Red Hat Enterprise Linux 8

RHEL での認証と認可の設定

SSSD、authselect、および sssctl を使用した認証および認可の設定

Last Updated: 2025-08-05

Red Hat Enterprise Linux 8 RHEL での認証と認可の設定

SSSD、authselect、および sssctl を使用した認証および認可の設定

Abstract

Red Hat Enterprise Linux (RHEL) を設定して、Red Hat Identity Management (IdM)、Active Directory (AD)、LDAP ディレクトリーなどのサービスに対してユーザーを認証および認可できます。このため、RHEL は System Security Services Daemon (SSSD) を使用してこれらのサービスと通信します。authselect および sssctl などのユーティリティーは、SSSD、Pluggable Authentication Modules (PAM)、および Name Service Switch (NSS) の設定をサポートします。

Table of Contents

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

Jira からのフィードバック送信 (アカウントが必要)

1. Jira の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
5. ダイアログの下部にある **Create** をクリックします。

第1章 システム認証の概要

セキュアなネットワーク環境の基礎の 1 つは、システムにアクセスできるユーザーを、許可されたユーザーだけに制限することです。認証では、アクセスを許可する前にユーザーのアイデンティティーを検証します。

どの Red Hat Enterprise Linux システムでも、ユーザーアイデンティティーを作成および管理するためのさまざまなサービスが利用できます。これには、ローカルシステムファイル、Kerberos や Samba などの大規模なアイデンティティードメインに接続するサービス、またはそれらのドメインを作成するツールが含まれます。

1.1. RHEL の認証方法

認証とは、アイデンティティーの確認を行うプロセスです。ネットワークのやり取りにおいて、認証とは、一方の当事者がもう一方の当事者のアイデンティティーを確認できるようにすることです。ネットワーク上で認証を使用する方法は、単純なパスワード、証明書、パスワードレス方式、ワンタイムパスワード (OTP) トークン、生体認証スキャンなど、多数あります。

認可では、認証された当事者がアクセスできる内容や実行できる内容を定義します。

認証では、当事者が自身のアイデンティティーを検証するために何らかの認証情報を提示する必要があります。必要な認証情報の種類は、使用される認証メカニズムによって定義されます。

システム上のローカルユーザー認証の種類

パスワードベースの認証

ほとんどのソフトウェアで、ユーザーは認識されたユーザー名とパスワードを提供することで認証できます。これは簡易認証とも呼ばれます。

証明書ベースの認証

証明書に基づくクライアント認証は、Secure Sockets Layer (SSL) プロトコルの一部です。クライアントは無作為に生成されたデータの一部に署名し、ネットワーク全体で証明書および署名されたデータの両方を送信します。サーバーは署名を検証し、証明書の有効性を確認します。

Kerberos 認証

Kerberos は、Ticket-Granting Ticket (TGT) と呼ばれる、有効期間が短い認証情報のシステムを確立します。ユーザーは、ユーザーを特定し、ユーザーにチケットを発行できることをシステムに示す認証情報、つまりユーザー名およびパスワードを提示します。TGT は、Web サイトや電子メールなどの他のサービスへのアクセスチケットを要求するために繰り返し使用できます。Kerberos を使用した認証では、ユーザーはこのように 1 回の認証プロセスのみを実行することになります。

スマートカードベースの認証

これは証明書ベースの認証の一種です。スマートカード (またはトークン) にはユーザー証明書が保存されています。ユーザーがトークンをシステムに挿入すると、システムが証明書を読み取ってアクセスを許可します。スマートカードを使用したシングルサインオンには、以下の 3 つの手順があります。

1. ユーザーがスマートカードをカードリーダーに挿入します。Red Hat Enterprise Linux のプラグ可能な認証モジュール (PAM) は、挿入されたスマートカードを検出します。
2. システムは、証明書をユーザーエントリーにマップし、スマートカードに表示された証明書を、証明書ベースの認証で説明されているように秘密鍵で暗号化して、ユーザーエントリーに保存されている証明書と比較します。
3. 証明書がキー配布センター (KDC) に対する検証に成功すると、ユーザーはログインを許可されます。

スマートカードベースの認証は、追加の識別メカニズムとして証明書を追加し、物理的なアクセス要件を追加することにより、Kerberos によって確立された単純な認証層に基づいています。

ワンタイムパスワード認証

ワンタイムパスワードにより、認証セキュリティーに関する手順が追加されます。この認証では、ユーザーのパスワードと自動的に生成されたワンタイムパスワードを組み合わせて使用します。

外部アイデンティティープロバイダー

OAuth 2 デバイス認可フローをサポートする外部アイデンティティープロバイダー (IdP) にユーザーを関連付けることができます。このユーザーが RHEL 9.1 以降で利用可能な SSSD バージョンで認証すると、ユーザーは、外部 IdP で認証と認可を実行した後、Kerberos チケットを使用した RHEL Identity Management (IdM) シングルサインオン機能を受け取ります。

関連情報

- スマートカード認証の管理
- Identity Management におけるワンタイムパスワード (OTP) 認証
- 外部 ID プロバイダーを使用した IdM に対する認証

1.2. RHEL におけるシングルサインオンの概要

一元的なアイデンティティーストアがない場合、各アプリケーションが独自のユーザー認証情報を維持管理します。その場合、ユーザーはアクセスするすべてのサービスまたはアプリケーションに対してパスワードを入力する必要があります。

管理者がシングルサインオン (SSO) を設定すると、1 つのパスワードストアが作成されます。ユーザーは、1 つのパスワードを使用して一度ログインするだけで、すべてのネットワークリソースにアクセスできるようになります。

Red Hat Enterprise Linux は、ワークステーションへのログイン、スクリーンセーバーのロック解除、Mozilla Firefox を使用した保護された Web ページへのアクセスなど、いくつかのリソースに対する SSO をサポートしています。特権アクセス管理 (PAM)、Name Service Switch (NSS)、Kerberos など、その他のシステムサービスが利用可能な場合は、これらのアイデンティティーソースを使用するように他のシステムアプリケーションを設定できます。

SSO は、ユーザーにとって便利であると同時に、サーバーとネットワークのセキュリティーをさらに強化するものでもあります。SSO はセキュアで効果的な認証に左右されます。RHEL は、SSO を可能にする認証メカニズムを 2 つ備えています。

- Kerberos レルムと Active Directory ドメインを使用した Kerberos ベースの認証
- スマートカードベースの認証

どちらの方法でも、(Kerberos レルムまたは公開鍵基盤の認証局によって) 一元的なアイデンティティーストアが作成されます。ローカルシステムの各サービスは、複数のローカルストアを維持する代わりに、それらのアイデンティティードメインを使用します。

1.3. ローカルユーザー認証に利用できるサービス

すべての Red Hat Enterprise Linux システムには、ローカルシステム上のローカルユーザーの認証を設定するサービスが用意されています。これには以下が含まれます。

認証設定

- 認証設定ツール **authselect** は、システムに対して、さまざまなアイデンティティーバックエンドと認証手段 (パスワード、指紋、スマートカードなど) を設定します。

アイデンティティーバックエンド設定

- Security System Services Daemon (SSSD) は、主に Microsoft Active Directory や IdM などの LDAP ベースのディレクトリーを中心に、複数のアイデンティティープロバイダーを設定します。これらのアイデンティティープロバイダーは、ローカルシステムとアプリケーションの両方で認証に使用できます。SSSD はパスワードとチケットをキャッシュし、認証情報を再利用することでオフライン認証とシングルサインオンを可能にします。

- **realmd** サービスは、認証バックエンド (IdM の SSSD) の設定を可能にするコマンドラインユーティリティーです。**realmd** サービスは、DNS レコードに基づいて利用可能な IdM ドメインを検出し、SSSD を設定してから、システムをアカウントとしてドメインに参加させます。

- NSS (Name Service Switch) は、ユーザー、グループ、またはホストの情報を返す低レベルのシステムコールのメカニズムです。NSS は、必要な情報を取得するのに使用するモジュールであるソースを決定します。たとえば、ユーザー情報は、**/etc/passwd** ファイルなどの従来の UNIX ファイルや LDAP ベースのディレクトリーに保存されていることがあります。また、ホストアドレスは、**/etc/hosts** ファイルや DNS レコードなどのファイルから読み取られることがあります。NSS は、このような情報が保存されている場所を特定します。

認証メカニズム

- プラグ可能な認証モジュール (PAM) は、認証ポリシーを設定するシステムを提供します。認証に PAM を使用するアプリケーションは、認証のさまざまな側面を制御する異なるモジュールを読み込みます。アプリケーションが使用する PAM モジュールは、アプリケーションの設定方法に基づいています。利用可能な PAM モジュールには、Kerberos、Winbind、SSSD、ローカルの UNIX ファイルベースの認証があります。

その他のサービスやアプリケーションも利用できますが、これらは一般的な設定です。

第2章 AUTHSELECT を使用したユーザー認証の設定

authselect は、特定のプロファイルを選択して、システム ID および認証ソースを設定できるようにするユーティリティーです。プロファイルは、Pluggable Authentication Modules (PAM) およびネットワークセキュリティーサービス (NSS) の設定を記述したファイルのセットです。デフォルトのプロファイルセットを使用することも、カスタムプロファイルを作成することもできます。

