** *sarah*。

1. 显示用户 *sjones* 的当前信息：

```
# id sjones
uid=1001(sjones) gid=6003 groups=6003,10(wheel)
```

2. 添加二级 **username**：

```
# sss_override user-add sjones -n sarah
```

3. 验证新 **username** 已添加并正确覆盖用户显示：

```
# id sarah
uid=1001(sjones) gid=6003(sjones) groups=6003(sjones),10(wheel)

# sss_override user-show sjones
user@ldap.example.com:sarah::::::
```

其他资源

- **sss_override** man page

7.2. 覆盖 LDAP UID 属性

作为管理员，您可以将现有主机配置为使用 LDAP 中的帐户。但是，LDAP 中用户（名称、UID、GID、主目录、shell）的值与本地系统中的值不同。您可以按照以下步骤定义不同的 UID 来覆盖 LDAP UID 属性。

先决条件

- **root** 访问权限
- 已安装 **sssd-tools**

步骤

1. 显示用户当前的 UID：

```
# id -u user-name
```

使用用户名称替换 *user-name*。

2. 覆盖用户帐户的 UID：

```
# sss_override user-add user-name -u new-UID
```

使用用户名替换 *user-name*，再将 *new-UID* 替换为新的 UID 号。

3. 使内存缓存过期：

```
# sss_cache --users
```

4. 使用 **sss_override user-add** 命令创建第一次覆盖后，重启 SSSD 以使更改生效：

```
# systemctl restart sssd
```

验证步骤

- 验证新 UID 是否已应用：

```
# id -u user-name
```

- *可选*。显示用户的覆盖：

```
# sss_override user-show user-name
user@ldap.example.com::new-UID:::::
```

例 7.2. 覆盖用户的 UID

使用 UID *6666* 覆盖用户 *sarah* 的 UID：

1. 显示 *sarah* 用户的当前 UID：

```
# id -u sarah
1001
```

2. 使用 UID *6666* 覆盖用户 *sarah* 的帐户的 UID：

```
# sss_override user-add sarah -u 6666
```

3. 手动使内存缓存过期：

```
# sss_cache --users
```

4. 重启 SSSD 以使更改生效：

```
# systemctl restart sssd
```

5. 验证是否应用了新的 UID，并正确覆盖用户显示：

```
# id sarah
6666

# sss_override user-show sarah
user@ldap.example.com::6666:::::
```

其他资源

- **sss_override** man page

7.3. 覆盖 LDAP GID 属性

作为管理员，您可以将现有主机配置为使用 LDAP 中的帐户。但是，LDAP 中用户（名称、UID、GID、主目录、shell）的值与本地系统中的值不同。您可以按照以下步骤定义不同的 GID 来覆盖 LDAP GID 属性。

先决条件

- **root** 访问权限
- 已安装 **sssd-tools**

步骤

1. 显示用户当前的 GID：

```
# id -g user-name
```

使用用户名称替换 *user-name*。

2. 覆盖用户帐户的 GID：

```
# sss_override user-add user-name -g new-GID
```

使用用户名替换 *user-name*，并使用新的 GID 号替换 *new-GID*。

3. 使内存缓存过期：

```
# sss_cache --users
```

4. 使用 **sss_override user-add** 命令创建第一次覆盖后，重启 SSSD 以使更改生效：

```
# systemctl restart sssd
```

验证步骤

- 验证是否应用了新的 GID：

```
# id -g user-name
```

- *可选*。显示用户的覆盖：

```
# sss_override user-show user-name
user@ldap.example.com:::6666::::
```

例 7.3. 覆盖用户的 GID

使用 GID *6666* 覆盖用户 *sarah* 的 GID：

1. 显示用户 *sarah* 的当前 GID：

```
# id -g sarah
6003
```

2. 使用 GID *6666* 覆盖用户 *sarah* 的帐户的 GID：

```
# sss_override user-add sarah -g 6666
```

3. 手动使内存缓存过期：

```
# sss_cache --users
```

4. 如果这是您的第一次覆盖，重启 SSSD 以使更改生效：

```
# systemctl restart sssd
```

5. 验证是否应用了新的 GID 并正确覆盖用户显示：

```
# id -g sarah
6666

# sss_override user-show sarah
user@ldap.example.com::6666:::::
```

其他资源

- **sss_override** man page

7.4. 覆盖 LDAP 主目录属性

作为管理员，您可以将现有主机配置为使用 LDAP 中的帐户。但是，LDAP 中用户（名称、UID、GID、主目录、shell）的值与本地系统中的值不同。您可以按照以下步骤定义不同的主目录来覆盖 LDAP 主目录属性。

先决条件

- **root** 访问权限
- 已安装 **sssd-tools**

步骤

1. 显示用户的当前主目录：

```
# getent passwd user-name
user-name:x:XXXX:XXXX::/home/home-directory:/bin/bash
```

使用用户名称替换 *user-name*。

2. 覆盖用户的主目录：

```
# sss_override user-add user-name -h new-home-directory
```

使用用户名替换 *user-name*，并使用新的主目录替换 *new-home-directory*。

3. 重启 SSSD 以使更改生效：

```
# systemctl restart sssd
```

验证步骤

- 验证是否定义了新主目录：

```
# getent passwd user-name
user-name:x:XXXX:XXXX::/home/new-home-directory:/bin/bash
```

- *可选*。显示用户的覆盖：

```
# sss_override user-show user-name
user@ldap.example.com:::::::new-home-directory::
```

例 7.4. 覆盖用户的主目录

使用 *admin* 覆盖 *sarah* 用户的主目录：

1. 显示 *sarah* 用户的当前主目录：

```
# getent passwd sarah
sarah:x:1001:6003::sarah:/bin/bash
```

2. 使用新主目录 *admin* 覆盖 *sarah* 用户的主目录：

```
# sss_override user-add sarah -h admin
```

3. 重启 SSSD 以使更改生效：

```
# systemctl restart sssd
```

4. 验证新主目录是否已定义，并正确覆盖用户显示：

```
# getent passwd sarah
sarah:x:1001:6003::admin:/bin/bash

# sss_override user-show user-name
user@ldap.example.com:::::::admin::
```

其他资源

- **sss_override** man page

7.5. 覆盖 LDAP SHELL 属性

作为管理员，您可以将现有主机配置为使用 LDAP 中的帐户。但是，LDAP 中用户（名称、UID、GID、主目录、shell）的值与本地系统中的值不同。您可以通过按照以下步骤定义不同的 shell 来覆盖 LDAP shell 属性。

先决条件

- **root** 访问权限
- 已安装 **sssd-tools**

步骤

1. 显示用户当前的 shell：

```
# getent passwd user-name
user-name:x:XXXX:XXXX::/home/home-directory:/bin/bash
```

使用用户名称替换 *user-name*。

2. 覆盖用户的 shell：

```
# sss_override user-add user-name -s new-shell
```

使用用户名替换 *user-name*，并使用新 shell 替换 *new-shell*。

3. 重启 SSSD 以使更改生效：

```
# systemctl restart sssd
```

验证步骤

- 验证是否定义了新 shell：

```
# getent passwd user-name
user-name:x:XXXX:XXXX::/home/home-directory:new-shell
```

- *可选*。显示用户的覆盖：

```
# sss_override user-show user-name
user@ldap.example.com::::::new-shell:
```

例 7.5. 覆盖用户的 shell

将用户 *sarah* 的 shell 从 **/bin/bash** 改为 **sbin/nologin** ：

1. 显示用户 *sarah* 的当前 shell：

```
# getent passwd sarah
sarah:x:1001:6003::sarah:/bin/bash
```

2. 使用新的 **/sbin/nologin** shell 覆盖用户 *sarah* 的 shell：

```
# sss_override user-add sarah -s /sbin/nologin
```

3. 重启 SSSD 以使更改生效：

```
# systemctl restart sssd
```

4. 验证新 shell 是否已定义并正确覆盖用户显示：

```
# getent passwd sarah
sarah:x:1001:6003::sarah:/sbin/nologin

# sss_override user-show user-name
user@ldap.example.com::::::/sbin/nologin:
```

其他资源

- **sss_override** man page

7.6. 列出主机上的覆盖

作为管理员，您可以列出主机上的所有用户和组覆盖，以验证是否已覆盖正确的属性。

先决条件

- **root** 访问权限
- 已安装 **sssd-tools**

步骤

- 列出所有用户覆盖：

```
# sss_override user-find
user1@ldap.example.com::8000::::/bin/zsh:
user2@ldap.example.com::8001::::/bin/bash:
...
```

- 列出所有组覆盖：

```
# sss_override group-find
group1@ldap.example.com::7000
group2@ldap.example.com::7001
...
```

7.7. 删除本地覆盖

如果要删除全局 LDAP 目录中定义的本地覆盖，请使用以下步骤：

先决条件

- **root** 访问权限
- 已安装 **sssd-tools**

步骤

- 要删除用户帐户的覆盖，请使用：

```
# sss_override user-del user-name
```

使用用户名称替换 *user-name*。更改会立即生效。

- 要为组群删除覆盖，请使用：

```
# sss_override group-del group-name
```

- 使用 **sss_override user-del** 或 **sss_override group-del** 命令删除第一次覆盖后，重启 SSSD 以使更改生效：

```
# systemctl restart sssd
```

当您为用户或组群删除覆盖时，此对象的所有覆盖都会被删除。

7.8. 导出和导入本地视图

您的本地覆盖保存在本地 SSSD 缓存中。您可以将用户和组覆盖从此缓存导出到文件，以创建备份。这样可确保即使缓存被清除，您也可以稍后恢复配置。

先决条件

- **root** 访问权限
- 已安装 **sssd-tools**

步骤

- 要备份用户和组视图，请使用：

```
# sss_override user-export /var/lib/sss/backup/sssd_user_overrides.bak
# sss_override group-export /var/lib/sss/backup/sssd_group_overrides.bak
```

- 要恢复用户和组视图，请使用：

```
# sss_override user-import /var/lib/sss/backup/sssd_user_overrides.bak
# sss_override group-import /var/lib/sss/backup/sssd_group_overrides.bak
```

第 8 章 配置 RHEL 主机以使用 AD 作为身份验证提供程序

作为系统管理员，您可以在不将主机加入 AD 的情况下，使用 Active Directory (AD) 作为 Red Hat Enterprise Linux (RHEL) 主机的身份验证供应商。

例如在以下情况下可以实现：

- 您不想向 AD 管理员授予对启用和禁用主机的控制权。
- 主机（可以是一个公司 PC）只表示被您公司中的某一用户使用。

重要

仅在很少情况下使用此方法。

考虑将系统完全加入 AD 或 Red Hat Identity Management (IdM)。将 RHEL 主机加入到域中可方便管理设置。如果您关注与将客户端直接加入 AD 中的客户端访问许可证，请考虑利用与 AD 信任协议中的 IdM 服务器。有关 IdM-AD 信任的更多信息，请参阅 规划 IdM 和 AD 之间的跨林信任 和 在 IdM 和 AD 之间安装信任 。

此流程可让名为 **AD_user** 的用户使用 **example.com** 域中的 Active Directory(AD)用户数据库中设置的密码登录到 **rhel_host** 系统。在这个示例中，**EXAMPLE.COM** Kerberos realm 对应于 **example.com** 域。

先决条件

- 有到 **rhel_host** 的 root 访问权限。
- **AD_user** 用户帐户存在于 **example.com** 域中。
- Kerberos realm 是 **EXAMPLE.COM**。
- **rhel_host** 尚未使用 **realm join** 命令加入到 AD。
- 已安装 **sssd-proxy** 软件包。

```
$ dnf install sssd-proxy
```

步骤

1. 在本地创建 **AD_user** 用户帐户而不为其分配密码：

```
# useradd AD_user
```

2. 打开 **/etc/nsswitch.conf** 文件进行编辑，并确保该文件包含以下行：

```
passwd:     sss files systemd
group:      sss files systemd
shadow:     files sss
```

3. 打开 **/etc/krb5.conf** 文件进行编辑，并确保该文件包含以下部分和项目：

```
# To opt out of the system crypto-policies configuration of krb5, remove the
# symlink at /etc/krb5.conf.d/crypto-policies which will not be recreated.
includedir /etc/krb5.conf.d/
```

```
[logging]
    default = FILE:/var/log/krb5libs.log
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kadmind.log

[libdefaults]
    dns_lookup_realm = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true
    rdns = false
    pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
    spake_preauth_groups = edwards25519
    default_realm = EXAMPLE.COM
    default_ccache_name = KEYRING:persistent:%{uid}

[realms]
 EXAMPLE.COM = {
     kdc = ad.example.com
     admin_server = ad.example.com
 }

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
```

4. 创建 **/etc/sssd/sssd.conf** 文件，并将以下部分和行插入到该文件中：

```
[sssd]
    services = nss, pam
    domains = EXAMPLE.COM

[domain/EXAMPLE.COM]
    id_provider = proxy
    proxy_lib_name = files
    auth_provider = krb5
    krb5_realm = EXAMPLE.COM
    krb5_server = ad.example.com
```

5. 更改 **/etc/sssd/sssd.conf** 文件的权限：

```
# chmod 600 /etc/sssd/sssd.conf
```

6. 启动安全性系统服务守护进程（SSSD）：

```
# systemctl start sssd
```

7. 启用 SSSD：

```
# systemctl enable sssd
```

8. 打开 **/etc/pam.d/system-auth** 文件，修改该文件使其包含以下部分和行：

```
# Generated by authselect on Wed May  8 08:55:04 2019
# Do not modify this file manually.

auth        required                                    pam_env.so
auth        required                                    pam_faildelay.so delay=2000000
auth        [default=1 ignore=ignore success=ok]        pam_succeed_if.so uid >= 1000 quiet
auth        [default=1 ignore=ignore success=ok]        pam_localuser.so
auth        sufficient                                  pam_unix.so nullok try_first_pass
auth        requisite                                   pam_succeed_if.so uid >= 1000 quiet_success
auth        sufficient                                  pam_sss.so forward_pass
auth        required                                    pam_deny.so

account     required                                    pam_unix.so
account     sufficient                                  pam_localuser.so
account     sufficient                                  pam_succeed_if.so uid < 1000 quiet
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     required                                    pam_permit.so

password    requisite                                   pam_pwquality.so try_first_pass local_users_only
password    sufficient                                  pam_unix.so sha512 shadow nullok try_first_pass
use_authtok
password    sufficient                                  pam_sss.so use_authtok
password    required                                    pam_deny.so

session     optional                                    pam_keyinit.so revoke
session     required                                    pam_limits.so
-session    optional                                    pam_systemd.so
session     [success=1 default=ignore]                  pam_succeed_if.so service in crond quiet
use_uid
session     required                                    pam_unix.so
session     optional                                    pam_sss.so
```