2.1. AUTHSELECT の使用方法

authselect ユーティリティーを使用して、Red Hat Enterprise Linux 8 ホストでユーザー認証を設定できます。

既製のプロファイルのいずれかを選択して、ID 情報および認証ソースおよびプロバイダーを設定できます。

- デフォルトの **sssd** プロファイルでは、LDAP 認証を使用するシステムの System Security Services Daemon (SSSD) が有効になります。
- **winbind** プロファイルは、Microsoft Active Directory と直接統合したシステムの Winbind ユーティリティーを有効にします。
- **nis** プロファイルにより、従来のネットワーク情報サービス (NIS) システムとの互換性が確保されます。
- **minimal** プロファイルは、システムファイルから直接ローカルユーザーおよびグループのみを提供します。これにより、管理者は不要になったネットワーク認証サービスを削除できます。

authselect プロファイルを特定のホストに対して選択すると、そのプロファイルは、そのホストにログインしているすべてのユーザーに適用されます。

Red Hat は、たとえば、ドメイン内でサービスを使用するために、データベースの LDAP、winbind、または NIS を使用してユーザーを認証している場合など、半集中型の ID 管理環境での **authselect** の使用を推奨しています。

警告

次の場合は、**authselect** を使用する必要はありません。

- ホストは Red Hat Enterprise Linux Identity Management (IdM) の一部です。ホストを IdM ドメインに参加させると、**ipa-client-install** コマンドは、ホストで SSSD 認証を自動的に設定します。
- ホストは SSSD 経由で Active Directory の一部です。**realm join** コマンドを呼び出して、ホストを Active Directory ドメインに参加させると、ホストで SSSD 認証が自動的に設定されます。

Red Hat は、**ipa-client-install** または **realm join** により設定された **authselect** プロファイルを変更しないことを推奨します。それらを変更する必要がある場合は、変更を加える前に現在の設定を表示して、必要に応じて元に戻すことができるようにします。

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

2.1.1. authselect によって変更されるファイルとディレクトリー

authselect によって変更される設定ファイルは限られているため、認証設定の管理とトラブルシューティングが容易です。

/etc/nsswitch.conf	GNU C ライブラリーおよびその他アプリケーションはこの NSS (Name Service Switch) を使用して、さまざまなカテゴリーの名前サービス情報を、どのソースから、どの順番で取得するかを決定します。情報の各カテゴリーは、データベース名で識別されます。

/etc/pam.d/* ファイル	Linux-PAM (Pluggable Authentication Modules) は、システムのアプリケーション (サービス) の認証タスクを処理するモジュールのシステムです。認証の性質は動的に設定できます。システム管理者は、個々のサービス提供アプリケーションがユーザーを認証する方法を選択できます。 **/etc/pam.d/** ディレクトリー内の設定ファイルには、サービスに必要な認証タスクを実行する PAM のリストと、個々の PAM が失敗した場合の PAM-API の適切な動作がリスト表示されます。 たとえば、これらのファイルには以下の情報が含まれています。 ● ユーザーパスワードのロックアウトの条件 ● スマートカードによる認証機能 ● フィンガープリントリーダーによる認証機能
/etc/dconf/db/distro.d/* ファイル	このディレクトリーは、**dconf** ユーティリティーの設定プロファイルを保持し、GNOME デスクトップグラフィカルユーザーインターフェイス (GUI) の設定を管理できます。

2.1.2. /etc/nsswitch.conf のデータプロバイダー

デフォルトの **sssd** プロファイルは、**/etc/nsswitch.conf** に **sss** エントリーを作成することで、SSSD を情報ソースとして確立します。

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

これは、これらの項目のいずれかに関する情報が要求されると、システムが最初に SSSD を調べることを意味します。

- ユーザー情報の **passwd**
- ユーザー **グループ** 情報のグループ
- NIS **netgroup** 情報の **netgroup**
- NFS 自動マウント情報の **automount**
- サービスに関する情報に関する **services**

sssd キャッシュ、および認証を提供するサーバーで、要求された情報が見つからない、または **sssd** を実行していないと、システムはローカルファイル (**/etc/***) を調べます。

たとえば、ユーザー ID に関する情報が要求されると、そのユーザー ID は、最初に **sssd** キャッシュで検索されます。そこで見つからない場合は、**/etc/passwd** ファイルが参照されます。同様に、ユーザーのグループ所属が要求されると、最初に **sssd** キャッシュで検索され、そこに見つからない場合に限

り、**/etc/group** ファイルが参照されます。

実際には、ローカルの **files** データベースは参照されません。最も重要な例外は、**root** ユーザーの場合です。これは、**sssd** で処理されることはありませんが、**files** で処理されます。

2.2. AUTHSELECT プロファイルの選択

システム管理者は、特定のホストの **authselect** ユーティリティーにプロファイルを選択できます。そのプロファイルはそのホストにログインしているすべてのユーザーに適用されます。

前提条件

- **authselect** コマンドを実行するには **root** 認証情報が必要です。

手順

- 認証プロバイダーに適した **authselect** プロファイルを選択します。たとえば、LDAP を使用している企業のネットワークにログインするには、**sssd** を選択します。

  ```
  # authselect select sssd
  ```

  - オプション: **authselect select sssd** コマンドまたは **authselect select winbind** コマンドに次のオプションを追加することで、デフォルトのプロファイル設定を変更できます。次に例を示します。
    - **with-faillock**
    - **with-smartcard**
    - **with-fingerprint**

  利用可能なオプションの完全なリストを確認するには、authconfig から authselect へのスクリプトの変換 か、システム上の **authselect-migration(7)** man ページを参照してください。

注記

authselect select 手順を完了する前に、プロファイルに関連する設定ファイルが正しく設定されていることを確認してください。たとえば、**sssd** デーモンが正しく設定されておらずアクティブではない場合に **authselect select** を実行すると、ローカルユーザーのみが、**pam_unix** を使用して認証できるようになります。

検証

1. SSSD の **sss** エントリーが **/etc/nsswitch.conf** にあることを確認します。

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

2. **pam_sss.so** エントリーの **/etc/pam.d/system-auth** ファイルの内容を確認します。

```
# 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
...
```

関連情報

- システム上の **authselect-migration(7)** man ページ
- authselect の使用方法
- 既製の authselect プロファイルの変更
- 独自の authselect プロファイルの作成とデプロイメント

2.3. 既製の AUTHSELECT プロファイルの変更

システム管理者は、ニーズに合わせてデフォルトプロファイルのいずれかを変更することができます。

以下の項目を除き、**/etc/authselect/user-nsswitch.conf** ファイルを変更できます。

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

その後 **authselect select profile_name** を実行すると、**/etc/authselect/user-nsswitch.conf** から **/etc/nsswitch.conf** ファイルに許容可能な変更が転送されます。受け入れられない変更は、デフォルトのプロファイル設定によって上書きされます。

重要

/etc/nsswitch.conf ファイルを直接編集しないでください。

手順

1. **authselect** プロファイルを選択します。以下に例を示します。

```
# authselect select sssd
```

2. 必要な変更で **/etc/authselect/user-nsswitch.conf** ファイルを編集します。
3. **/etc/authselect/user-nsswitch.conf** ファイルから変更を適用します。

```
# authselect apply-changes
```

検証

- **/etc/nsswitch.conf** ファイルで、**/etc/authselect/user-nsswitch.conf** からの変更が伝播されているのを確認してください。

関連情報

- authselect の使用方法

2.4. 独自の AUTHSELECT プロファイルの作成とデプロイメント

システム管理者は、デフォルトプロファイルのいずれかのカスタムコピーを作成して、カスタムプロファイルを作成およびデプロイできます。

これは、同梱の authselect プロファイル を変更するのに特に便利です。カスタムプロファイルをデプロイすると、そのプロファイルは指定したホストにログインしているすべてのユーザーに適用されます。

手順

1. カスタムプロファイルを作成するために、**authselect create-profile** コマンドを実行します。**<custom_profile>** を目的のプロファイル名に置き換えます。たとえば、**/etc/nsswitch.conf** ファイル内の項目を自分で設定するオプションを使用して、既製の **sssd** プロファイルに基づいてプロファイルを作成するには、次のコマンドを使用します。

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

> **警告**
>
> **/etc/authselect/custom/<custom_profile>/{password-auth,system-auth,fingerprint-auth,smartcard-auth,postlogin}** を変更する予定の場合は、**--symlink-pam** オプションを指定せずに上記のコマンドを入力してください。これは、**authselect-libs** のアップグレード中に変更が確実に維持されるために行います。

コマンドに **--symlink-pam** オプションを含めると、PAM テンプレートがコピーではなく元のプロファイルファイルへのシンボリックリンクになります。また、**--symlink-meta** オプションを含めると、README や REQUIREMENTS などのメタファイルも、コピーではなく元のプロ

ファイルファイルへのシンボリックリンクになります。これにより、元のプロファイルの PAM テンプレートおよびメタファイルへの今後の更新が、カスタムプロファイルにも反映されます。

このコマンドにより、**/etc/authselect/custom/<custom_profile>/** ディレクトリーに **/etc/nsswitch.conf** ファイルのコピーが作成されます。

2. **/etc/authselect/custom/<custom_profile>/nsswitch.conf** ファイルを設定します。

3. パラメーターとして **custom/<custom_profile>** を指定した **authselect select** コマンドを実行し、カスタムプロファイルを選択します。

```
# authselect select custom/<custom_profile>
```

お使いのマシン用に **<custom_profile>** プロファイルを選択すると、後で Red Hat によって **sssd** プロファイルが更新されたときに、**/etc/nsswitch.conf** ファイルに加えた更新以外のすべての更新を利用できます。

例2.1 sssd プロファイルに基づくカスタムプロファイルの作成

sssd プロファイルに基づいてプロファイルを作成できます。このプロファイルでは、**dns** または **myhostname** データベース内ではなく、**/etc/hosts** ファイル内のホスト名のローカル静的テーブル検索のみを参照します。

1. **/etc/nsswitch.conf** ファイルで、次の行を編集します。

```
hosts:      files
```

2. **/etc/nsswitch.conf** への変更を除外するカスタムプロファイルを **sssd** に基づいて作成します。

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

3. プロファイルを選択します。

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

4. オプション: カスタムプロファイルを選択したことにより、以下が実行されたことを確認します。

   - 選択した **sssd** プロファイルに従って **/etc/pam.d/system-auth** ファイルが作成された。
   - **/etc/nsswitch.conf** の設定が変更されていない。

```
hosts:      files
```

注記

authselect select sssd を実行すると、対照的に **hosts: files dns myhostname** という結果になります。

–

関連情報

- authselect の使用方法

2.5. AUTHCONFIG から AUTHSELECT へのスクリプトの変換

ipa-client-install または **realm join** を使用してドメインに参加する場合は、スクリプトの **authconfig** 呼び出しを削除しても問題はありません。これができない場合は、各 **authconfig** コールを、同等の **authselect** コールに置き換えてください。その場合は、正しいプロファイルと適切なオプションを選択します。さらに、必要な設定ファイルを編集します。

- **/etc/krb5.conf**
- **/etc/sssd/sssd.conf** (**sssd** プロファイルの場合) または **/etc/samba/smb.conf** (**winbind** プロファイルの場合)

authconfig オプションと同等の authselect プロファイルオプション と authconfig オプションと authselect プロファイルの関係 では、**authconfig** オプションと同等の **authselect** を示しています。

表2.1 authconfig オプションと authselect プロファイルの関係

authconfig オプション	authselect プロファイル
--enableldap --enableldapauth	**sssd**
--enablesssd --enablesssdauth	**sssd**
--enablekrb5	**sssd**
--enablewinbind --enablewinbindauth	**winbind**
--enablenis	**nis**

表2.2 authconfig オプションと同等の authselect プロファイルオプション

authconfig オプション	authselect プロファイル機能
--enablesmartcard	**with-smartcard**
--enablefingerprint	**with-fingerprint**
--enableecryptfs	**with-ecryptfs**
--enablemkhomedir	**with-mkhomedir**
--enablefaillock	**with-faillock**

authconfig オプション	authselect プロファイル機能
--enablepamaccess	**with-pamaccess**
--enablewinbindkrb5	**with-krb5**

authconfig コマンドと同等の authselect コマンドの例 では、**authconfig** へのキックスタートの呼び出しを **authselect** へのキックスタートの呼び出しに変換する事例を紹介します。

表2.3 authconfig コマンドと同等の authselect コマンドの例

authconfig コマンド	同等の authselect コマンド
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 --enablewinbind --enablewinbindauth --winbindjoin=Administrator --updateall	**realm join -U Administrator --client-software=winbind WINBINDDOMAIN**

2.6. 関連情報

- What is pam_faillock and how to use it in Red Hat Enterprise Linux 8 & 9? (Red Hat ナレッジベース)
- Set Password Policy/Complexity in Red Hat Enterprise Linux 8 (Red Hat ナレッジベース)

第3章 SSSD とその利点について

System Security Services Daemon (SSSD) は、リモートディレクトリーと認証メカニズムにアクセスするためのシステムサービスです。SSSD の仕組み、SSSD を使用する利点、設定ファイルの処理方法、設定できるアイデンティティーおよび認証プロバイダーについて説明します。

3.1. SSSD の仕組み

System Security Services Daemon (SSSD) サービスを使用すると、リモートディレクトリーと認証メカニズムにアクセスできます。SSSD **クライアント** であるローカルシステムを、外部のバックエンドシステム (**プロバイダー**) に接続できます。

SSSD は、次のような複数のアイデンティティーおよび認証プロバイダーをサポートしています。

- LDAP ディレクトリー
- Identity Management (IdM) ドメイン
- Active Directory (AD) ドメイン
- Kerberos レルム

SSSD は、以下の 2 段階で機能します。

1. クライアントをリモートプロバイダーに接続し、ID 情報および認証情報を取得します。
2. 取得した認証情報を使用して、クライアントにユーザーと認証情報のローカルキャッシュを作成します。

ローカルシステムのユーザーは、リモートプロバイダーに保存されているユーザーアカウントを使用して認証できます。

SSSD は、ローカルシステムでユーザーアカウントを作成しません。ただし、SSSD は、IdM ユーザーのホームディレクトリーを作成するように設定できます。作成が完了すると、IdM ユーザーのホームディレクトリーと、クライアント上のコンテンツは、ユーザーがログアウトしても削除されません。

図3.1 SSSD の仕組み

SSSD は、NSS (Name Service Switch) や PAM (Pluggable Authentication Modules) などの複数のシステムサービスのキャッシュを提供することもできます。

注記

ユーザー情報のキャッシュには SSSD サービスのみを使用します。同じシステムでキャッシュ用に Name Service Caching Daemon (NSCD) と SSSD の両方を実行すると、パフォーマンスの問題や競合が発生する可能性があります。

3.2. SSSD を使用する利点

System Security Services Daemon (SSSD) を使用すると、ユーザーアイデンティティーの取得とユーザー認証に関してさまざまな利点が得られます。

オフライン認証

SSSD は、必要に応じて、リモートプロバイダーから取得したユーザー ID および認証情報のキャッシュを保持します。この設定では、セッションの開始時にすでにリモートプロバイダーに対して一度認証されている場合は、リモートプロバイダーまたはクライアントがオフラインであってもリソースに対して正常に認証できます。

単一のユーザーアカウント: 認証プロセスの一貫性の向上

SSSD では、オフライン認証用に中央アカウントとローカルユーザーアカウントの両方を維持する必要はありません。条件は次のとおりです。

- 特定のセッションでは、ユーザーが最低でも一度ログインしている必要があります。ユーザーが初めてログインしたときに、クライアントはリモートプロバイダーに接続する必要があります。
- SSSD でキャッシュを有効にする必要があります。
  SSSD を使用しないと、リモートユーザーには、多くの場合、複数のユーザーアカウントが存在します。たとえば、仮想プライベートネットワーク (VPN) に接続するには、リモートユーザーが、ローカルシステム用のアカウントのほかに、VPN システム用の別のアカウントが必要になります。このシナリオでは、最初にプライベートネットワーク上で認証して、リモートサーバーからユーザーを取得し、ユーザー認証情報をローカルでキャッシュする必要があります。

  SSSD では、キャッシュおよびオフライン認証により、リモートユーザーはローカルマシンに認証することで、ネットワークリソースに接続できます。SSSD は次にネットワークの認証情報を維持します。

ID プロバイダーおよび認証プロバイダーへの負荷の軽減

情報をリクエストすると、クライアントはまずローカルの SSSD キャッシュを確認します。SSSD は、キャッシュで情報が利用できない場合に限り、リモートプロバイダーに問い合わせます。

3.3. クライアントごとに複数の SSSD 設定ファイル

SSSD のデフォルト設定ファイルは **/etc/sssd/sssd.conf** です。このファイルとは別に、SSSD は、**/etc/sssd/conf.d/** ディレクトリーが ***.conf** ファイルのすべてからその設定を読み取ることができます。

この組み合わせにより、すべてのクライアントでデフォルトの **/etc/sssd/sssd.conf** ファイルを使用し、追加の設定ファイルに追加設定を追加して、クライアントごとに機能を個別に拡張できます。

SSSD が設定ファイルを処理する方法

SSSD は、以下の順番で設定ファイルを読み取ります。

1. プライマリー **/etc/sssd/sssd.conf** ファイル。

2. **/etc/sssd/conf.d/** 内のその他の ***.conf** ファイル。これらのファイルはアルファベット順に処理されます。

同じパラメーターが複数の設定ファイルに表示されると、SSSD は最後に読み取るパラメーターを使用します。

注記

SSSD は、**conf.d** ディレクトリー内の隠しファイル (**.** で始まるファイル) を読み込みません。

3.4. SSSD の ID プロバイダーおよび認証プロバイダー

SSSD クライアントは、外部 ID および認証プロバイダー (LDAP ディレクトリー、Identity Management (IdM)、Active Directory (AD) ドメイン、Kerberos レルムなど) に接続できます。次に、SSSD クライアントは SSSD プロバイダーを使用して ID および認証リモートサービスにアクセスします。SSSD が、異なる ID プロバイダーおよび認証プロバイダー、またはそれらの組み合わせを使用するように設定できます。

SSSD ドメインとしてのアイデンティティーおよび認証プロバイダー
アイデンティティーおよび認証プロバイダーは、SSSD 設定ファイル **/etc/sssd/sssd.conf** で **ドメイン** として設定します。プロバイダーは、ファイルの **[domain/<domain_name>]** または **[domain/default]** セクションにリストします。