9. 将 **/etc/pam.d/system-auth** 文件的内容复制到 **/etc/pam.d/password-auth** 文件中。输入 **yes** 来确认覆盖文件的当前内容：

```
# cp /etc/pam.d/system-auth /etc/pam.d/password-auth
cp: overwrite '/etc/pam.d/password-auth'? yes
```

验证步骤

1. 为 **AD_user** 请求 Kerberos 票据 (TGT)。根据请求输入 **AD_user** 密码：

```
# kinit AD_user
Password for AD_user@EXAMPLE.COM:
```

2. 显示获得的 TGT：

```
# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: AD_user@EXAMPLE.COM

Valid starting     Expires            Service principal
11/02/20 04:16:38  11/02/20 14:16:38  krbtgt/EXAMPLE.COM@EXAMPLE.COM
 renew until 18/02/20 04:16:34
```

AD_user 已使用 **EXAMPLE.COM** Kerberos 域中的凭据成功登录 **rhel_host**。

第 9 章 使用 SSSD 报告主机的用户访问权限

安全系统服务守护进程 (SSSD) 跟踪用户可以或无法访问哪些客户端。本章论述了使用 **sssctl** 工具创建访问控制报告并显示用户数据。

先决条件

- SSSD 软件包安装在网络环境中

9.1. SSSCTL 命令

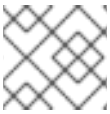
sssctl 是一个命令行工具，提供获取安全系统服务守护进程 (SSSD) 状态信息的统一方法。

您可以使用 **sssctl** 工具收集有关的信息：

- 域状态
- 客户端用户身份验证
- 用户对特定域的客户端的访问
- 有关缓存的内容的信息

使用 **sssctl** 工具，您可以：

- 管理 SSSD 缓存
- 管理日志
- 检查配置文件



注意

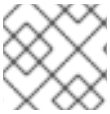
sssctl 工具取代了 **sss_cache** 和 **sss_debuglevel** 工具。

其他资源

- **sssctl --help**

9.2. 使用 SSSCTL 生成访问控制报告

您可以列出应用到您要运行报告的机器的访问控制规则，因为 SSSD 控制哪些用户可以登录到客户端。



注意

访问报告不准确，因为这个工具不会跟踪由密钥发布中心（KDC）锁定的用户。

先决条件

- 您必须使用管理员权限登录
- **ssctl** 工具可用于 RHEL 7、RHEL 8 和 RHEL 9 系统。

步骤

- 要为 **idm.example.com** 域生成报告，请输入：

```
[root@client1 ~]# sssctl access-report idm.example.com
1 rule cached

Rule name: example.user
 Member users: example.user
 Member services: sshd
```

9.3. 使用 SSSCTL 显示用户授权详情

sssctl user-checks 命令有助于调试使用系统安全服务守护进程 (SSSD) 进行用户查找、身份验证和授权的应用中的问题。

sssctl user-checks [USER_NAME] 命令显示通过 Name Service Switch (NSS) 获取的用户数据，以及 D-Bus 接口的 InfoPipe responseer。显示的数据显示用户是否被授权使用 **system-auth** 可插拔验证模块 (PAM) 服务登录。

命令有两个选项：

- **-a** 用于 PAM 操作
- **-s** 用于 PAM 服务

如果没有定义 **-a** 和 **-s** 选项，**sssctl** 会使用默认选项：**-a acct -s system-auth**。

先决条件

- 您必须使用管理员权限登录
- **ssctl** 工具可用于 RHEL 7、RHEL 8 和 RHEL 9 系统。

步骤

- 要显示特定用户的用户数据，请输入：

```
[root@client1 ~]# sssctl user-checks -a acct -s sshd example.user
user: example.user
action: acct
service: sshd
....
```

其他资源

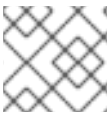
- **sssctl user-checks --help**

第 10 章 使用 SSSD 查询域信息

安全系统服务守护进程 (SSSD) 可以列出身份管理 (IdM) 中的域，以及 Active Directory 中的域，这些域通过跨林信任连接到 IdM。

10.1. 使用 SSSCTL 列出域

您可以使用 **sssctl domain-list** 命令来调试域拓扑的问题。



注意

这个状态可能立即不可用。如果该域不可见，请重复该命令。

先决条件

- 您必须使用管理员权限登录
- **ssctl** 工具可用于 RHEL 7、RHEL 8 和 RHEL 9 系统。

步骤

1. 要显示 sssctl 命令的帮助信息，请输入：

```
[root@client1 ~]# sssctl --help
....
```

2. 要显示可用域列表，请输入：

```
[root@client1 ~]# sssctl domain-list
implicit_files
idm.example.com
ad.example.com
sub1.ad.example.com
```

该列表包含 Active Directory 和 Identity Management 间的跨林信任域。

10.2. 使用 SSSCTL 验证域状态

您可以使用 **sssctl domain-status** 命令来调试域拓扑的问题。

注意

这个状态可能立即不可用。如果该域不可见，请重复该命令。

先决条件

- 您必须使用管理员权限登录
- **ssctl** 工具可用于 RHEL 7、RHEL 8 和 RHEL 9 系统。

步骤

1. 要显示 sssctl 命令的帮助信息，请输入：

```
[root@client1 ~]# sssctl --help
```

2. 要显示特定域的用户数据，请输入：

```
[root@client1 ~]# sssctl domain-status idm.example.com
Online status: Online

Active servers:
IPA: server.idm.example.com

Discovered IPA servers:
- server.idm.example.com
```

域 **idm.example.com** 在线，并在可应用命令的客户端可见。

如果域不可用，则结果为：

```
[root@client1 ~]# sssctl domain-status ad.example.com
Unable to get online status
```

第 11 章 使用 SSSD 限制 PAM 服务的域

可插拔验证模块 (PAM) 是身份验证和授权的通用框架。Red Hat Enterprise Linux 中的大多数系统应用程序依赖于底层 PAM 配置进行身份验证和授权。

系统安全服务守护进程 (SSSD) 可让您限制 PAM 服务可以访问哪些域。SSSD 根据运行特定 PAM 服务的用户评估来自 PAM 服务的身份验证请求。这意味着，如果 PAM 服务用户可以访问 SSSD 域，PAM 服务也可以访问该域。

11.1. 关于 PAM

可插拔验证模块 (PAM) 提供集中式身份验证机制，系统应用可以使用此机制将身份验证中继到集中配置的框架。

PAM 可插拔，因为存在用于不同类型身份验证源（如 Kerberos、SSSD、NIS 或本地文件系统）的 PAM 模块。您可以对不同的身份验证源进行优先排序。

此模块化架构为管理员提供了很大的灵活性来为系统设置身份验证策略。PAM 对开发人员和管理员而言是有用的系统，原因如下：

- PAM 提供一种常见身份验证方案，可用于各种应用。
- PAM 为系统管理员提供了对身份验证的显著灵活性和控制力。
- PAM 提供单个全文档库，使开发人员无需创建自己的身份验证方案即可编写程序。

11.2. 域访问限制选项

以下选项可以用来限制对所选域的访问：

/etc/sssd/sssd.conf 中的 pam_trusted_users

这个选项接受代表 SSSD 信任的 PAM 服务的数字 UID 或用户名列表。默认设置是 **all**，这意味着所有服务用户都是受信任的，可以访问任何域。

/etc/sssd/sssd.conf 中的 pam_public_domains

这个选项接受公共 SSSD 域列表。公共域是即使不可信 PAM 服务用户也可访问的域。选项也接受 **all** 和 **none** 值。默认值为 **none**，这意味着没有域是公共域，不受信任的服务用户无法访问任何域。

PAM 配置文件的 domains

此选项指定 PAM 服务可以对其进行身份验证的域列表。如果您在没有指定任何 **domains** 的情况下使用域，PAM 服务将无法对任何域进行身份验证，例如：

```
auth     required   pam_sss.so domains=
```

如果 PAM 配置文件使用 **domains**，则 PAM 服务能够在可信用户下运行时对所有域进行身份验证。

/etc/sssd/sssd.conf SSSD 配置文件中的 **domain** 选项还指定 SSSD 尝试验证的域列表。请注意，**PAM** 配置文件中的 domain 选项无法扩展 **sssd.conf** 中的域列表，它只能通过指定较短的列表来限制 **sssd.conf** 域列表。因此，如果在 PAM 文件中指定了域，但没有在 **sssd.conf** 中指定，则 PAM 服务无法对该域进行身份验证。

默认设置 **pam_trusted_users = all** 和 **pam_public_domains = none** 指定所有 PAM 服务用户都是可信并可访问任何域。将 **domain** 选项用于 PAM 配置文件会限制对域的访问。

使用 PAM 配置文件中的 **domains** 指定域，**sssd.conf** 包含 **pam_public_domains** 也需要在 **pam_public_domains** 中指定域。如果未包含所需域，**pam_public_domains** 选项将使 PAM 服务无法针对域进行身份验证，以防此服务在不受信任的用户下运行。

注意

PAM 配置文件中定义的域限制仅适用于身份验证操作，不适用于用户查找。

其他资源

- 有关 **pam_trusted_users** 和 **pam_public_domains** 选项的详情，请查看 **sssd.conf(5)** 手册页。
- 有关 PAM 配置文件中使用的 **domain** 选项的更多详细信息，请参阅 **pam_sss(8)man** page。

11.3. 限制 PAM 服务的域

此流程演示了如何针对域限制 PAM 服务身份验证。

先决条件

- SSSD 已安装并运行。

步骤

1. 配置 SSSD 以访问所需的域。在 **/etc/sssd/sssd.conf** 文件中的 **domain** 选项中定义 SSSD 可对其进行身份验证的域：

```
[sssd]
domains = domain1, domain2, domain3
```

2. 通过在 PAM 配置文件中设置 **domain** 选项来指定 PAM 服务可以进行身份验证的域。例如：

```
auth        sufficient    pam_sss.so forward_pass domains=domain1
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
password    sufficient    pam_sss.so use_authtok
```

在本例中，您将允许 PAM 服务仅对 **domain1** 进行身份验证。

验证步骤

- 根据 **domain1** 进行验证。它必须成功。

第 12 章 在本地 SSSD 配置中消除拼写错误

您可以使用 **sssctl config-check** 命令测试主机上的 **/etc/sssd/sssd.conf** 文件是否包含任何拼写错误。

先决条件

- 以 root 身份登录。
- **sssd-tools** 软件包已安装。

步骤

1. 输入 **sssctl config-check** 命令：

```
# sssctl config-check

Issues identified by validators: 1
[rule/allowed_domain_options]: Attribute 'ldap_search' is not allowed in section
'domain/example1'. Check for typos.

Messages generated during configuration merging: 0

Used configuration snippet files: 0
```

2. 打开 **/etc/sssd/sssd.conf** 文件并更正拼写错误。例如，如果您收到上一步中的出错信息，将 **ldap_search** 替换为 **ldap_search_base**:

```
[...]
[domain/example1]
ldap_search_base = dc=example,dc=com
[...]
```

3. 保存这个文件。
4. 重启 SSSD：

```
# systemctl restart sssd
```

验证步骤

- 输入 **sssctl config-check** 命令：

```
# sssctl config-check

Issues identified by validators: 0

Messages generated during configuration merging: 0

Used configuration snippet files: 0
```

/etc/sssd/sssd.conf 文件现在没有拼写错误。

第 13 章 IDM 中 SSSD 身份验证故障排除

在 Identity Management（IdM）环境中的身份验证涉及许多组件：

在 IdM 客户端中：

- SSSD 服务。
- Name Services Switch (NSS)。
- 可插拔验证模块 (PAM)。

在 IdM 服务器上：

- SSSD 服务。
- IdM 目录服务器。
- IdM Kerberos 密钥分发中心 (KDC)。

如果您要以 Active Directory (AD) 用户进行身份验证：

- AD 域控制器上的目录服务器。
- AD 域控制器上的 Kerberos 服务器。

要验证用户，您必须使用 SSSD 服务执行以下功能：

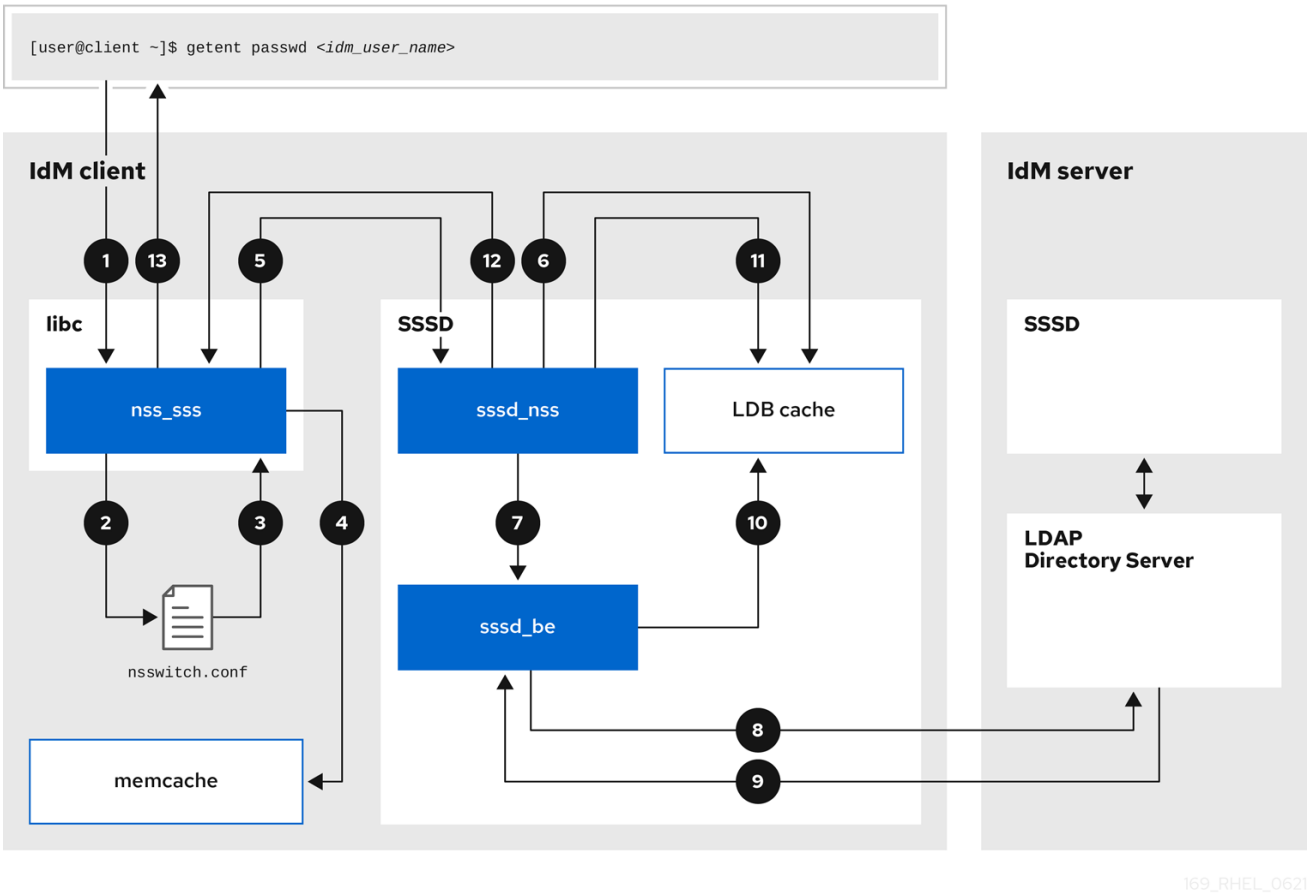
- 从身份验证服务器检索用户信息。
- 提示用户输入其凭据，将这些凭据传递到身份验证服务器，然后处理结果。