1 つのドメインを次のいずれかのプロバイダーとして設定できます。

- UID や GID などのユーザー情報を提供する **アイデンティティープロバイダー**。
  - ドメインを **アイデンティティープロバイダー** として指定するには、**/etc/sssd/sssd.conf** ファイルの **[domain/<domain_name>]** セクションで **id_provider** オプションを使用します。
- 認証要求を処理する **認証プロバイダー**
  - ドメインを **認証プロバイダー** として指定するには、**/etc/sssd/sssd.conf** の **[domain/<domain_name>]** セクションで **auth_provider** オプションを使用します。
- 認可要求を処理する **アクセス制御プロバイダー**
  - ドメインを **アクセス制御プロバイダー** として指定するには、**/etc/sssd/sssd.conf** の **[domain/<domain_name>]** セクションで **access_provider** オプションを使用します。デフォルトでは、オプションは **permit** に設定されており、常にすべてのアクセスを許可します。詳細は **sssd.conf**(5) man ページを参照してください。
- 対応するすべての操作が 1 台のサーバー内で実行される場合など、これらのプロバイダーの組み合わせ
  - この場合、**id_provider**、**auth_provider**、**access_provider** オプションはすべて、**/etc/sssd/sssd.conf** の同じ **[domain/<domain_name>]** または **[domain/default]** セクションにリストします。

注記

SSSD に複数のドメインを設定できます。少なくともいずれかのドメインを設定する必要があります。設定しないと、SSSD は起動しません。

プロキシープロバイダー
プロキシープロバイダーは、SSSD と、SSSD が直接アクセスできないリソース間の中間リレーとして機能します。プロキシープロバイダーを使用する場合、SSSD はプロキシーサービスに接続し、プロキシーは指定されたライブラリーを読み込みます。

SSSD がプロキシープロバイダーを使用して以下を有効にするように設定できます。

- 指紋スキャナーなどの別の認証方法
- NIS などのレガシーシステム
- **/etc/passwd** ファイルで ID プロバイダーとして定義されるローカルシステムアカウントと、Kerberos などのリモート認証プロバイダー
- スマートカードを使用したローカルユーザーの認証

利用可能なアイデンティティーおよび認証プロバイダーの組み合わせ
次のアイデンティティープロバイダーと認証プロバイダーの組み合わせを使用するように SSSD を設定できます。

表3.1 利用可能なアイデンティティーおよび認証プロバイダーの組み合わせ

アイデンティティープロバイダー	認証プロバイダー
Identity Management [a]	Identity Management
Active Directory	Active Directory
LDAP	LDAP
LDAP	Kerberos
Proxy	Proxy
Proxy	LDAP
Proxy	Kerberos

[a] LDAP プロバイダータイプの拡張

関連情報

- authselect でユーザー認証の設定
- SSSD を使用したドメイン情報のクエリー [1]
- SSSD を使用したホストのユーザーアクセスに関するレポート

[1] **sssctl** ユーティリティーを使用してドメインのステータスをリスト表示して確認するには、Active Directory (AD) フォレストとの信頼関係にある Identity Management (IdM) にホストを登録する必要があります。

第4章 LDAP を使用し、TLS 認証を必要とする SSSD の設定

System Security Services Daemon (SSSD) は、Red Hat Enterprise Linux ホストで ID データの取得と認証を管理するデーモンです。システム管理者は、スタンドアロンの LDAP サーバーをユーザーアカウントデータベースとして使用するようにホストを設定できます。管理者は、LDAP サーバーとの接続を TLS 証明書で暗号化する必要があるという要件も指定できます。

注記

TLS を強制する SSSD 設定オプション **ldap_id_use_start_tls** のデフォルトは **false** です。ID 検索に TLS なしで **ldap://** を使用すると、攻撃ベクトル、つまり中間者 (MITM) 攻撃のリスクが発生します。これにより、LDAP 検索で返されるオブジェクトの UID または GID を変更することで、ユーザーの権限を借用する可能性があります。

セットアップが信頼できる環境で動作していることを確認し、**id_provider = ldap** に暗号化されていない通信を使用しても安全かどうかを判断してください。注記: **id_provider = ad** および **id_provider = ipa** は、SASL および GSSAPI によって保護された暗号化接続を使用するため、影響を受けません。

暗号化されていない通信を使用することが安全ではない場合は、**/etc/sssd/sssd.conf** ファイルで **ldap_id_use_start_tls** オプションを **true** に設定して TLS を強制する必要があります。

4.1. SSSD を使用して、暗号化された方法で LDAP からデータを取得する OPENLDAP クライアント

LDAP オブジェクトの認証方法は、Kerberos パスワードまたは LDAP パスワードのいずれかになります。LDAP オブジェクトの認証および認可に関する質問は、ここでは扱いません。

重要

LDAP で SSSD を設定するのは、SSSD および LDAP で高度な専門知識を必要とする複雑な手順です。代わりに、Active Directory や Red Hat Identity Management (IdM) などの統合型の自動ソリューションを使用することを検討してください。IdM の詳細は Identity Management の計画 を参照してください。

4.2. LDAP を使用し、TLS 認証を必要とする SSSD の設定

以下の手順に従って、Red Hat Enterprise Linux (RHEL) システムを OpenLDAP クライアントとして設定します。

以下のクライアント設定を使用します。

- RHEL システムが OpenLDAP ユーザーアカウントデータベースに保存されているユーザーを認証する。
- RHEL システムが SSSD (System Security Services Daemon) サービスを使用してユーザーデータを取得する。
- RHEL システムが TLS で暗号化された接続で OpenLDAP サーバーと通信する。

> **注記**
>
> または、この手順に従って、RHEL システムを Red Hat Directory Server のクライアントとして設定することもできます。

前提条件

- OpenLDAP サーバーがインストールされ、ユーザー情報を含めて設定されている。
- LDAP クライアントとして設定するホストの root 権限がある。
- LDAP クライアントとして設定するホストで、**/etc/sssd/sssd.conf** ファイルが作成され、**ldap** を **autofs_provider** および **id_provider** として指定するように設定されている。
- OpenLDAP サーバー証明書を発行した認証局からの PEM 形式の証明書チェーンがあり、**core-dirsrv.ca.pem** という名前のローカルファイルに保存されている。

手順

1. 必要なパッケージをインストールします。

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

2. 認証プロバイダーを **sssd** に切り替えます。

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

3. OpenLDAP サーバーの SSL/TLS 証明書を発行した認証局からのルート CA 署名証明書チェーンを含む **core-dirsrv.ca.pem** ファイルを **/etc/openldap/certs** フォルダーにコピーします。

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

4. LDAP サーバーの URL と接尾辞を **/etc/openldap/ldap.conf** ファイルに追加します。

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

5. **/etc/openldap/ldap.conf** ファイルで、**/etc/openldap/certs/core-dirsrv.ca.pem** を参照する **TLS_CACERT** パラメーターの行を追加します。

```
# 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. **/etc/sssd/sssd.conf** ファイルで、環境の値を **ldap_uri** パラメーターおよび **ldap_search_base** パラメーターに追加し、**ldap_id_use_start_tls** を **True** に設定します。

```
[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. **/etc/sssd/sssd.conf** で、**[domain]** セクションの **ldap_tls_cacert** および **ldap_tls_reqcert** の値を変更して TLS 認証要件を指定します。

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

8. **/etc/sssd/sssd.conf** ファイルの権限を変更します。

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

9. SSSD サービスおよび **oddjobd** デーモンを再起動して有効にします。

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

10. オプション: LDAP サーバーが非推奨の TLS 1.0 または TLS 1.1 プロトコルを使用している場合は、クライアントシステムのシステム全体の暗号化ポリシーを LEGACY レベルに切り替えて、RHEL がこのプロトコルを使用して通信できるようにします。

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

詳細は、Red Hat カスタマーポータルのナレッジベース記事 Strong crypto defaults in RHEL 8 and deprecation of weak crypto algorithms、およびシステム上の **update-crypto-policies(8)** man ページを参照してください。

検証

- **id** コマンドを使用し、LDAP ユーザーを指定して、LDAP サーバーからユーザーデータを取得できることを確認します。

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

システム管理者は、**id** コマンドを使用して LDAP からユーザーをクエリーできるようになりました。このコマンドは、正しいユーザー ID とグループメンバーシップを返します。

第5章 ID プロバイダーおよび認証プロバイダーの追加設定

System Security Services Daemon (SSSD) は、リモートディレクトリーと認証メカニズムにアクセスするためのシステムサービスです。SSSD の主な設定ファイルは **/etc/sssd/sssd.conf** です。本章では、**/etc/sssd/sssd.conf** ファイルを次のように変更して、SSSD サービスおよびドメインを設定する方法を概説します。

- オフライン認証を有効にするため、SSSD による完全なユーザー名の解釈と出力方法を調整します。
- DNS サービスディスカバリー、シンプルアクセスプロバイダールール、および SSSD が LDAP アクセスフィルターを適用するように設定します。

5.1. SSSD が完全なユーザー名を解釈する方法の調整

SSSD は、完全なユーザー名の文字列を解析して、ユーザー名とドメインコンポーネントにします。デフォルトでは、SSSD は Python 構文の次の正規表現に基づいて、**<user_name>@<domain_name>** という形式の完全なユーザー名を解釈します。

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

注記

Identity Management および Active Directory プロバイダーの場合、デフォルトのユーザー名の形式は **<user_name>@<domain_name>** または **<NetBIOS_name>\<user_name>** です。