要了解更多有关 SSSD 服务和存储用户信息的服务器之间的信息流，以便可以排除您环境中身份验证尝试失败的问题，请参阅：

1. 使用 SSSD 获取 IdM 用户信息时的数据流
2. 使用 SSSD 获取 AD 用户信息时的数据流
3. 以 IdM 中的 SSSD 用户身份进行身份验证时的数据流
4. 缩小身份验证问题的范围
5. SSSD 日志文件和日志记录级别
6. 在 sssd.conf 文件中为 SSSD 启用详细日志记录
7. 使用 sssctl 命令为 SSSD 启用详细的日志记录
8. 从 SSSD 服务收集调试日志，对 IdM 服务器的身份验证问题进行故障排除
9. 从 SSSD 服务收集调试日志，以对 IdM 客户端的身份验证问题进行故障排除
10. 跟踪 SSSD 后端中的客户端请求
11. 使用日志分析器工具跟踪客户端请求

13.1. 使用 SSSD 获取 IDM 用户信息时的数据流

下图使用 **getent passwd *<idm_user_name>*** 命令在请求 IdM 用户信息的过程中简化 IdM 客户端和 IdM 服务器之间的信息流。



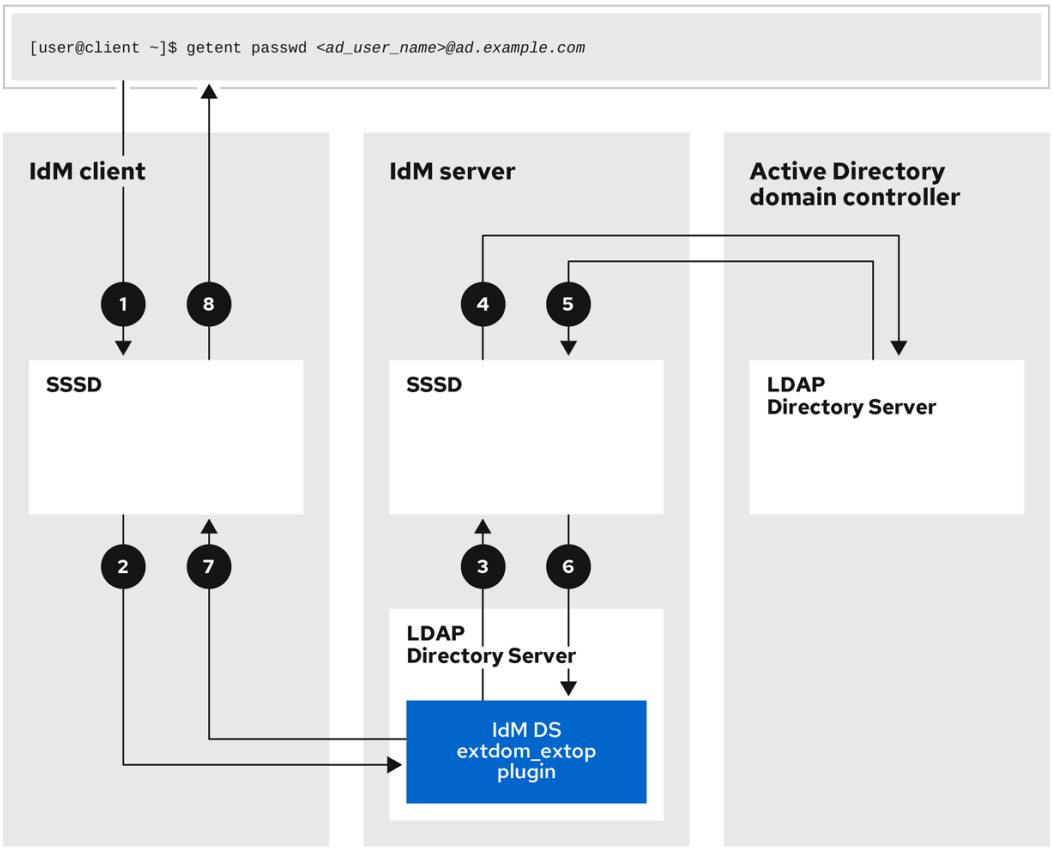
1. **getent** 命令会触发来自 **libc** 库的 **getpwnam** 调用。
2. **libc** 库引用 **/etc/nsswitch.conf** 配置文件来检查哪个服务负责提供用户信息，并发现 **SSSD** 服务的条目。
3. **libc** 库打开 **ss_sss** 模块。
4. nss_sss 模块检查内存映射缓存以获取用户信息。如果缓存中存在数据，则 **ss_sss** 模块会返回它。
5. 如果用户信息不在内存映射缓存中，则会将请求传递给 SSSD **sssd_nss** 响应程序进程。
6. SSSD 服务检查其缓存。如果缓存中存在数据并有效，**sssd_nss** 响应程序会从缓存中读取数据并将其返回到应用。
7. 如果缓存中没有数据或数据已过期，**sssd_nss** 响应器将查询相应的后端进程并等待回复。SSSD 服务在 IdM 环境中使用 IPA 后端，通过 **sssd.conf** 配置文件中的 **id_provider=ipa** 启用。
8. **sssd_be** 后端进程连接到 IdM 服务器，并从 IdM LDAP 目录服务器请求信息。
9. IdM 服务器上的 SSSD 后端响应 IdM 客户端上的 SSSD 后端进程。
10. 客户端上的 SSSD 后端将生成的数据存储在 SSSD 缓存中，并提醒已更新缓存的响应程序进程。

11. **sssd_nss** 前端响应器进程从 SSSD 缓存检索信息。
12. **sssd_nss** 响应器将用户信息发送到 **ss_sss** 响应者，以完成请求。
13. **libc** 库将用户信息返回到请求它的应用程序。

13.2. 使用 SSSD 获取 AD 用户信息时的数据流

如果您在 IdM 环境和活动目录(AD)域之间建立了跨林信任，则检索 IdM 客户端 AD 用户信息时，信息流与检索 IdM 用户信息时的信息流非常相似，只是多了一个联系 AD 用户数据库的步骤。

下图是当用户使用命令 **getent passwd *<ad_user_name@ad.example.com>*** 请求 AD 用户的信息时信息流的一种简化。这个图并没有包括使用 SSSD 检索 IdM 用户信息时的数据流 中讨论的内部详细信息。它侧重于 IdM 客户端上的 SSSD 服务、IdM 服务器上的 SSSD 服务和 AD 域控制器上的 LDAP 数据库之间的通信。



1. IdM 客户端为 AD 用户信息查找其本地 SSSD 缓存。
2. 如果 IdM 客户端没有用户信息，或者信息是 stale，客户端上的 SSSD 服务会联系 IdM 服务器上的 **extdom_extop** 插件来执行 LDAP 扩展操作并请求信息。
3. IdM 服务器上的 SSSD 服务在其本地缓存中查找 AD 用户信息。
4. 如果 IdM 服务器在其 SSSD 缓存中没有用户信息，或者其信息为过时，它将执行 LDAP 搜索，以从 AD 域控制器请求用户信息。
5. IdM 服务器上的 SSSD 服务从 AD 域控制器接收 AD 用户信息，并将其存储在其缓存中。
6. **extdom_extop** 插件从 IdM 服务器上的 SSSD 服务接收信息，该服务完成 LDAP 扩展操作。

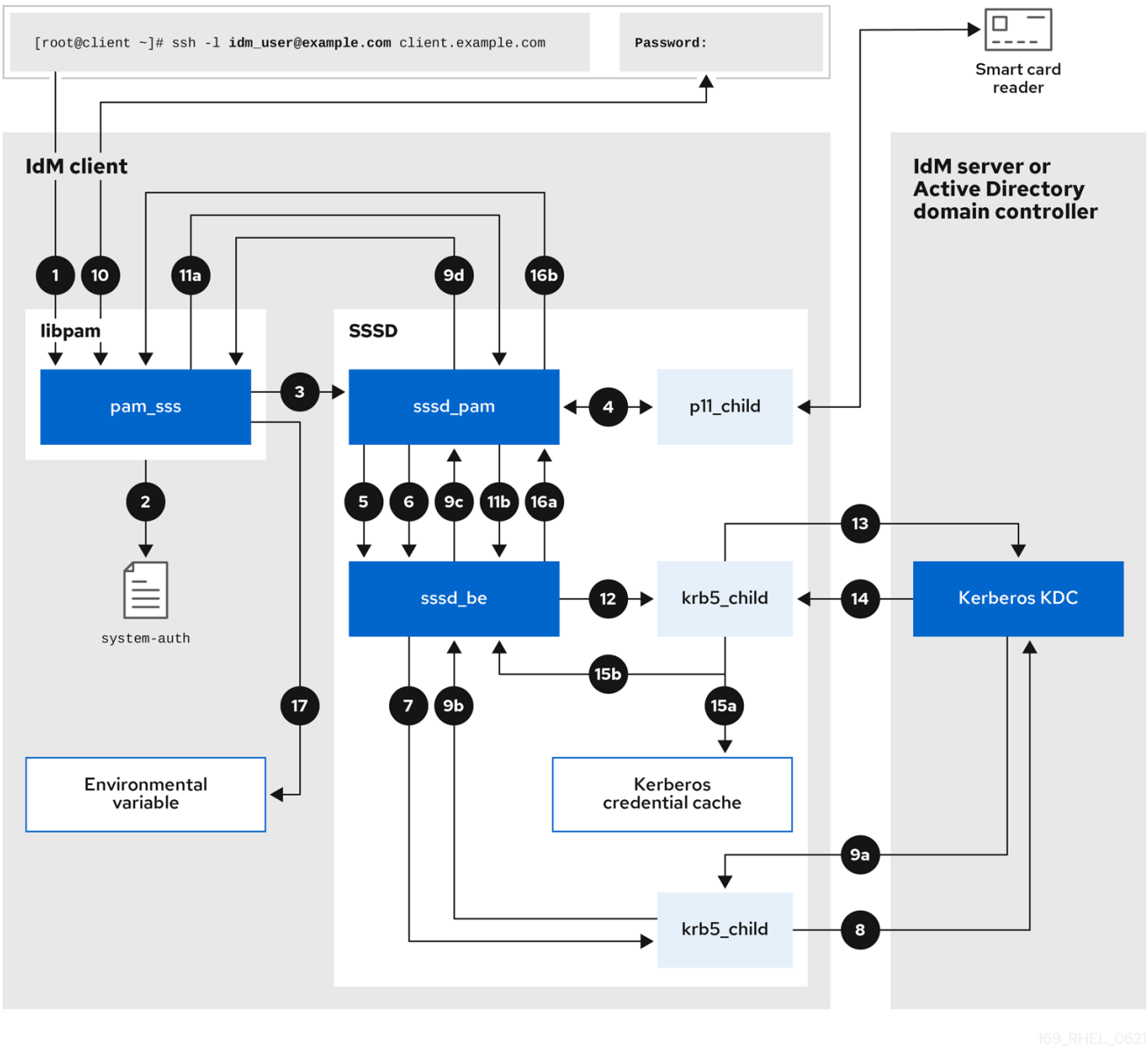
7. IdM 客户端上的 SSSD 服务从 LDAP 扩展操作接收 AD 用户信息。
8. IdM 客户端将 AD 用户信息存储在其 SSSD 缓存中，并将信息返回给请求它的应用程序。

13.3. 以 IDM 中的 SSSD 用户身份进行身份验证时的数据流

以 IdM 服务器或客户端中的用户身份进行身份验证涉及以下组件：

- 启动身份验证请求的服务，如 sshd 服务。
- 可插拔验证模块 (PAM) 库及其模块。
- SSSD 服务、其响应者和后端。
- 智能卡读取器（如果配置了智能卡验证）。
- 身份验证服务器：
  - IdM 用户通过 IdM Kerberos 密钥分发中心 (KDC) 进行身份验证。
  - Active Directory (AD) 用户通过 AD 域控制器 (DC) 进行身份验证。

下图是用户在尝试通过命令行上的 SSH 服务在本地登录主机期间需要进行身份验证时的简化信息流。



1. 使用 **ssh** 命令尝试身份验证会触发 **libpam** 库。

2. **libpam** 库引用 **/etc/pam.d/** 目录中与请求身份验证尝试的服务对应的 PAM 文件。在本例中，libpam 库涉及通过本地主机上的 SSH 服务进行身份验证，**libpam** 库检查 **/etc/pam.d/system-auth** 配置文件并发现 SSSD PAM 的 **pam_sss.so** 条目：

```
auth    sufficient    pam_sss.so
```

3. 要确定哪些身份验证方法可用，**libpam** 库会打开 **pam_sss** 模块，并将 **SSS_PAM_PREAUTH** 请求发送到 SSSD 服务的 **sssd_pam** PAM 响应者。

4. 如果配置了智能卡验证，SSSD 服务会生成一个临时 **p11_child** 进程，以检查智能卡并从中检索证书。

5. 如果为用户配置了智能卡验证，**sssd_pam** 响应程序会尝试将智能卡中的证书与用户匹配。**sssd_pam** 响应器还搜索用户所属的组，因为组成员身份可能会影响访问控制。

6. **sssd_pam** 响应程序将 **SSS_PAM_PREAUTH** 请求发送到 **sssd_be** 后端响应程序，以查看服务器支持的身份验证方法，如密码或双因素身份验证。在 IdM 环境中，SSSD 服务使用 IPA 响应器，默认的身份验证方法是 Kerberos。在本例中，用户使用简单的 Kerberos 密码进行身份验

证。

7. **sssd_be** 响应器生成一个临时 **krb5_child** 进程。
8. **krb5_child** 进程联系 IdM 服务器上的 KDC，并检查可用的身份验证方法。
9. KDC 响应请求：
   a. **krb5_child** 进程评估回复，并将结果发回到 **sssd_be** 后端进程。
   b. **sssd_be** 后端进程会收到结果。
   c. **sssd_pam** 响应器会收到结果。
   d. **pam_sss** 模块会收到结果。
10. 如果为用户配置了密码身份验证，**pam_sss** 模块将提示用户输入其密码。如果配置了智能卡验证，**pam_sss** 模块会提示用户输入其智能卡 PIN。
11. 模块会发送带有用户名和密码的 **SSS_PAM_AUTHENTICATE** 请求，该请求经过以下操作：
    a. **sssd_pam** 响应器。
    b. **sssd_be** 后端进程。
12. **sssd_be** 进程生成一个临时 **krb5_child** 进程来联系 KDC。
13. **krb5_child** 进程尝试使用用户提供的用户名和密码从 KDC 检索 Kerberos Ticket Granting Ticket (TGT)。
14. **krb5_child** 进程接收身份验证尝试的结果。
15. **krb5_child** 进程：
    a. 将 TGT 存储到凭据缓存中。
    b. 将身份验证结果返回到 **sssd_be** 后端进程。
16. 身份验证结果从 **sssd_be** 进程传输到：
    a. **sssd_pam** 响应器。
    b. **pam_sss** 模块。
17. **pam_sss** 模块使用用户 TGT 的位置设置环境变量，以便其他应用可以引用它。

13.4. 缩小身份验证问题的范围

要成功验证用户，您必须能够使用 SSSD 服务从存储用户信息的数据库检索用户信息。以下流程描述了测试身份验证流程的不同组件的步骤，以便您可以在用户无法登录时缩小身份验证问题的范围。

步骤

1. 验证 SSSD 服务及其进程是否正在运行。

```
[root@client ~]# pstree -a | grep sssd
  |-sssd -i --logger=files
```

```
|   |-sssd_be --domain implicit_files --uid 0 --gid 0 --logger=files
|   |-sssd_be --domain example.com --uid 0 --gid 0 --logger=files
|   |-sssd_ifp --uid 0 --gid 0 --logger=files
|   |-sssd_nss --uid 0 --gid 0 --logger=files
|   |-sssd_pac --uid 0 --gid 0 --logger=files
|   |-sssd_pam --uid 0 --gid 0 --logger=files
|   |-sssd_ssh --uid 0 --gid 0 --logger=files
|   `-sssd_sudo --uid 0 --gid 0 --logger=files
|-sssd_kcm --uid 0 --gid 0 --logger=files
```

2. 验证客户端可以通过 IP 地址联系用户数据库服务器。

```
[user@client ~]$ ping <IP_address_of_the_database_server>
```

如果此步骤失败，请检查您的网络和防火墙设置是否允许 IdM 客户端和服务器之间进行直接通信。请参阅使用和配置 firewalld。

3. 验证客户端可以通过完全限定的主机名发现并联系 IdM LDAP 服务器（适用于 IdM 用户）或 AD 域控制器（ AD 用户）。

```
[user@client ~]$ dig -t SRV _ldap._tcp.example.com @<name_server>
[user@client ~]$ ping <fully_qualified_host_name_of_the_server>
```

如果此步骤失败，请检查您的 Dynamic Name Service (DNS) 设置，包括 **/etc/resolv.conf** 文件。请参阅配置 DNS 服务器顺序。

注意

默认情况下，SSSD 服务会尝试通过 DNS 服务 (SRV) 记录自动发现 LDAP 服务器和 AD DC。另外，您可以通过在 **sssd.conf** 配置文件中设置以下选项，将 SSSD 服务限制为使用特定的服务器：

- **ipa_server = *<fully_qualified_host_name_of_the_server>***
- **ad_server = *<fully_qualified_host_name_of_the_server>***
- **ldap_uri = *<fully_qualified_host_name_of_the_server>***

如果使用这些选项，请验证您可以联系它们中列出的服务器。

4. 验证客户端是否可以对 LDAP 服务器进行身份验证，并使用 **ldapsearch** 命令检索用户信息。

   a. 如果您的 LDAP 服务器是 IdM 服务器，如 **server.example.com**，检索主机的 Kerberos 票据，并使用主机 Kerberos 主体进行身份验证数据库搜索：

   ```
   [user@client ~]$ kinit -k 'host/client.example.com@EXAMPLE.COM'
   [user@client ~]$ ldapsearch -LLL -Y GSSAPI -h server.example.com -b
   “dc=example,dc=com” uid=<user_name>
   ```

   b. 如果您的 LDAP 服务器是 Active Directory (AD) 域控制器 (DC)，如 **server.ad.example.com**，请检索主机的 Kerberos 票据，并使用主机 Kerberos 主体执行数据库搜索：

```
[user@client ~]$ kinit -k 'CLIENT$@AD.EXAMPLE.COM'
[user@client ~]$ ldapsearch -LLL -Y GSSAPI -h server.ad.example.com -b
“dc=example,dc=com” sAMAccountname=<user_name>
```

c. 如果您的 LDAP 服务器是普通 LDAP 服务器，且您在 **sssd.conf** 文件中设置了 **ldap_default_bind_dn** 和 **ldap_default_authtok** 选项，请验证是同一个 **ldap_default_bind_dn** 帐户：

```
[user@client ~]$ ldapsearch -xLLL -D "cn=ldap_default_bind_dn_value" -W -h
ldapserver.example.com -b “dc=example,dc=com” uid=<user_name>
```

如果此步骤失败，请验证您的数据库设置是否允许您的主机搜索 LDAP 服务器。

5. 由于 SSSD 服务使用 Kerberos 加密，因此请以无法登录的用户身份获得 Kerberos 票据。

a. 如果您的 LDAP 服务器是 IdM 服务器：

```
[user@client ~]$ kinit <user_name>
```

b. 如果 LDAP 服务器数据库是 AD 服务器：

```
[user@client ~]$ kinit <user_name@AD.EXAMPLE.COM>
```

如果此步骤失败，请验证您的 Kerberos 服务器是否正常运行，所有服务器都已同步其时间，并且用户帐户未被锁定。

6. 验证您是否可以检索有关命令行的用户信息。

```
[user@client ~]$ getent passwd <user_name>
[user@client ~]$ id <user_name>
```

如果这一步失败，请验证客户端上的 SSSD 服务是否可以接收用户数据库的信息：

a. 查看 **/var/log/messages** 日志文件中的错误。

b. 在 SSSD 服务中启用详细的日志记录，收集调试日志，并查看日志以确定问题的根源。

c. *（可选）*创建一个红帽技术支持问题单，并提供您收集的故障排除信息。

7. 如果您被允许在主机上运行 **sudo**，请使用 **sssctl** 工具验证用户是否被允许登录。

```
[user@client ~]$ sudo sssctl user-checks -a auth -s ssh <user_name>
```

如果这一步失败，请验证您的授权设置，如 PAM 配置、IdM HBAC 规则和 IdM RBAC 规则：

a. 确保用户的 UID 等于或大于 **UID_MIN**，它在 **/etc/login.defs** 文件中定义。

b. 查看 **/var/log/secure** 和 **/var/log/messages** 日志文件中的授权错误。

c. 在 SSSD 服务中启用详细的日志记录，收集调试日志，并查看日志以确定问题的根源。

d. *（可选）*创建一个红帽技术支持问题单，并提供您收集的故障排除信息。

其他资源

- 在 sssd.conf 文件中为 SSSD 启用详细日志记录
- 使用 sssctl 命令为 SSSD 启用详细的日志记录
- 从 SSSD 服务收集调试日志，对 IdM 服务器的身份验证问题进行故障排除
- 从 SSSD 服务收集调试日志，以对 IdM 客户端的身份验证问题进行故障排除

13.5. SSSD 日志文件和日志记录级别

每个 SSSD 服务都记录到 **/var/log/sssd/** 目录中自己的日志文件。对于 **example.com** IdM 域中的 IdM 服务器，其日志文件可能类似这样：

```
[root@server ~]# ls -l /var/log/sssd/
total 620
-rw-------. 1 root root      0 Mar 29 09:21 krb5_child.log
-rw-------. 1 root root  14324 Mar 29 09:50 ldap_child.log
-rw-------. 1 root root 212870 Mar 29 09:50 sssd_example.com.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_ifp.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_implicit_files.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd.log
-rw-------. 1 root root 219873 Mar 29 10:03 sssd_nss.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_pac.log
-rw-------. 1 root root  13105 Mar 29 09:21 sssd_pam.log
-rw-------. 1 root root   9390 Mar 29 09:21 sssd_ssh.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_sudo.log
```

13.5.1. SSSD 日志文件用途

krb5_child.log

Kerberos 身份验证中涉及的短期帮助程序进程的日志文件。

ldap_child.log

与 LDAP 服务器通信的简短帮助程序进程的日志文件，涉及获取 Kerberos 票据。

sssd_<example.com>.log

对于 **sssd.conf** 文件中的每个域部分，SSSD 服务会将与 LDAP 服务器通信的信息记录到单独的日志文件中。例如，在名为 **example.com** 的 IdM 域环境中，SSSD 服务将其信息记录到名为 **sssd_example.com.log** 的文件中。如果主机直接与名为 **ad.example.com** 的 AD 域集成，信息将记录到名为 **sssd_ad.example.com.log** 的文件中。

注意

如果您有一个 IdM 环境以及与 AD 域的跨林信任，则有关 AD 域的信息仍会记录到 IdM 域的日志文件中。

类似地，如果主机直接集成到 AD 域，则任何子域的信息都会写入到主域的日志文件中。

selinux_child.log

用于检索和设置 SELinux 信息的短生命帮助器进程的日志文件。

sssd.log

SSSD 监控并与其响应器和后端进程通信的日志文件。

sssd_ifp.log

InfoPipe 响应器的日志文件，它提供了一个可通过系统总线访问的公共 D-Bus 接口。

sssd_nss.log

用于检索用户和组信息的 Name Services Switch (NSS) 响应器的日志文件。

sssd_pac.log

Microsoft Privilege Attribute 证书 (PAC) 响应器的日志文件，从 AD Kerberos 票据收集 PAC，并从 PAC 中生成 AD 用户的信息，从而避免直接从 AD 请求它。

sssd_pam.log

可插拔验证模块 (PAM) 响应器的日志文件。

sssd_ssh.log

SSH 响应器进程的日志文件。

13.5.2. SSSD 日志记录级别

设置一个 debug 级别后，也会启用它以下的所有 debug 级别。例如，把 debug 级别设置为 6 后，也会启用 debug 级别 0 到 5。

表 13.1. SSSD 日志记录级别

级别	Description
0	**致命故障**。阻止 SSSD 服务启动或导致它终止的错误。
1	**关键故障**。错误没有导致 SSSD 服务被终止，但至少有一个主要功能无法正常工作。
2	**严重故障**。这个错误声明特定请求或操作失败。**这是默认的调试日志级别。**
3	**小故障**。在级别 2 中捕获的操作失败的错误。
4	**配置设置**。
5	**功能**数据。
6	跟踪**操作**功能的消息。
7	跟踪**内部控制**功能的消息。
8	**功能内部**变量的内容。
9	极**低级别跟踪**信息。

13.6. 在 SSSD.CONF 文件中为 SSSD 启用详细日志记录

默认情况下，SSSD 服务仅记录严重故障（调试级别 2），但它不会记录对身份验证问题进行故障排除所需的详情。

要在 SSSD 服务重启过程中永久启用详细的日志记录，请在 **/etc/sssd/sssd.conf** 配置文件的每个部分添加 **debug_level=*<integer>*** 选项，其中 ***<integer>*** 值是一个 0 到 9 之间的数字。debug 级别 0 到 3 会记录大错误的日志，级别 8 和更高级别会提供大量详细的日志消息。**级别 6 是调试身份验证问题的一个良好起点。**

先决条件

- 您需要 root 密码来编辑 **sssd.conf** 配置文件并重新启动 SSSD 服务。

步骤

1. 在文本编辑器中打开 **/etc/sssd/sssd.conf** 文件。
2. 将 **debug_level** 选项添加到文件的每个部分，并将 debug 级别设置为您选择的详细程度。

```
[domain/example.com]
debug_level = 6
id_provider = ipa
...

[sssd]
debug_level = 6
services = nss, pam, ifp, ssh, sudo
domains = example.com

[nss]
debug_level = 6

[pam]
debug_level = 6

[sudo]
debug_level = 6

[ssh]
debug_level = 6

[pac]
debug_level = 6

[ifp]
debug_level = 6
```

3. 保存并关闭 **sssd.conf** 文件。
4. 重启 SSSD 服务以加载新的配置设置。

```
[root@server ~]# systemctl restart sssd
```

其他资源

- SSSD 日志文件和日志记录级别

13.7. 使用 SSSCTL 命令为 SSSD 启用详细的日志记录

默认情况下，SSSD 服务仅记录严重故障（调试级别 2），但它不会记录对身份验证问题进行故障排除所需的详情。

您可以在命令行中使用 **sssctl debug-level *<integer>*** 命令更改 SSSD 服务的 debug 级别，其中 ***<integer>*** 是 0 到 9 之间的一个数字。debug 级别 0 到 3 会记录大错误的日志，级别 8 和更高级别会提供大量详细的日志消息。级别 6 是调试身份验证问题的一个良好起点。

先决条件

- 您需要 root 密码来运行 **sssctl** 命令。

步骤

- 使用 sssctl debug-level 命令将所选的调试级别设置为您所需的详细程度。

```
[root@server ~]# sssctl debug-level 6
```