SSSD による完全なユーザー名の解釈方法は、**re_expression** オプションを **/etc/sssd/sssd.conf** ファイルに追加し、カスタム正規表現を定義することで調整できます。

- 正規表現をグローバルに定義するには、正規表現のグローバル例の定義 の例で示されているように **sssd.conf** ファイルの **[sssd]** セクションに正規表現を追加します。
- 特定のドメインに正規表現を定義するには、特定のドメインで正規表現の定義 の例にあるように、**sssd.conf** ファイルの対応するドメインセクション (**[domain/LDAP]** など) に正規表現を追加します。

前提条件

- **root** アクセス

手順

1. **/etc/sssd/sssd.conf** ファイルを開きます。
2. **re_expression** オプションを使用して、カスタムの正規表現を定義します。

   例5.1 正規表現のグローバルでの定義

   すべてのドメインに対してグローバルに正規表現を定義するには、**sssd.conf** ファイルの **[sssd]** セクションに **re_expression** を追加します。

   次の glob 表現を使用して、ユーザー名を **<domain>_<username>_** または **<domain>@<user_name>** という形式で定義できます。

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

例5.2 特定のドメインへの正規表現の定義

特定のドメインに個別に正規表現を定義するには、**sssd.conf** ファイルの対応するドメインセクションに **re_expression** を追加します。

次の glob 表現を使用して、LDAP ドメインのユーザー名を **<domain>_<username>_** または **<domain>@<user_name>** という形式で定義できます。

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

関連情報

- システム上の **sssd.conf(5)** man ページ

5.2. SSSD が完全なユーザー名を出力する方法の調整

/etc/sssd/sssd.conf ファイルで **use_fully_qualified_names** オプションが有効になっている場合、SSSD はデフォルトで次の拡張に基づいて、**<name>@<domain>** という形式で完全なユーザー名を出力します。

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

注記

use_fully_qualified_names が設定されていない場合や、信頼されるドメインに対して明示的に **false** に設定されている場合に、ドメインコンポーネントのないユーザー名のみを出力します。

full_name_format オプションを **/etc/sssd/sssd.conf** ファイルに追加してカスタム拡張を定義することで、SSSD による完全なユーザー名の出力形式を調整できます。

前提条件

- **root** アクセス

手順

1. **root** として **/etc/sssd/sssd.conf** ファイルを開きます。
2. すべてのドメインに対してグローバルに拡張を定義するには、**sssd.conf** の **[sssd]** セクションに **full_name_format** を追加します。

```
[sssd]
```

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

この場合、ユーザー名は **user@domain.test** と表示されます。

3. 特定のドメインのユーザー名出力形式を定義するには、**sssd.conf** の対応するドメインセクションに **full_name_format** を追加します。

   - **%2$s\%1$s** を使用して Active Directory (AD) ドメインの拡張を設定するには、以下を実行します。

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

     この場合、ユーザー名は **ad.domain\user** と表示されます。

   - **%3$s\%1$s** を使用して Active Directory (AD) ドメインの拡張を設定するには、以下を実行します。

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

     この場合、Active Directory ドメインのフラットドメイン名が **AD** に設定されている場合、ユーザー名は **AD\user** と表示されます。

注記

SSSD は、名前の設定で名前のドメインコンポーネントを削除できるため、認証エラーが発生する可能性があります。**full_name_format** を標準以外の値に設定すると、これを標準形式に変更するように要求する警告が表示されます。

関連情報

- システム上の **sssd.conf(5)** man ページの **SPECIAL SECTIONS** と **DOMAIN SECTIONS**

5.3. オフライン認証の有効化

SSSD は、デフォルトでは、ユーザーの認証情報をキャッシュしません。認証要求の処理時に、SSSD は常にアイデンティティープロバイダーに問い合わせします。プロバイダーが利用できない場合は、ユーザー認証に失敗します。

アイデンティティープロバイダーが利用できない場合にユーザーが認証できるようにするには、**/etc/sssd/sssd.conf** ファイルで **cache_credentials** を **true** に設定して認証情報キャッシュを有効にできます。キャッシュされた認証情報とは、パスワードと、2 要素認証が使用されている場合の最初の認証要素を指します。スマートカード認証の場合、**cache_credentials** を true に設定したり、追加の設定を行ったりする必要はありません。正常に実行されたオンライン認証がキャッシュに記録されている限り、オフラインでも動作するはずです。

重要

SSSD は、パスワードをプレーンテキストでキャッシュしません。パスワードのハッシュのみを保存します。

認証情報はソルト付きの SHA-512 ハッシュとして保存されますが、攻撃者がブルートフォース攻撃を使用してキャッシュファイルにアクセスし、パスワードを解読できた場合、セキュリティーリスクが生じる可能性があります。キャッシュファイルにアクセスするには、RHEL のデフォルトである特権アクセスが必要です。

前提条件

- **root** アクセス

手順

1. **/etc/sssd/sssd.conf** ファイルを開きます。
2. ドメインセクションで、**cache_credentials = true** 設定を追加します。

```
[domain/<domain_name>]
cache_credentials = true
```

3. 推奨 (任意): アイデンティティープロバイダーが利用できない場合に SSSD がオフライン認証を許可する期間の制限を設定します。
   a. SSSD と連携するように PAM サービスを設定します。
   b. **offline_credentials_expiration** オプションを使用して、時間制限を指定します。
      制限は日単位で設定されることに注意してください。

      たとえば、最終ログインに成功してから 3 日間、オフライン認証を可能にするには、以下を使用します。

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

関連情報

- システム上の **sssd.conf(5)** man ページ
- authselect を使用したユーザー認証の設定

5.4. DNS サービスディスカバリーの設定

DNS サービス検出を使用すると、アプリケーションが特定タイプの特定サービスに対して指定のドメインの SRV レコードを確認し、必要なタイプのサーバーを返すことができます。ID または認証サーバーが **/etc/sssd/sssd.conf** ファイルで明示的に定義されていない場合は、SSSD は DNS サービス検出を使用してサーバーを動的に検出できます。

たとえば、**sssd.conf** に **id_provider = ldap** 設定が含まれているものの、**ldap_uri** オプションでホスト名または IP アドレスが指定されていない場合に、SSSD は DNS サービス検出を使用してサーバーを動的に検出します。

注記

SSSD が検出するのは、プライマリーサーバーのみで、バックアップサーバーを動的には検出できません。

前提条件

- **root** アクセス

手順

1. **/etc/sssd/sssd.conf** ファイルを開きます。
2. プライマリーサーバーの値を **_srv_** に設定します。
   LDAP プロバイダーの場合、プライマリーサーバーは **ldap_uri** オプションを使用して設定されます。

```
[domain/<ldap_domain_name>]
id_provider = ldap
ldap_uri = _srv_
```

3. パスワード変更プロバイダーでサービス検出を有効にするには、サービスタイプを設定します。

```
[domain/<ldap_domain_name>]
id_provider = ldap
ldap_uri = _srv_

chpass_provider = ldap
ldap_chpass_dns_service_name = ldap
```

4. オプション: デフォルトでは、サービス検出はシステムホスト名のドメイン部分をドメイン名として使用します。別の DNS ドメインを使用するには、**dns_discovery_domain** オプションを使用してドメイン名を指定します。
5. オプション: デフォルトでは、サービス検出は LDAP サービスタイプをスキャンします。別のサービスタイプを使用するには、**ldap_dns_service_name** オプションを使用してタイプを指定します。
6. オプション: デフォルトでは、SSSD は IPv4 アドレスの検索を試みます。試行に失敗すると、SSSD は IPv6 アドレスの検索を試行します。この動作をカスタマイズするには、**lookup_family_order** オプションを使用します。
7. サービス検出を使用するすべてのサービスについて、DNS レコードを DNS サーバーに追加します。

```
_<service_name>.<protocol>.<domain_name> <TTL> <priority> <weight> <port_number>
<hostname>_
```

関連情報

- DNS サービス検出に関する RFC 2782
- システム上の **sssd.conf(5)** man ページ

5.5. SIMPLE アクセスプロバイダールールの設定

simple アクセスプロバイダーは、ユーザー名またはグループのリストに基づいてアクセスを許可または拒否します。これにより、特定のマシンへのアクセスを制限できます。

たとえば、**simple** アクセスプロバイダーを使用して、特定のユーザーまたはグループへのアクセスを制限できます。他のユーザーまたはグループは、設定済みの認証プロバイダーに対して正常に認証されている場合でもログインできません。

前提条件

- **root** アクセス

手順

1. **/etc/sssd/sssd.conf** ファイルを開きます。
2. **access_provider** オプションを **simple** に設定します。

```
[domain/<domain_name>]
access_provider = simple
```

3. ユーザーのアクセス制御ルールを定義します。
   a. ユーザーへのアクセスを許可するには、**simple_allow_users** オプションを使用します。
   b. ユーザーへのアクセスを拒否するには、**simple_deny_users** オプションを使用します。

重要

特定のユーザーへのアクセスを拒否する場合には、他のユーザーすべてにアクセスを自動的に許可します。特定ユーザーにアクセスを許可する方が拒否するよりも安全であると考えられます。