其他资源

- SSSD 日志文件和日志记录级别

13.8. 从 SSSD 服务收集调试日志，对 IDM 服务器的身份验证问题进行故障排除

如果您在尝试以 IdM 用户身份对 IdM 服务器进行身份验证时遇到问题，请在服务器上的 SSSD 服务中启用详细的调试日志，并收集尝试检索用户信息的日志。

先决条件

- 您需要 root 密码来运行 **sssctl** 命令并重新启动 SSSD 服务。

步骤

1. 在 IdM 服务器上启用详细的 SSSD 调试日志。

```
[root@server ~]# sssctl debug-level 6
```

2. 对于遇到身份验证问题的用户，在 SSSD 缓存中使相关的对象无效，这样使您不会绕过 LDAP 服务器来从缓存的 SSSD 中获取信息。

```
[root@server ~]# sssctl cache-expire -u idmuser
```

3. 通过删除旧的 SSSD 日志来最大程度减少数据集的故障排除。

```
[root@server ~]# sssctl logs-remove
```

4. 尝试切换至遇到身份验证问题的用户，同时在尝试前后收集时间戳。这些时间戳进一步缩小了数据集的范围。

```
[root@server sssd]# date; su idmuser; date
Mon Mar 29 15:33:48 EDT 2021
su: user idmuser does not exist
```

```
Mon Mar 29 15:33:49 EDT 2021
```

5. *（可选）如果您不想继续收集详细的*SSSD 日志，请降低 debug 级别。

```
[root@server ~]# sssctl debug-level 2
```

6. 查看 SSSD 日志，了解失败请求的信息。例如，检查 **/var/log/sssd/sssd_example.com.log** 文件表明 SSSD 服务没有在 **cn=accounts,dc=example,dc=com** LDAP 子树中找到用户。这可能表示用户不存在，或者存在于其他位置。

```
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [dp_get_account_info_send] (0x0200):
Got request for [0x1][BE_REQ_USER][name=idmuser@example.com]
...
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sdap_get_generic_ext_step] (0x0400):
calling ldap_search_ext with [(&(uid=idmuser)(objectclass=posixAccount)(uid=)(&
(uidNumber=)(!(uidNumber=0))))][cn=accounts,dc=example,dc=com].
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sdap_get_generic_op_finished]
(0x0400): Search result: Success(0), no errmsg set
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sdap_search_user_process] (0x0400):
Search for users, returned 0 results.
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sysdb_search_by_name] (0x0400):
No such entry
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sysdb_delete_user] (0x0400): Error: 2
(No such file or directory)
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sysdb_search_by_name] (0x0400):
No such entry
(Mon Mar 29 15:33:49 2021) [sssd[be[example.com]]]
[ipa_id_get_account_info_orig_done] (0x0080): Object not found, ending request
```

7. 如果您无法确定导致身份验证问题的原因：

   a. 收集您最近生成的 SSSD 日志。

   ```
   [root@server ~]# sssctl logs-fetch sssd-logs-Mar29.tar
   ```

   b. 创建一个红帽技术支持问题单并提供：

      i. SSSD 日志：**sssd-logs-Mar29.tar**

      ii. 与日志对应的请求的控制台输出，包括时间戳和用户名：

      ```
      [root@server sssd]# date; id idmuser; date
      Mon Mar 29 15:33:48 EDT 2021
      id: ‘idmuser’: no such user
      Mon Mar 29 15:33:49 EDT 2021
      ```

13.9. 从 SSSD 服务收集调试日志，以对 IDM 客户端的身份验证问题进行故障排除

如果您在尝试以 IdM 用户身份向 IdM 客户端进行身份验证时遇到问题，请验证您是否可以检索有关 IdM 服务器的用户信息。如果您无法检索有关 IdM 服务器的用户信息，您将无法在 IdM 客户端（其从 IdM 服务器检索信息）上检索它。

确认身份验证问题不源自 IdM 服务器后，从 IdM 服务器和 IdM 客户端收集 SSSD 调试日志。

先决条件

- 用户仅在 IdM 客户端而不是 IdM 服务器中存在身份验证问题。
- 您需要 root 密码来运行 **sssctl** 命令并重新启动 SSSD 服务。

步骤

1. **在客户端中：**在文本编辑器中打开 **/etc/sssd/sssd.conf** 文件。
2. **在客户端中：**将 **ipa_server** 选项添加到文件的 **[domain]** 部分，并将其设置为 IdM 服务器。这可避免 IdM 客户端自动发现其他 IdM 服务器，从而将此测试限制为一个客户端和一个服务器。

```
[domain/example.com]
ipa_server = server.example.com
...
```

3. **在客户端中：**保存并关闭 **sssd.conf** 文件。
4. **在客户端中：**重启 SSSD 服务以载入配置更改。

```
[root@client ~]# systemctl restart sssd
```

5. **在服务器和客户端中：**启用详细的 SSSD 调试日志。

```
[root@server ~]# sssctl debug-level 6
```

```
[root@client ~]# sssctl debug-level 6
```

6. **在服务器和客户端中：**对于出现身份验证问题的用户，SSSD 缓存中的对象无效，因此不会绕过 LDAP 数据库并检索 SSSD 的信息。

```
[root@server ~]# sssctl cache-expire -u idmuser
```

```
[root@client ~]# sssctl cache-expire -u idmuser
```

7. **在服务器和客户端中：**通过删除旧的 SSSD 日志来最大程度减少数据集的故障排除。

```
[root@server ~]# sssctl logs-remove
```

```
[root@server ~]# sssctl logs-remove
```

8. **在客户端中：**尝试切换到用户在尝试之前和之后收集时间戳时遇到身份验证问题的用户。这些时间戳进一步缩小了数据集的范围。

```
[root@client sssd]# date; su idmuser; date
Mon Mar 29 16:20:13 EDT 2021
su: user idmuser does not exist
Mon Mar 29 16:20:14 EDT 2021
```

9. *（可选）***在服务器和客户端中：**如果您不想继续收集详细的 SSSD 日志，请降低 debug 级别。

```
[root@server ~]# sssctl debug-level 0
```

```
[root@client ~]# sssctl debug-level 0
```

10. **在服务器和客户端中：**查看 SSSD 日志，了解失败请求的信息。
    a. 在客户端日志中查看来自客户端的请求。
    b. 在服务器日志中查看来自客户端的请求。
    c. 在服务器日志中检查请求的结果。
    d. 查看客户端收到来自服务器的请求结果的结果。
11. 如果您无法确定导致身份验证问题的原因：
    a. 收集您最近在 IdM 服务器和 IdM 客户端中生成的 SSSD 日志。根据主机名或角色标记它们。

```
[root@server ~]# sssctl logs-fetch sssd-logs-server-Mar29.tar
```

```
[root@client ~]# sssctl logs-fetch sssd-logs-client-Mar29.tar
```

    b. 创建一个红帽技术支持问题单并提供：
        i. SSSD 调试日志：
            A. 来自服务器的 **sssd-logs-server-Mar29.tar**。
            B. 来自客户端的 **sssd-logs-client-Mar29.tar**
        ii. 与日志对应的请求的控制台输出，包括时间戳和用户名：

```
[root@client sssd]# date; su idmuser; date
Mon Mar 29 16:20:13 EDT 2021
su: user idmuser does not exist
Mon Mar 29 16:20:14 EDT 2021
```

13.10. 跟踪 SSSD 后端中的客户端请求

SSSD 以异步方式处理请求，并将来自不同请求的消息添加到同一日志文件中，您可以使用唯一的请求标识符和客户端 ID 来在后端日志中跟踪客户端请求。唯一的请求标识符以 **RID#<integer>** 形式添加到调试日志中，客户端 ID 的格式为 **[CID #<integer>]**。这可让您隔离与单个请求相关的日志，您可以跨多个 SSSD 组件的日志文件从头到尾跟踪请求。

先决条件

- 您已启用了调试日志，并且已从 IdM 客户端提交了请求。
- 您必须具有 root 特权才能显示 SSSD 日志文件的内容。

步骤

1. 要查看 SSSD 日志文件，请使用 **less** 工具打开日志文件。例如，查看 **/var/log/sssd/sssd_example.com.log** ：

```
[root@server ~]# less /var/log/sssd/sssd_example.com.log
```

2. 查看 SSSD 日志，以获取有关客户端请求的信息。

```
(2021-07-26 18:26:37): [be[testidm.com]] [dp_req_destructor] (0x0400): [RID#3] Number of
active DP request: 0
(2021-07-26 18:26:37): [be[testidm.com]] [dp_req_reply_std] (0x1000): [RID#3] DP Request
AccountDomain #3: Returning [Internal Error]: 3,1432158301,GetAccountDomain() not
supported
(2021-07-26 18:26:37): [be[testidm.com]] [dp_attach_req] (0x0400): [RID#4] DP Request
Account #4: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-07-26 18:26:37): [be[testidm.com]] [dp_attach_req] (0x0400): [RID#4] Number of
active DP request: 1
```

SSSD 日志文件中的这个示例输出显示了两个不同的请求的唯一标识符 **RID#3** 和 **RID#4**。

但是，对 SSSD 客户端接口的单一客户端请求通常会在后端触发多个请求，因此客户端请求和后端中的请求之间不是 1 到 1 的对应关系。虽然后端中的多个请求有不同的 RID 号，但每个初始后端请求都包括唯一的客户端 ID，以便管理员可以跟踪单个客户端请求的多个 RID 号。

以下示例显示了一个客户端请求 **[sssd.nss CID #1]** 和多个在后端生成的请求，**[RID#5]** 到 **[RID#13]**:

```
(2021-10-29 13:24:16): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#5] DP Request [Account #5]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:16): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#6] DP Request [AccountDomain
#6]: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:16): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#7] DP Request [Account #7]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#8] DP Request [Initgroups #8]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#9] DP Request [Account #9]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#10] DP Request [Account #10]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#11] DP Request [Account #11]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#12] DP Request [Account #12]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#13] DP Request [Account #13]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
```

13.11. 使用日志分析器工具跟踪客户端请求

系统安全服务守护进程(SSSD)包含一个日志解析工具，其可用于跟踪跨多个 SSSD 组件日志文件的从头到尾的请求。

13.11.1. 日志分析器工具如何工作

通过使用日志解析工具，您可以跟踪跨多个 SSSD 组件日志文件的从头到尾的请求。您可以使用 **sssctl analyze** 命令运行分析器工具。

日志分析器工具可帮助您对 SSSD 中的 NSS 和 PAM 问题进行故障排除，并更容易查看 SSSD 调试日志。您只能提取和打印跨 SSSD 进程的与某些客户端请求相关的 SSSD 日志。

SSSD 会分别跟踪用户和组身份信息(**id**,**getent**)和用户身份验证（**su**、**ssh**）信息。NSS 响应器中的客户端 ID(CID)与 PAM 响应者中的 CID 无关，在分析 NSS 和 PAM 请求时会看到重叠数字。使用 **sssctl analyze** 命令和 **--pam** 选项来查看 PAM 请求。

注意

从 SSSD 内存缓存返回的请求不会被记录，且不能被日志分析器工具跟踪。

其他资源

- **sudo sssctl analyze request --help**
- **sudo sssctl analyze --help**
- **sssd.conf** 手册页
- **sssctl** 手册页

13.11.2. 运行日志分析器工具

按照以下流程，使用日志分析器工具跟踪 SSSD 中的客户端请求。

先决条件

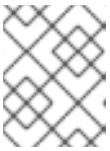
- 您必须在 [$responder] 部分中将 **debug_level** 设为至少为 7，设置 **/etc/sssd/sssd.conf** 文件的 [domain/$domain] 部分以必启用日志解析功能。
- 分析的日志必须来自使用 **libtevent** 链 ID 支持构建的 SSSD 的兼容版本，它是 RHEL 8.5 及之后版本中的 SSSD。

步骤

1. 在 **list** 模式下运行日志分析器工具以确定您在跟踪的请求的客户端 ID，添加 **-v** 选项以显示详细输出：

```
# sssctl analyze request list -v
```

此时会显示最近发出的对 SSSD 的客户端请求的详细列表。



注意

如果分析 PAM 请求，请使用 **--pam** 选项运行 **sssctl analyze request list** 命令。

2. 使用 **show [unique client ID]** 选项运行日志分析器工具，以显示与指定客户端 ID 号相关的日志：

```
# sssctl analyze request show 20
```

3. 如果需要，您可以针对日志文件运行日志分析器工具，例如：

```
# sssctl analyze request --logdir=/tmp/var/log/sssd
```

其他资源

其他资源

- **sssctl analyze request list --help**
- **sssctl analyze request show --help**
- **sssctl** 手册页。

13.12. 其他资源

- 常规 SSSD 调试流程

第 14 章 为单点登录配置应用程序

单点登录(SSO)是一种身份验证方案，允许您通过一个登录过程登录多个系统。您可以将浏览器和电子邮件客户端配置为使用 Kerberos 票据、SSL 认证或令牌来对用户进行身份验证。

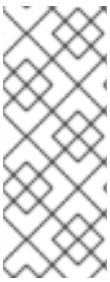
不同应用程序的配置可能有所不同。本章演示了如何为 Mozilla Thunderbird 电子邮件客户端和 Mozilla Firefox Web 浏览器配置 SSO 身份验证模式作为示例。

14.1. 先决条件

- 已安装以下应用程序：
  - Mozilla Firefox *版本 88*
  - Mozilla Thunderbird *版本 78*

14.2. 将 FIREFOX 配置为使用 KERBEROS 进行单点登录

您可以将 Firefox 配置为使用 Kerberos 作为对 Intranet 站点和其他受保护网站的单点登录(SSO)。为此，您必须首先将 Firefox 配置为将 Kerberos 凭据发送到适当的密钥分发中心(KDC)。



注意

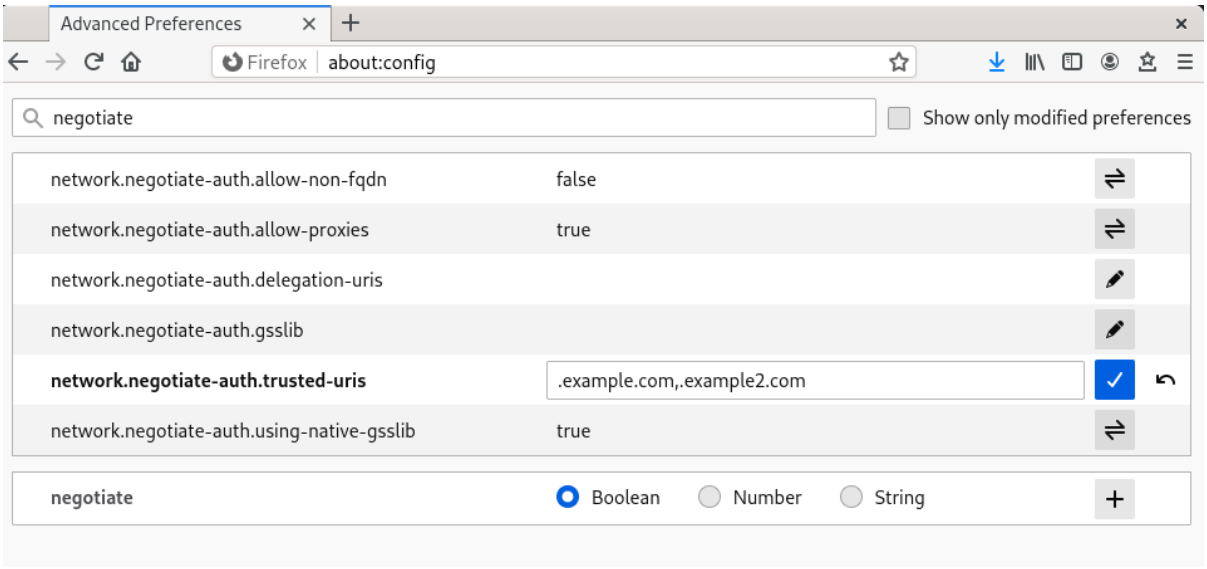
即使 Firefox 配置为传递 Kerberos 凭据，它仍需要有效的 Kerberos 票据才能使用。要生成 Kerberos 票据，请使用 **kinit** 命令并在 KDC 上提供用户密码。

```
[jsmith@host ~] $ kinit
Password for jsmith@EXAMPLE.COM:
```

步骤

1. 在 Firefox 的地址栏中，键入 **about:config** 以显示当前配置选项的列表。
2. 在 **Filter** 字段中，键入 **negotiate** 来限制选项列表。
3. 双击 **network.negotiate-auth.trusted-uris** 项。
4. 输入要进行身份验证的域的名称，包括前面的句点(.)。如果要添加多个域，使用以逗号分隔的列表形式输入它们。