4. グループのアクセス制御ルールを定義します。以下のいずれかを選択します。
   a. グループへのアクセスを許可するには、**simple_allow_groups** オプションを使用します。
   b. グループへのアクセスを拒否するには、**simple_deny_groups** オプションを使用します。

重要

特定のグループへのアクセスを拒否する場合には、他のグループすべてに、アクセスを自動的に許可します。特定グループにアクセスを許可する方が拒否するよりも安全であると考えられます。

例5.3 特定のユーザーおよびグループへのアクセス許可

次の例では、**alice**、**bob**、および **engineers** グループのメンバーにアクセスを許可し、他のすべてのユーザーのアクセスを拒否します。

```
[domain/<domain_name>]
access_provider = simple
```

```
simple_allow_users = alice, bob
simple_allow_groups = engineers
```

重要

拒否リストを空にすると、すべてのユーザーがアクセスできるようになります。

注記

信頼できる AD ユーザーを **simple_allow_users** リストに追加する場合は、必ず完全修飾ドメイン名 (FQDN) 形式 (例: aduser@ad.example.com) を使用してください。異なるドメインの短縮名は同じである可能性があるため、これによりアクセス制御設定に関する問題が回避されます。

関連情報

- システム上の **sssd-simple(5)** man ページ

5.6. LDAP アクセスフィルターを適用するための SSSD 設定

/etc/sssd/sssd.conf で **access_provider** オプションが設定されている場合に、SSSD は指定されたアクセスプロバイダーを使用して、システムにアクセスできるユーザーを評価します。使用しているアクセスプロバイダーが LDAP プロバイダータイプの拡張である場合は、システムへのアクセス許可用にユーザーが一致する必要がある LDAP アクセス制御フィルターを指定することもできます。

たとえば、Active Directory (AD) サーバーをアクセスプロバイダーとして使用する場合は、Linux システムへのアクセスを制限できます。指定されたフィルターに該当しない他のユーザーはすべて、アクセスが拒否されます。

注記

アクセスフィルターは LDAP ユーザーエントリーにのみ適用されます。そのため、ネスト化されたグループでこのタイプのアクセス制御を使用すると機能しない可能性があります。ネストされたグループにアクセス制御を適用するには、**simple** アクセスプロバイダールールの設定 を参照してください。

重要

オフラインキャッシュを使用する場合、SSSD は、ユーザーが最後にオンラインログインの試行に成功したかどうかを確認します。直近のオンラインログイン中に正常にログインしたユーザーは、アクセスフィルターに一致しない場合でも、オフラインでログインできるようになります。

前提条件

- **root** アクセス

手順

1. **/etc/sssd/sssd.conf** ファイルを開きます。
2. **[domain]** セクションで、アクセス制御フィルターを指定します。

- LDAP の場合は、**ldap_access_filter** オプションを使用します。
- AD の場合は、**ad_access_filter** オプションを使用します。さらに、**ad_gpo_access_control** オプションを **disabled** に設定して、GPO ベースのアクセス制御を無効にする必要があります。

例5.4 特定の AD ユーザーへのアクセス許可

たとえば、**admins** ユーザーグループに属し、属性セットが **unixHomeDirectory** の AD ユーザーにのみアクセスを許可するには、以下を使用します。

```
[domain/<ad_domain_name>]
access provider = ad
[... file truncated ...]
ad_access_filter = (&(memberOf=cn=admins,ou=groups,dc=example,dc=com)
(unixHomeDirectory=*))
ad_gpo_access_control = disabled
```

SSSD は、エントリーの **authorizedService** または **host** 属性により結果を確認することもできます。実際、ユーザーエントリーおよび設定に応じて、全オプションの MDASH LDAP フィルター、**authorizedService** および **host** の MDASH を評価できます。**ldap_access_order** パラメーターは、評価すべき順に、使用するアクセスコントロールの手法をすべて表示します。

```
[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
```

関連情報

- システム上の **sssd-ldap(5)** および **sssd-ad(5)** man ページ

第6章 SSSD クライアント側のビュー

SSSD には **sss_override** ユーティリティーがあるので、ローカルマシンに固有の POSIX ユーザーまたはグループ属性の値を表示するローカルビューを作成できます。**ipa** を除くすべての **id_provider** 値のオーバーライドを設定できます。

ipa プロバイダーを使用している場合は、IPA で ID ビューを一元的に定義します。詳細は ID ビューを使用した IdM クライアントのユーザー属性値の上書き を参照してください。

SSSD のパフォーマンスに与える可能性のある悪影響については、SSSD パフォーマンスにおける ID ビューによる悪影響の可能性 を参照してください。

6.1. LDAP ユーザー名属性のオーバーライド

管理者は、LDAP のアカウントを使用するように既存のホストを設定できます。しかし、LDAP 内のユーザーの値 (名前、UID、GID、ホームディレクトリー、シェル) が、ローカルシステム上の値と異なる場合があります。ローカルの **username** を定義することで、LDAP の **username** 属性をオーバーライドできます。

前提条件

- **root** アクセス
- **sssd-tools** パッケージがインストールされている。

手順

1. ユーザーの現在の情報を表示します。

```
# id <ldap_username>
```

   <ldap_username> は、ユーザーの LDAP **username** に置き換えます。

2. ローカルユーザー名を追加します。

```
# sss_override user-add <ldap_username> -n <local_username>
```

   <ldap_username> は、LDAP **username** に置き換えます。**<local_username>** は、目的のローカルユーザー名に置き換えます。

3. **sss_override user-add** コマンドを使用して最初のオーバーライドを作成したら、SSSD を再起動して変更を反映します。

```
# systemctl restart sssd
```

検証

- ローカルユーザー名が追加されたことを確認します。

```
# id <local_username>
```

- オプション: ユーザーのオーバーライドを表示します。

```
# sss_override user-show <ldap_username>
user@ldap.example.com:_<local_username>_::::::
```

例6.1 ローカルユーザー名の定義

LDAP ユーザー **sjones** にローカルユーザー名 **sarah** を追加するには、次の手順を実行します。

1. LDAP ユーザー **sjones** の現在の情報を表示します。

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

2. ローカルユーザー名 **sarah** を追加します。

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

3. ローカルユーザー名が追加されており、ユーザーのオーバーライドが正しく表示されることを確認します。

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

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

関連情報

- システム上の **sss_override** man ページ

6.2. LDAP UID 属性のオーバーライド

管理者は、LDAP のアカウントを使用するように既存のホストを設定できます。しかし、LDAP 内のユーザーの値 (名前、UID、GID、ホームディレクトリー、シェル) が、ローカルシステム上の値と異なる場合があります。次の手順で別の UID を定義することにより、LDAP UID 属性をオーバーライドできます。

前提条件

- **root** アクセス
- **sssd-tools** パッケージがインストールされている。

手順

1. ユーザーの現在の UID を表示します。

```
# id -u <ldap_username>
```

<ldap_username> は、ユーザーの LDAP **username** に置き換えます。

2. ユーザーのアカウントの UID をオーバーライドします。

```
# sss_override user-add <ldap_username> -u <local_uid>
```

<ldap_username> は、ユーザーの LDAP **username** に置き換えます。**<local_uid>** は、新しい UID 番号に置き換えます。

3. インメモリーキャッシュを失効させます。

```
# sss_cache --users
```

4. **sss_override user-add** コマンドを使用して最初のオーバーライドを作成したら、SSSD を再起動して変更を反映します。

```
# systemctl restart sssd
```

検証

- ローカル UID が適用されていることを確認します。

```
# id -u <ldap_username>
```

- オプション: ユーザーのオーバーライドを表示します。

```
# sss_override user-show <ldap_username>
user@ldap.example.com::_<local_uid>_:::::
```

例6.2 ユーザーの LDAP UID のオーバーライド

ユーザー **sarah** の LDAP UID をローカル UID **6666** でオーバーライドするには、次の手順を実行します。

1. LDAP ユーザー **sarah** の現在の UID を表示します。

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

2. ユーザー sarah のアカウントの UID を UID 6666 でオーバーライドします。

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

3. インメモリーキャッシュを手動で失効させます。

```
# sss_cache --users
```

4. SSSD を再起動して変更を適用します。

```
# systemctl restart sssd
```

5. 新しい UID が適用されており、ユーザーのオーバーライドが正しく表示されることを確認します。

```
# id sarah
6666

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

関連情報

- システム上の **sss_override** man ページ

6.3. LDAP GID 属性のオーバーライド

管理者は、LDAP のアカウントを使用するように既存のホストを設定できます。しかし、LDAP 内のユーザーの値 (名前、UID、GID、ホームディレクトリー、シェル) が、ローカルシステム上の値と異なる場合があります。次の手順で別の GID を定義することにより、LDAP の GID 属性をオーバーライドできます。

前提条件

- **root** アクセス
- **sssd-tools** がインストールされている

手順

1. ユーザーの現在の GID を表示します。

   ```
   # id -g <ldap_username>
   ```

   <ldap_username> は、ユーザー名に置き換えます。

2. ユーザーのアカウントの GID をオーバーライドします。

   ```
   # sss_override user-add <ldap_username> -g <local_gid>
   ```

   <ldap_username> は、ユーザー名に置き換えます。**<local_gid>** は、ローカル GID 番号に置き換えます。

3. インメモリーキャッシュを失効させます。

   ```
   # sss_cache --users
   ```

4. **sss_override user-add** コマンドを使用して最初のオーバーライドを作成したら、SSSD を再起動して変更を反映します。

   ```
   # systemctl restart sssd
   ```

検証

- ローカル GID が適用されていることを確認します。

```
# id -g <ldap_username>
```

- オプション: ユーザーのオーバーライドを表示します。

```
# sss_override user-show <ldap_username>
user@ldap.example.com::: 6666::::
```

> **例6.3 ユーザーの LDAP GID のオーバーライド**
>
> ユーザー **sarah** の GID を GID **6666** でオーバーライドするには、次の手順を実行します。
>
> 1. ユーザー **sarah** の現在の GID を表示します。
>
> ```
> # id -g sarah
> 6003
> ```
>
> 2. ユーザー sarah のアカウントの GID を GID **6666** でオーバーライドします。
>
> ```
> # sss_override user-add sarah -g 6666
> ```
>
> 3. インメモリーキャッシュを手動で失効させます。
>
> ```
> # sss_cache --users
> ```
>
> 4. これが最初のオーバーライドである場合は、SSSD を再起動して変更を反映します。
>
> ```
> # systemctl restart sssd
> ```
>
> 5. 新しい GID が適用されており、ユーザーのオーバーライドが正しく表示されることを確認します。
>
> ```
> # id -g sarah
> 6666
>
> # sss_override user-show sarah
> user@ldap.example.com::6666:::::
> ```

関連情報

- システム上の **sss_override** man ページ

6.4. LDAP ホームディレクトリー属性のオーバーライド

管理者は、LDAP のアカウントを使用するように既存のホストを設定できます。しかし、LDAP 内のユーザーの値 (名前、UID、GID、ホームディレクトリー、シェル) が、ローカルシステム上の値と異なる場合があります。別のホームディレクトリーを定義することで、LDAP ホームディレクトリー属性をオーバーライドできます。

前提条件

- **root** アクセス
- **sssd-tools** がインストールされている

手順

1. ローカルに保存されているユーザーの現在のホームディレクトリーを表示します。

```
# getent passwd <ldap_username>
<ldap_username>:x:XXXX:XXXX::/home/<home_directory>:/bin/bash
```

<ldap_username> は、ユーザー名に置き換えます。ローカルで表示されるとおりのホームディレクトリーの値が出力に表示されます。この値は LDAP レコードとは異なる場合があります。

2. ユーザーのホームディレクトリーをオーバーライドします。

```
# sss_override user-add <ldap_username> -h <new_home_directory>
```

<ldap_username> は、ユーザー名に置き換えます。**<new_home_directory>** は、新しいホームディレクトリーに置き換えます。

3. SSSD を再起動して変更を適用します。

```
# systemctl restart sssd
```

検証

- 新しいホームディレクトリーが定義されていることを確認します。

```
# getent passwd <ldap_username>
<ldap_username>:x:XXXX:XXXX::/home/<new_home_directory>:/bin/bash
```

- オプション: ユーザーのオーバーライドを表示します。

```
# sss_override user-show <ldap_username>
user@ldap.example.com:::::::<new_home_directory>::
```

例6.4 ユーザーのホームディレクトリーのオーバーライド

ユーザー **sarah** のホームディレクトリーを **admin** でオーバーライドするには、次の手順を実行します。

1. ユーザー **sarah** の現在のホームディレクトリーを表示します。

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

2. ユーザー **sarah** のホームディレクトリーを新しいホームディレクトリー **admin** でオーバーライドします。

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

3. SSSD を再起動して変更を適用します。

```
# systemctl restart sssd
```

4. 新しいホームディレクトリーが定義されており、ユーザーのオーバーライドが正しく表示されることを確認します。

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

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

関連情報

- システム上の **sss_override** man ページ

6.5. LDAP シェル属性のオーバーライド

管理者は、LDAP のアカウントを使用するように既存のホストを設定できます。しかし、LDAP 内のユーザーの値 (名前、UID、GID、ホームディレクトリー、シェル) が、ローカルシステム上の値と異なる場合があります。別のシェルを定義することで、LDAP シェル属性をオーバーライドできます。

前提条件

- **root** アクセス
- **sssd-tools** がインストールされている

手順

1. ローカルに保存されているユーザーの現在のシェルを表示します。

```
# getent passwd <ldap_username>
<ldap_username>:x:XXXX:XXXX::/home/<home_directory>:_<currentshell>_
```

<ldap_username> は、ユーザー名に置き換えます。

2. ユーザーのシェルをオーバーライドします。

```
# sss_override user-add <ldap_username> -s <new_shell>
```

<ldap_username> は、ユーザー名に置き換えます。**<new_shell>** は、新しいシェルに置き換えます。

3. SSSD を再起動して変更を適用します。

```
# systemctl restart sssd
```

検証

- 新しいシェルが定義されていることを確認します。

```
# getent passwd <ldap_username>
<ldap_username>:x:XXXX:XXXX::/home/<home_directory>:_<new_shell>_
```

- オプション: ユーザーのオーバーライドを表示します。

```
# sss_override user-show <ldap_username>
user@ldap.example.com::::::_<new_shell>_:
```

例6.5 ユーザーのシェルのオーバーライド

ユーザー **sarah** のシェルを **/bin/bash** から **sbin/nologin** に変更するには、次のコマンドを実行します。

1. ユーザー **sarah** の現在のシェルを表示します。

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

2. ユーザー sarah のシェルを新しい **/sbin/nologin** シェルでオーバーライドします。

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

3. SSSD を再起動して変更を適用します。

```
# systemctl restart sssd
```

4. 新しいシェルが定義されており、ユーザーのオーバーライドが正しく表示されることを確認します。

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

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

関連情報

- システム上の **sss_override** man ページ

6.6. ホストのオーバーライドのリスト表示

管理者は、ホスト上の全ユーザーおよびグループのオーバーライドをリスト表示して、正しい属性がオーバーライドされていることを確認できます。

前提条件

- **root** アクセス
- **sssd-tools** がインストールされている

手順

- すべてのユーザーのオーバーライドをリスト表示します。

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

- すべてのグループのオーバーライドをリスト表示します。

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

6.7. ローカルのオーバーライドの削除

グローバル LDAP ディレクトリーで定義されているローカルオーバーライドを削除できます。

前提条件

- **root** アクセス
- **sssd-tools** がインストールされている

手順

- ユーザーアカウントのオーバーライドを削除するには、次のコマンドを使用します。

```
# sss_override user-del <local_username>
```

  <local_username> は、ユーザー名に置き換えます。変更はすぐに有効になります。

- グループのオーバーライドを削除するには、次のコマンドを使用します。

```
# sss_override group-del <group_name>
```

- **sss_override user-del** または **sss_override group-del** コマンドを使用して最初のオーバーライドを削除したら、SSSD を再起動して変更を反映します。

```
# systemctl restart sssd
```

ユーザーまたはグループのオーバーライドを削除すると、このオブジェクトのオーバーライドがすべて削除されます。

6.8. ローカルビューのエクスポートおよびインポート

ローカルのオーバーライドは、ローカルの SSSD キャッシュに保存されています。このキャッシュからファイルにユーザーおよびグループのオーバーライドをエクスポートして、バックアップを作成できます。バックアップを作成することでキャッシュが削除されても、後で設定を復元できます。

前提条件

- **root** アクセス
- **sssd-tools** がインストールされている

手順

- ユーザーおよびグループビューのバックアップを作成するには、以下を使用します。

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

- ユーザーおよびグループビューを復元するには、以下を使用します。

```
# 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7章 AD を認証プロバイダーとして使用する RHEL ホストの設定

システム管理者は、ホストを Active Directory (AD) に参加させることなく、Red Hat Enterprise Linux (RHEL) ホストの認証プロバイダーとして AD を使用できます。

この方法は、次の場合に使用してください。

- AD 管理者がホストの有効化と無効化を制御できないようにする必要がある。
- そのホスト (企業 PC) は、社内で 1 人のユーザーのみが使用する予定である。

重要

この方法は、ホストを AD に参加させない特別な理由がある場合にのみ使用してください。

代わりに、システムを AD または Red Hat Identity Management (IdM) に完全に参加させることを検討してください。RHEL ホストをドメインに参加させると、設定を簡単に管理できます。クライアントを直接 AD に参加させることに関連するクライアントアクセスライセンスについて懸念がある場合は、AD との信頼関係にある IdM サーバーを活用することを検討してください。IdM-AD 信頼の詳細は、IdM と AD との間のフォレスト間の信頼の計画 と IdM と AD との間のフォレスト間の信頼のインストール を参照してください。

この手順を完了すると、**AD_user** が **example.com** ドメインの AD ユーザーデータベースに設定されたパスワードを使用して **rhel_host** システムにログインできるようになります。Kerberos レルム **EXAMPLE.COM** は **example.com** ドメインに対応します。

前提条件

- **rhel_host** への root アクセスがある。
- **AD_user** ユーザーアカウントが **example.com** ドメインにある。
- Kerberos レルムが **EXAMPLE.COM** である。
- **realm join** コマンドを使用して **rhel_host** が AD に参加していない。
- **sssd-proxy** パッケージがインストールされている。

```
# dnf install sssd-proxy
```