图 14.1. 手动 Firefox 配置



其他资源

- 有关在身份管理中将 Firefox 配置为使用 Kerberos 的详情，请查看 Linux 域身份、身份验证和策略指南中的相应部分。

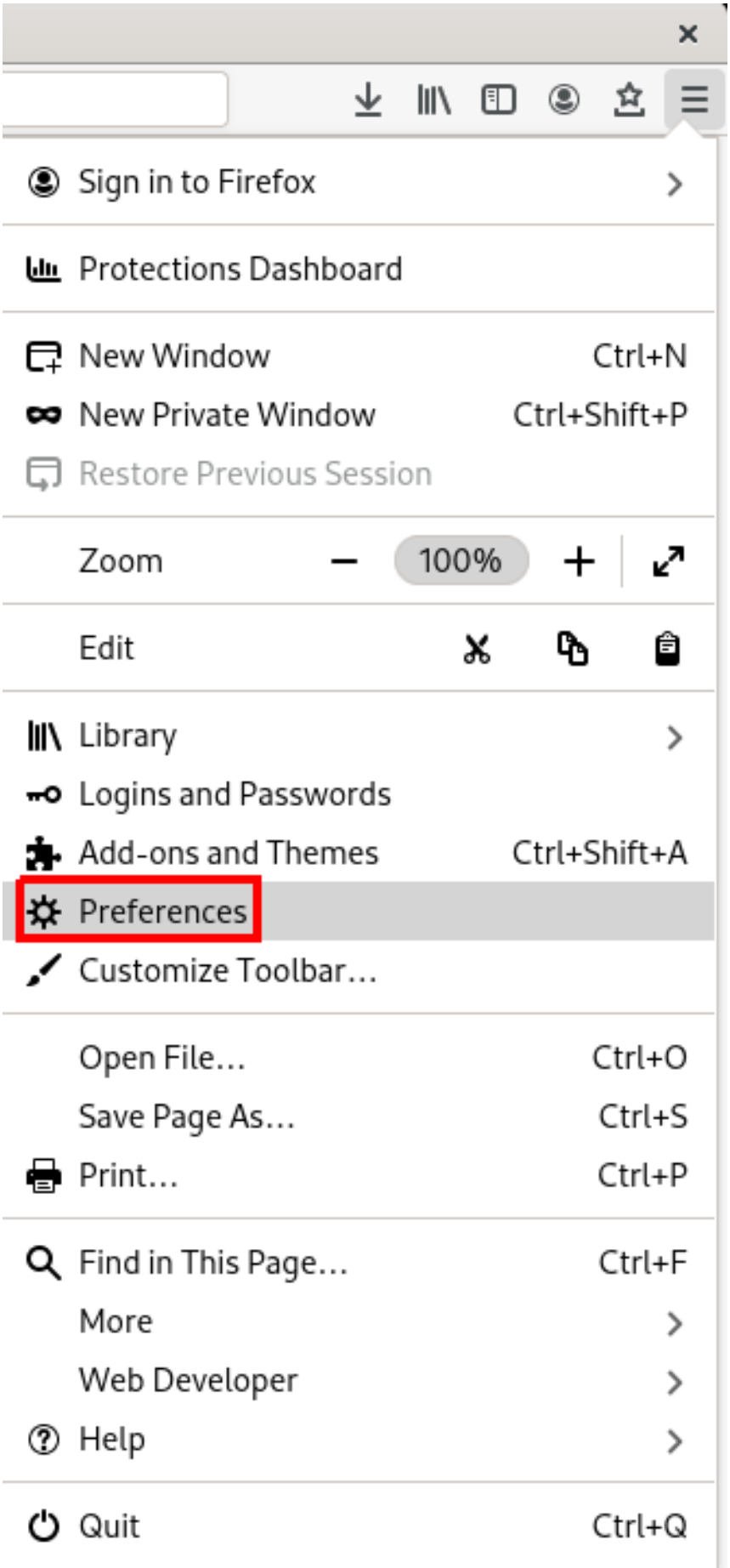
14.3. 在 FIREFOX 中查看证书

下例演示了如何在 Mozilla Firefox 中查看证书。

要在 Firefox 中查看证书，您需要打开**证书管理器（Certificate Manager）**。

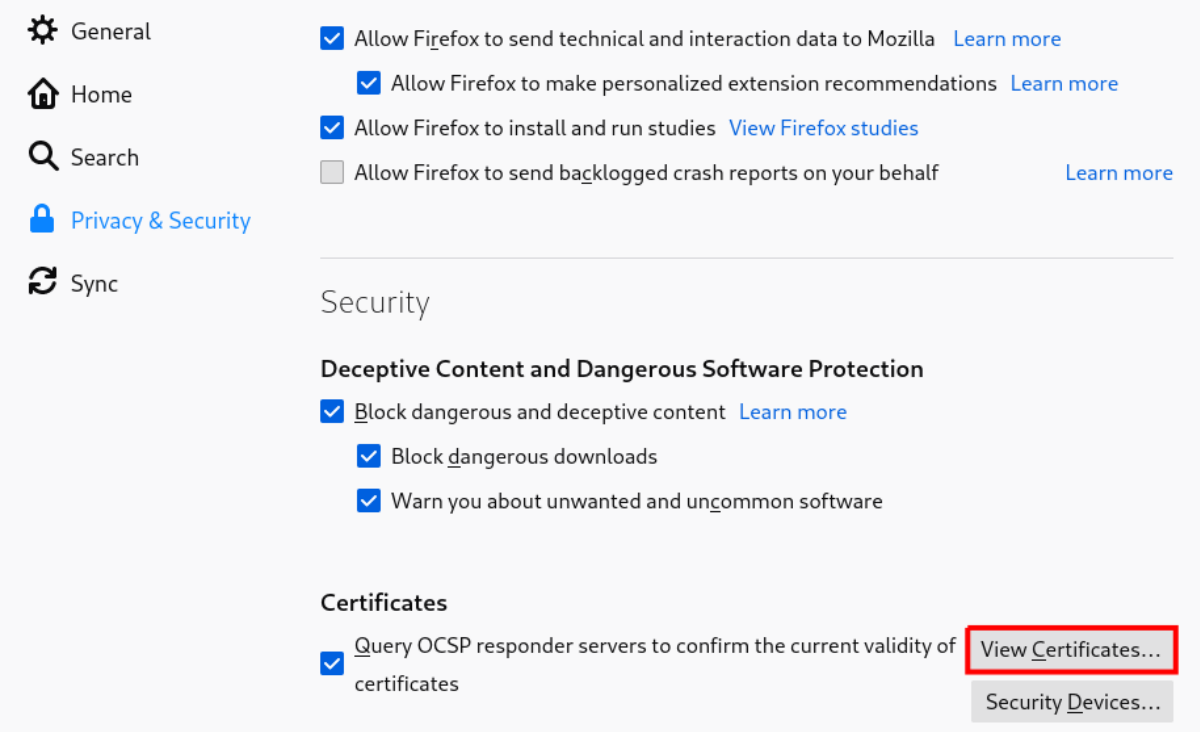
步骤

1. 在 Mozilla Firefox 中，打开 Firefox 菜单并选择 **Preferences**。



2. 在左侧面板中，选择 **Privacy & Security** 部分。

3. 向下滚动到 证书 部分。
4. 点 **View Certificates** 打开 **Certificate Manager**。



14.4. 在 FIREFOX 中导入 CA 证书

下例演示了如何在 Mozilla Firefox 中导入证书。

先决条件

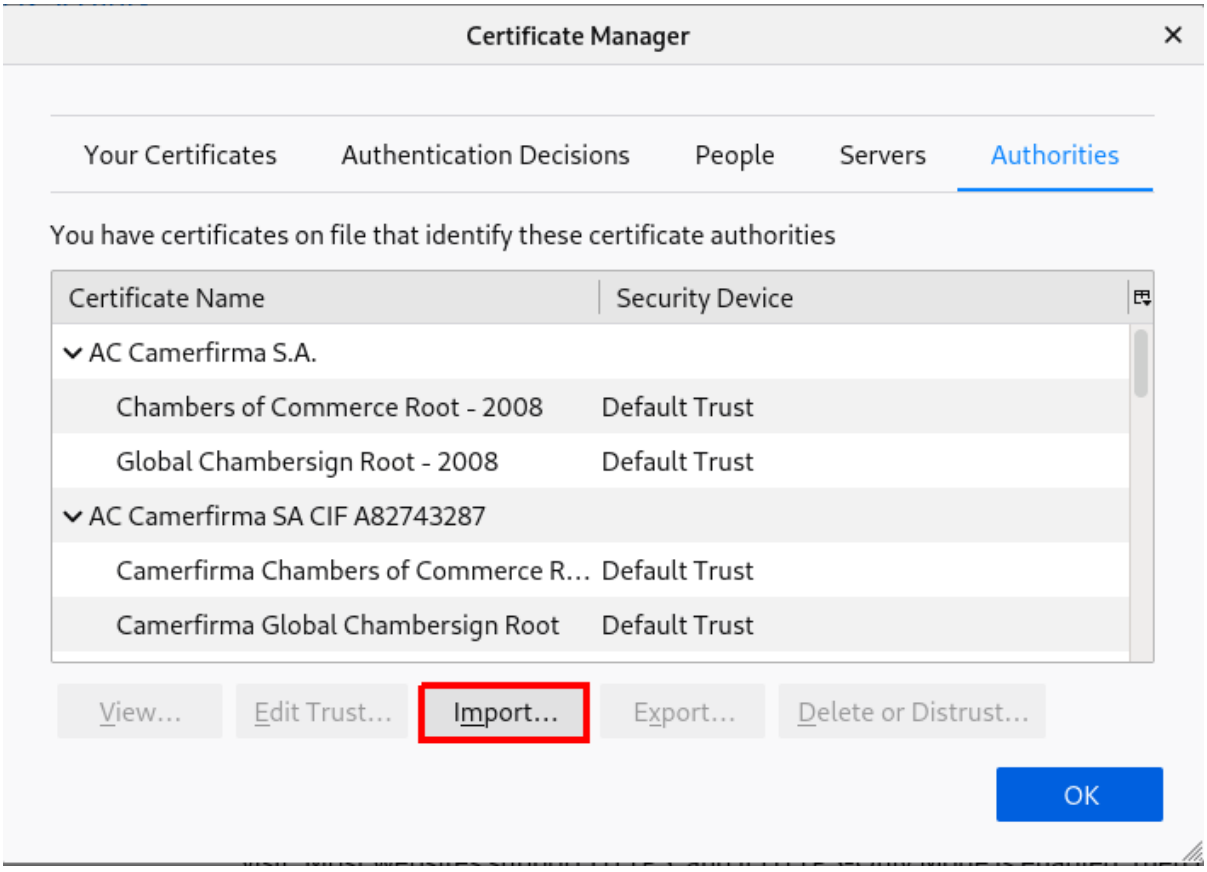
- 您的设备中有 CA 证书。

导入 CA 证书：

步骤

1. 打开 证书管理器.
2. 选择 **Authorities** 标签页，点 **Import**。

图 14.2. 在 Firefox 中导入 CA 证书



3. 从您的设备中选择下载的 CA 证书。

14.5. 在 FIREFOX 中编辑证书信任设置

下例演示了如何在 Mozilla Firefox 中编辑证书设置。

先决条件

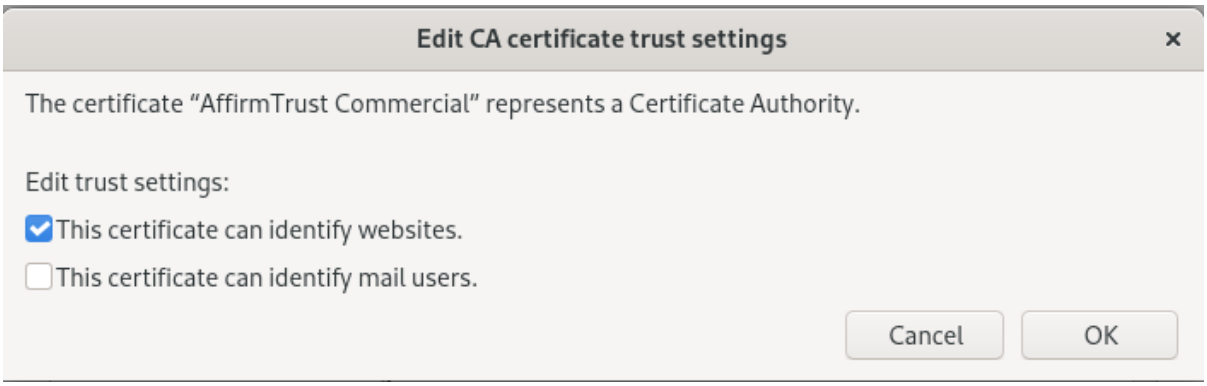
1. 您已成功导入证书。

设置证书信任设置：

步骤

1. 打开证书管理器。
2. 在 **Authorities** 选项卡下，选择适当的证书，再单击 **Edit Trust**。
3. 编辑证书信任设置。

图 14.3. 在 Firefox 中编辑证书信任设置



14.6. 在 FIREFOX 中导入用于身份验证的个人证书

下例演示了如何在 Mozilla Firefox 中导入用于身份验证的个人证书。

先决条件

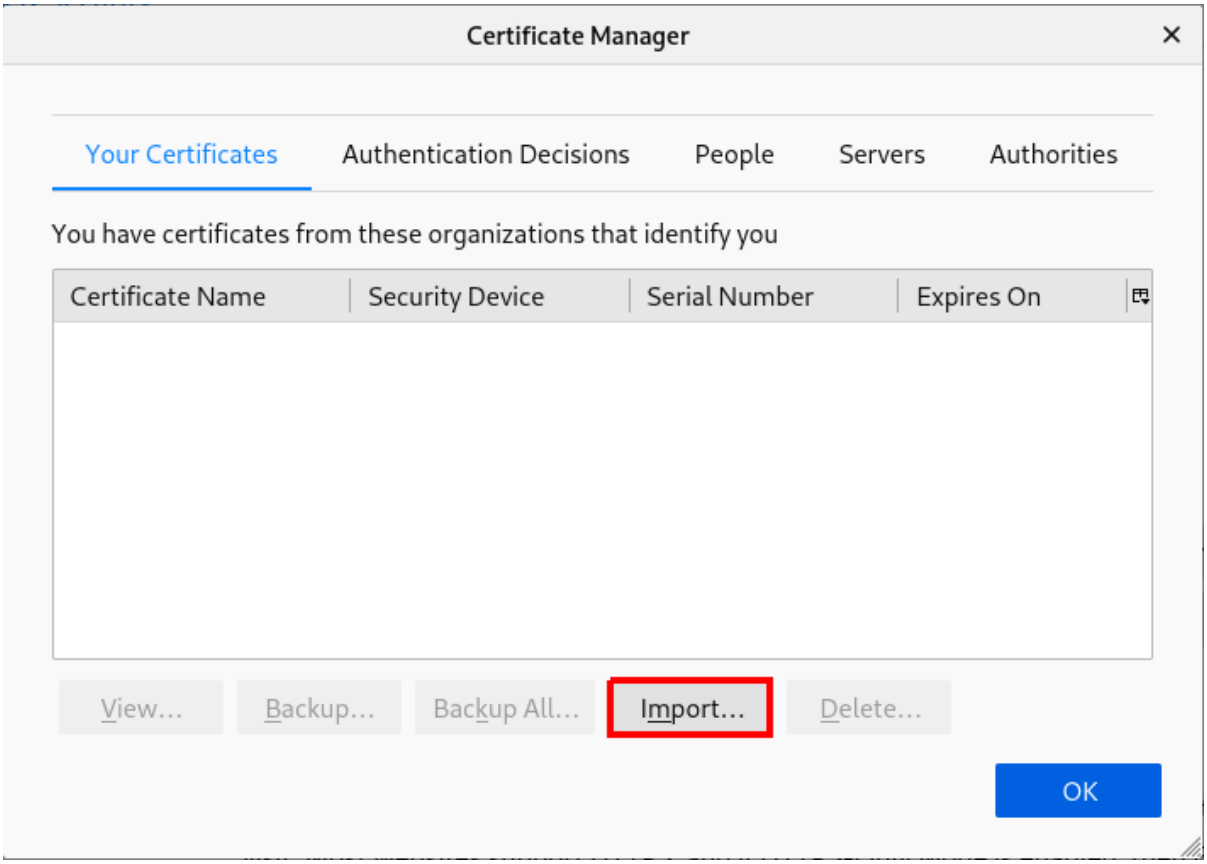
1. 您设备上存储了一个个人证书。

使用个人证书进行身份验证：

步骤

1. 打开**证书管理器**。
2. 选择 **您的证书** 选项卡，然后单击 **导入**。

图 14.4. 在 Firefox 中导入用于身份验证的个人证书



3. 从您的计算机选择适当的证书。

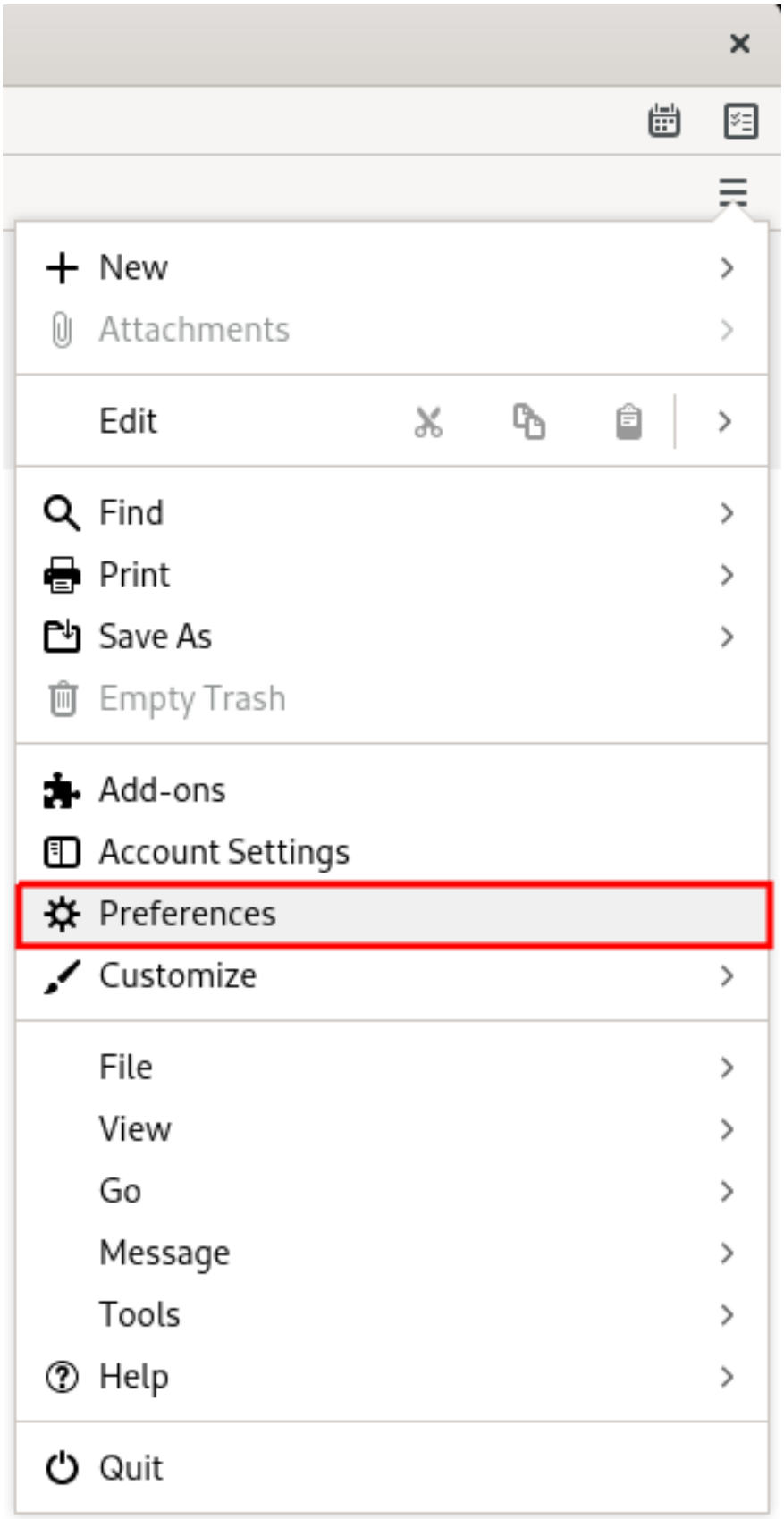
14.7. 在 THUNDERBIRD 中查看证书

以下示例演示了如何在 Mozilla Thunderbird 电子邮件客户端中查看证书。

步骤

1. 在 Mozilla Thunderbird 中，打开主菜单并选择 **Preferences**。

图 14.5. 从菜单中选择首选项

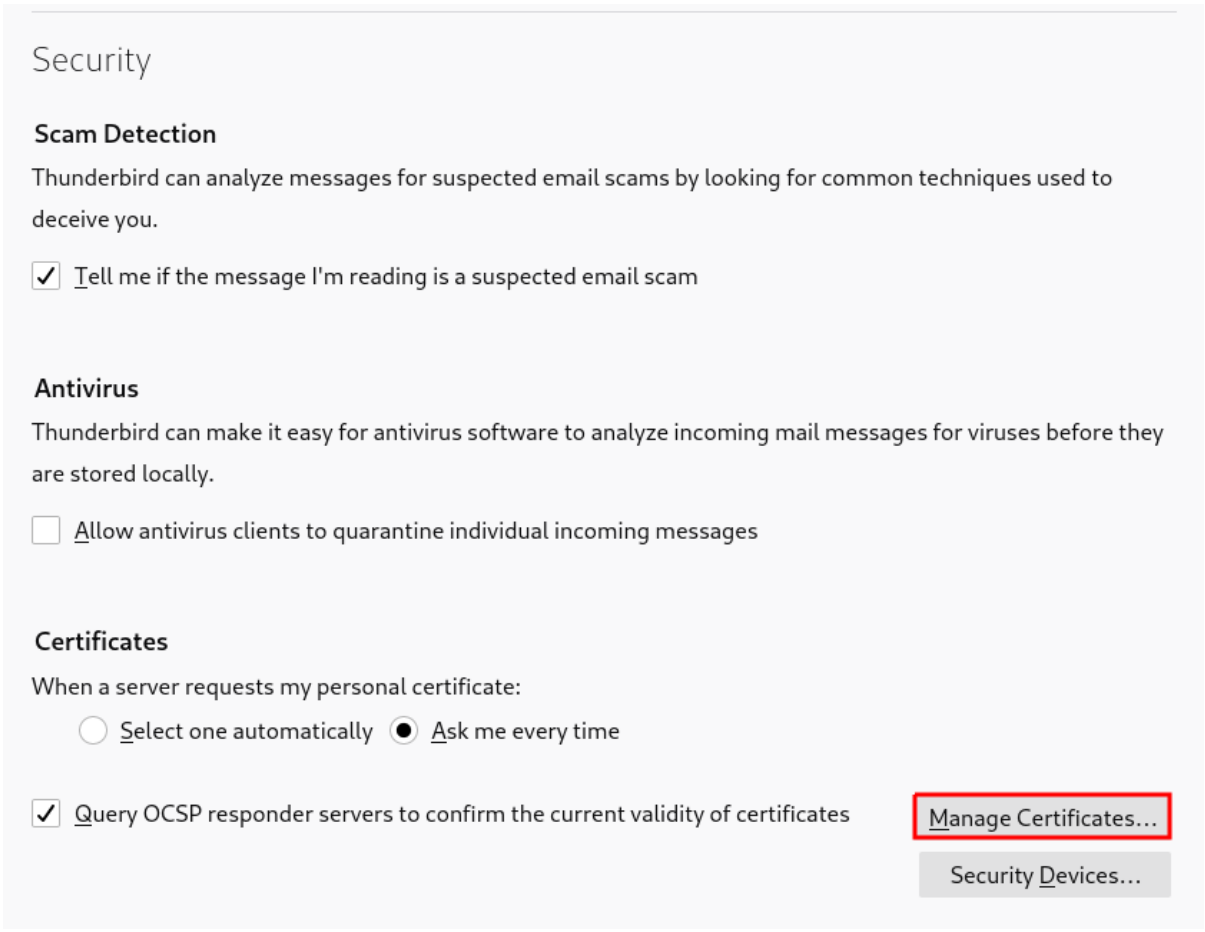


2. 在左侧面板中，选择 **Privacy & Security** 部分。

图 14.6. 选择 security 部分

3. 向下滚动到 证书 部分。
4. 点 **Manage Certificates** 打开 **Certificate Manager**。

图 14.7. 打开证书管理器



14.8. 在 THUNDERBIRD 中导入证书

以下示例演示了如何在 Mozilla Thunderbird 电子邮件客户端中导入证书。

先决条件

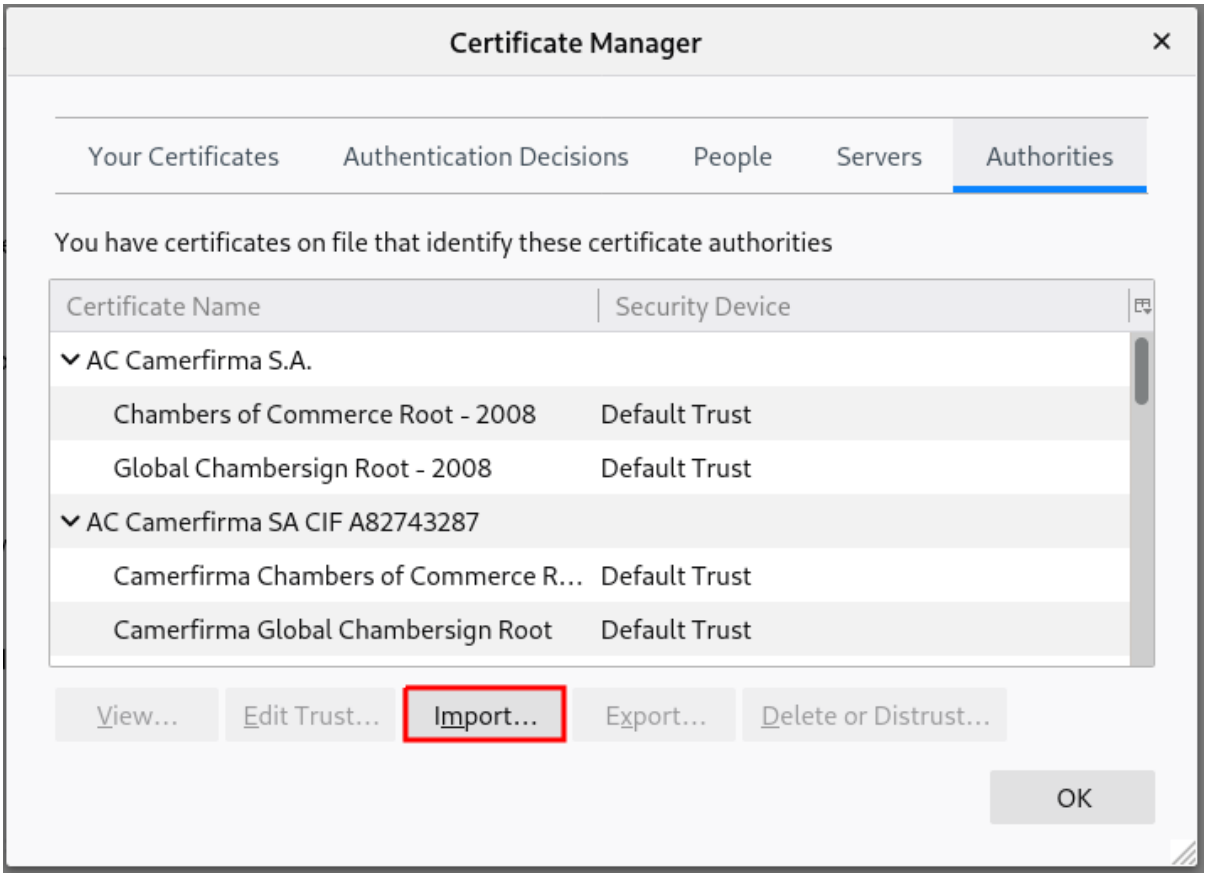
- 您的设备上存储了一个 CA 证书。

导入 CA 证书：

步骤

1. 打开证书管理器。
2. 选择 **Authorities** 标签页，点 **Import**。

   图 14.8. 在 Thunderbird 中导入 CA 证书



3. 选择下载的 CA 证书。

14.9. 编辑 THUNDERBIRD 中的证书信任设置

以下示例演示了如何在 Mozilla Thunderbird 电子邮件客户端中编辑证书设置。

先决条件

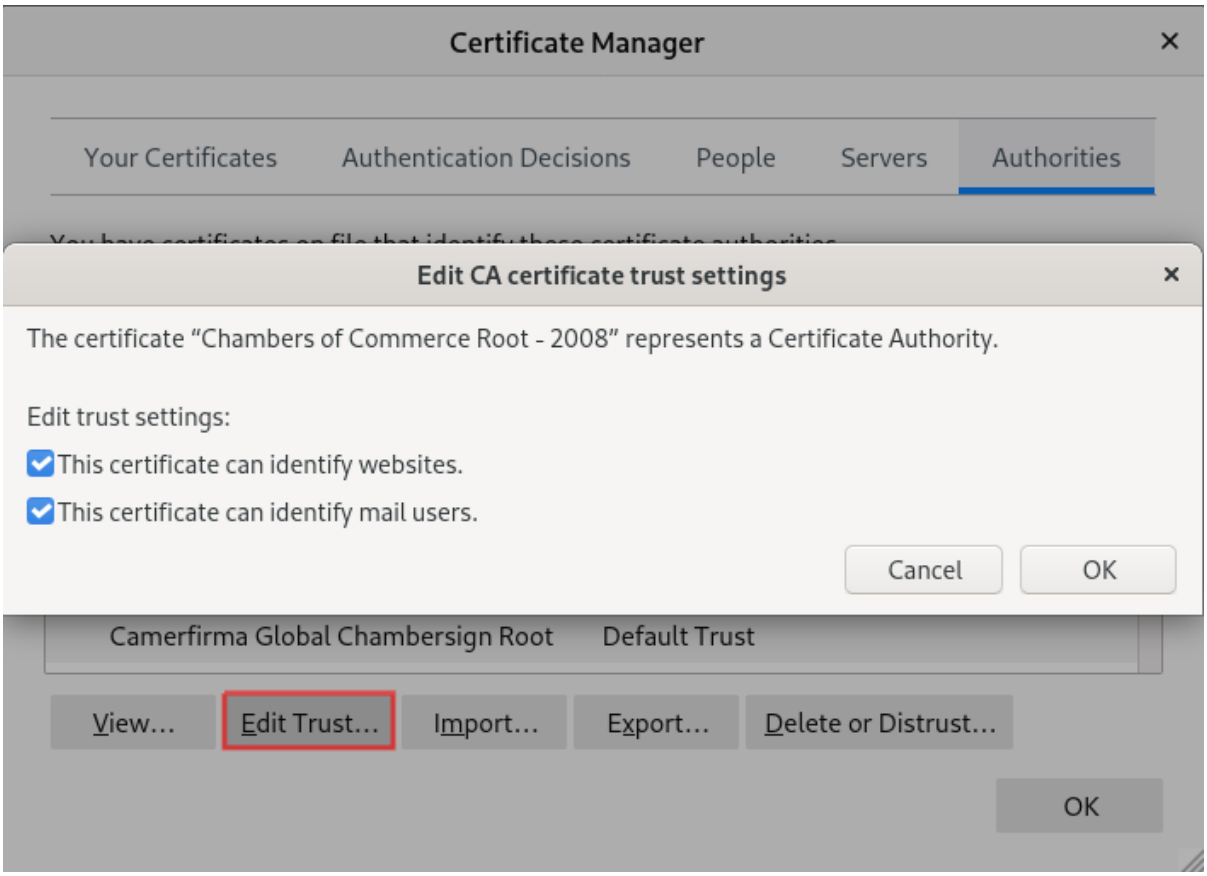
- 您已成功导入证书。

设置证书信任关系：

步骤

1. 打开证书管理器。
2. 在 **Authorities** 选项卡下，选择适当的证书，再单击 **Edit Trust**。
3. 编辑证书信任设置。

图 14.9. 编辑 Thunderbird 中的证书信任设置



14.10. 在 THUNDERBIRD 中导入个人证书

以下示例演示了如何在 Mozilla Thunderbird 电子邮件客户端中导入用于个人身份验证的证书。

先决条件

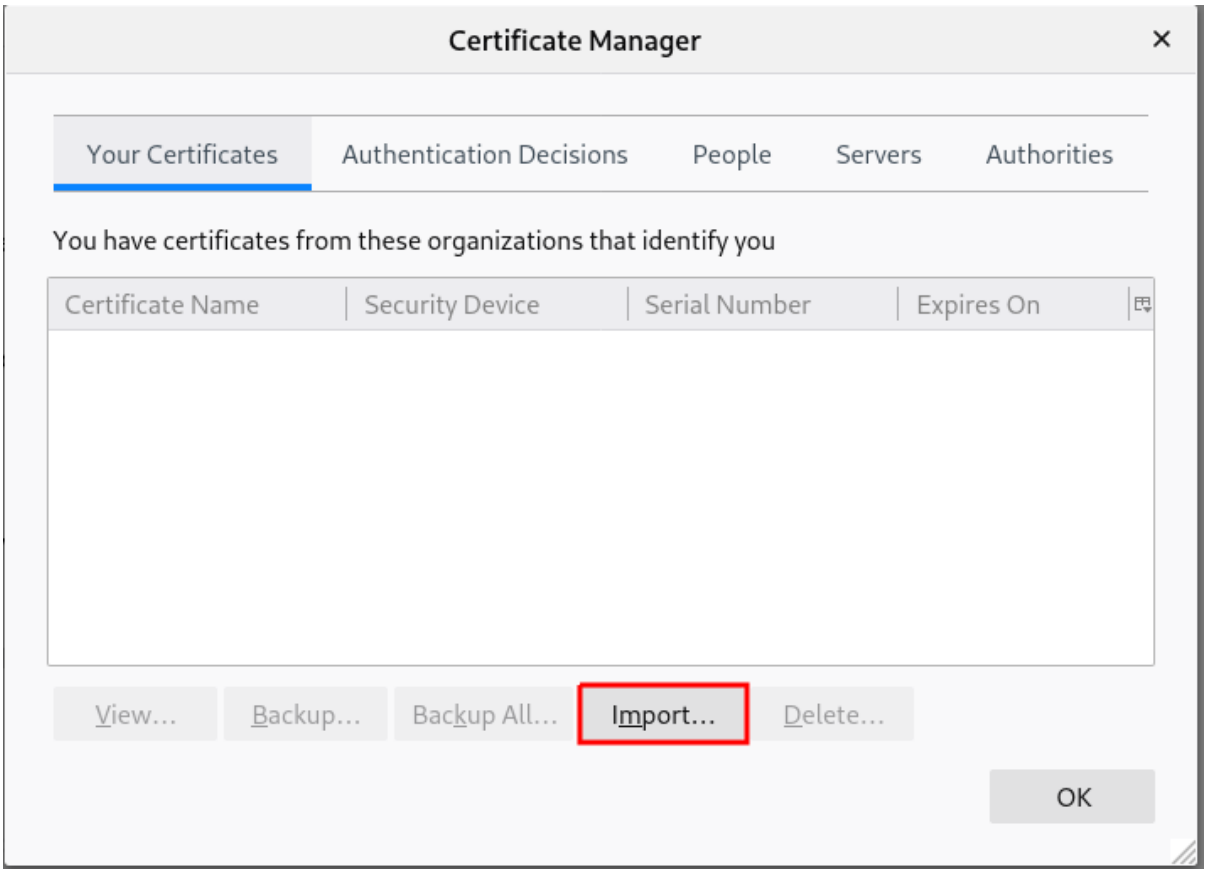
1. 您设备上存储了一个个人证书。

使用个人证书进行身份验证：

步骤

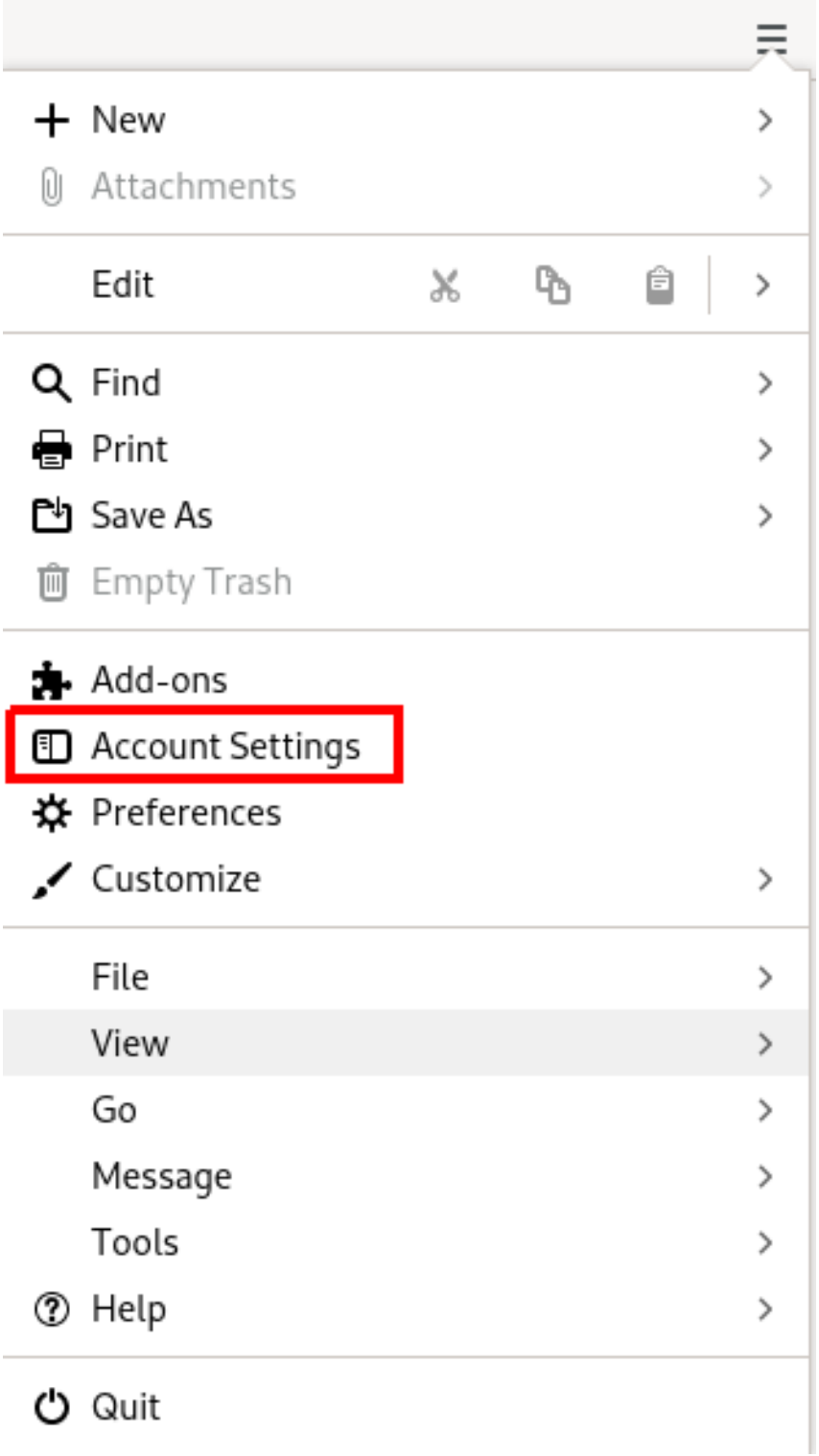
1. 打开**证书管理器**。
2. 在您的 **证书选项卡** 下，单击 **导入**。

图 14.10. 在 Thunderbird 中导入用于身份验证的个人证书



3. 从您的计算机中选择所需的证书。
4. 关闭 **证书管理器**。
5. 打开主菜单并选择**帐户设置**。

图 14.11. 从菜单中选择帐户设置



6. 在您的帐户电子邮件地址的左侧面板中选择 **End-To-End Encryption**。选择端到端加密部分。

7. 在 **S/MIME** 部分下，单击第一个选择按钮，以选择要用于签名邮件的个人证书。

8. 在 **S/MIME** 部分下，单击第二个选择按钮，以选择用于加密和解密邮件的个人证书。
选择用于签名和加密/解密的证书.

注意

如果您忘记导入有效证书，则可以使用 **Manage S/MIME** 直接打开证书管理器。