手順

1. パスワードを割り当てずに、**AD_user** ユーザーアカウントをローカルに作成します。

```
# useradd AD_user
```

2. **/etc/nsswitch.conf** ファイルを開いて編集し、以下の行が含まれていることを確認します。

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

3. **/etc/krb5.conf** ファイルを開いて編集し、以下のセクションと項目が含まれるようにします。

```
# 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. **/etc/sssd/sssd.conf** ファイルを作成し、以下のセクションと行をそのファイルに追加します。

```
[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. **/etc/sssd/sssd.conf** ファイルの権限を変更します。

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

6. SSSD (Security System Services Daemon) を起動します。

```
# systemctl start sssd
```

7. SSSD を有効にします。

```
# systemctl enable sssd
```

8. **/etc/pam.d/system-auth** ファイルを開き、以下のセクションと行が含まれるように変更します。

```
# 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. **/etc/pam.d/system-auth** ファイルの内容を **/etc/pam.d/password-auth** ファイルにコピーします。**yes** と入力して、ファイルの現在の内容を上書きします。

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

検証

1. **AD_user** 用の Kerberos チケット保証チケット (TGT) を要求します。 **AD_user** のパスワードを必要に応じて入力します。

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

2. 取得した TGT を表示します。

```
# 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 は、Kerberos ドメイン **EXAMPLE.COM** の認証情報を使用して **rhel_host** に正常にログインしました。

第8章 SSSD を使用したホストのユーザーアクセスに関するレポート

SSSD (Security System Services Daemon) は、どのユーザーがクライアントにアクセスできるか、できないかを追跡します。本章では、**sssctl** ツールを使用してアクセス制御レポートを作成し、ユーザーデータを表示する方法を説明します。

前提条件

- SSSD パッケージがネットワーク環境にインストールされている。

8.1. SSSCTL コマンド

sssctl は、SSSD (Security System Services Daemon) のステータスに関する情報を取得できるように一貫した方法を提供するコマンドラインツールです。

sssctl ユーティリティーを使用して、以下の情報を収集できます。

- ドメインの状態
- クライアントユーザー認証
- 特定のドメインのクライアントへのユーザーアクセス
- キャッシュされたコンテンツに関する情報

sssctl ツールを使用すると、以下が可能になります。

- SSSD キャッシュの管理
- ログの管理
- 設定ファイルの確認

注記

sssctl ツールは、**sss_cache** ツールおよび **sss_debuglevel** ツールに代わるものです。

関連情報

- **sssctl --help**

8.2. SSSCTL を使用したアクセス制御レポートの生成

SSSD は、クライアントにログインできるユーザーを制御するため、レポートを実行しているマシンに適用されるアクセス制御ルールをリスト表示できます。

注記

キー配布センター (KDC) がロックアウトしたユーザーをツールが追跡しないため、アクセスレポートは正確ではありません。

前提条件

- 管理者権限でログインしている。

手順

- アクセス制御レポートを生成するには、次のコマンドを実行します。**<domain_name>** はドメイン名に置き換えます。

```
[root@client1 ~]# sssctl access-report <domain_name>
1 rule cached

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

8.3. SSSCTL を使用したユーザー認可の詳細の表示

System Security Services Daemon (SSSD) に依存するアプリケーションの認証および認可の問題をトラブルシューティングするには、**sssctl user-checks** コマンドを使用します。

sssctl user-checks <user_name> を実行すると、Name Service Switch (NSS) および D-Bus インターフェイスの InfoPipe レスポンダーから入手できるユーザーデータが表示されます。出力には、ユーザーが **system-auth** Pluggable Authentication Module (PAM) サービスを使用してログインすることが許可されているかどうかが表示されます。

コマンドには、2 つのオプションがあります。

- **-a** (PAM アクション用)
- **-s** (PAM サービス用)

-a および **-s** オプションを指定しない場合、**sssctl** ツールのデフォルトのオプション (**-a acct -s system-auth**) が使用されます。

前提条件

- 管理者権限でログインしている。

手順

- 特定ユーザーのユーザーデータを表示するには、以下を入力します。

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

関連情報

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

第9章 SSSD を使用したドメイン情報のクエリー

sssctl を使用すると、System Security Services Daemon (SSSD) からドメイン関連データを取得して分析できます。SSSD は、Identity Management (IdM) 内のドメインと、フォレスト間の信頼によって IdM に接続されている Active Directory 内のドメインをリスト表示できます。

利用可能なドメインをリスト表示し、そのステータスを確認して、アイデンティティーと認証の問題をトラブルシューティングできます。

9.1. SSSCTL を使用したドメインのリスト表示

sssctl domain-list コマンドを使用して、ドメイントポロジーの問題をデバッグできます。

注記

ステータスはすぐに利用できない可能性があります。ドメインが表示されない場合は、コマンドを繰り返します。

前提条件

- 管理者権限でログインしている。

手順

1. オプション: **sssctl** コマンドのヘルプを表示するには、次のように入力します。

```
[user@client1 ~]$ sssctl --help
....
```

2. 利用可能なドメインのリストを表示するには、以下を入力します。

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

リストには、Active Directory と Identity Management との間でフォレスト間の信頼関係にあるドメインが含まれます。

9.2. SSSCTL でドメインステータスの確認

sssctl domain-status コマンドを使用して、ドメイントポロジーの問題をデバッグできます。

注記

ステータスはすぐに利用できない可能性があります。ドメインが表示されない場合は、コマンドを繰り返します。

前提条件

- 管理者権限でログインしている。

手順

1. オプション: **sssctl** コマンドのヘルプを表示するには、次のように入力します。

```
[user@client1 ~]$ sssctl --help
```

2. 特定のドメインのユーザーデータを表示するには、次のように入力します。**<domain_name>** は実際のドメイン名に置き換えます。

```
[root@client1 ~]# sssctl domain-status <domain_name>
```

ドメイン idm.example.com の出力例

```
Online status: Online

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

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

ドメイン **idm.example.com** はオンラインになり、コマンドを設定したクライアントから表示されます。

ドメインが利用できないと、結果は以下のようになります。

```
[root@client1 ~]# sssctl domain-status <domain_name>
Unable to get online status
```

第10章 SSSD を使用した PAM サービスのドメインの制限

プラグ可能な認証モジュール (PAM) は、認証および認可の一般的なフレームワークです。Red Hat Enterprise Linux のほとんどのシステムアプリケーションは、認証と認可の基礎となる PAM 設定に依存しています。

SSSD (System Security Services Daemon) を使用すると、PAM サービスがアクセスできるドメインを制限できます。SSSD は、特定の PAM サービスを実行するユーザーに基づいて PAM サービスからの認証要求を評価します。つまり、PAM サービスユーザーが SSSD ドメインにアクセスできる場合は、PAM サービスもそのドメインにアクセスできることを意味します。

10.1. PAM について

Pluggable Authentication Module (PAM) は、システムアプリケーションが、中央で設定したフレームワークに認証を中継するのに使用できる集中認証メカニズムを提供します。

PAM モジュールは、Kerberos、SSSD、NIS、ローカルファイルシステムなどのさまざまなタイプの認証ソースに存在するため、PAM はプラグ可能です。別の認証ソースに優先順位を付けることができます。

このモジュラーアーキテクチャーにより、管理者はシステムの認証ポリシーを柔軟に設定することができます。PAM は、開発者および管理者にとって以下のような便利なシステムです。

- PAM は、幅広いアプリケーションで使用できる一般的な認証スキームを提供します。
- PAM は、システム管理者に対して、優れた柔軟性と制御性を提供します。
- PAM では、完全にドキュメント化されたライブラリーが 1 つ提供され、開発者が独自の認証スキームを作成しなくてもプログラムを作成できるようになります。

10.2. ドメインアクセス制限のオプション

選択したドメインへのアクセスを制限するには、次のオプションを使用できます。

/etc/sssd/sssd.conf の pam_trusted_users

SSSD が信頼する PAM サービスの数値 UID またはユーザー名をリスト表示します。デフォルト設定は **all** です。つまり、すべてのサービスユーザーが信頼され、どのドメインにもアクセスできます。

/etc/sssd/sssd.conf の pam_public_domains

信頼されていない PAM サービスユーザーがアクセスできるパブリック SSSD ドメインを指定します。このオプションには、**all** および **none** の値を指定できます。デフォルト値は **none** です。つまり、ドメインが公開されておらず、信頼できないサービスユーザーはどのドメインにもアクセスできません。

PAM 設定ファイルの domains

PAM サービスが認証できるドメインのリストを指定します。ドメインを指定せずに **domains** を使用すると、PAM サービスがどのドメインに対しても認証できなくなります。次に例を示します。

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

PAM 設定ファイルで **domains** が使用されている場合、PAM サービスは、信頼できるユーザーで実行されているときに、すべてのドメインに対して認証できます。

/etc/sssd/sssd.conf SSSD 設定ファイルの **domains** オプションでも、SSSD が認証を試みるドメインのリストを指定します。PAM 設定ファイルの **domains** オプションは、**sssd.conf** 内のドメイン

のリストを拡張することはできず、より短いリストを指定して **sssd.conf** のドメインのリストを制限することしかできないことに注意してください。そのため、ドメインが PAM ファイルで指定されていても、**sssd.conf** では指定されていない場合、PAM サービスはドメインに対して認証を行いません。

デフォルト設定の **pam_trusted_users = all** および **pam_public_domains = none** は、すべての PAM サービスユーザーを信頼し、任意のドメインへのアクセスを許可することを指定します。ドメインへのアクセスを制限するには、PAM 設定ファイルの **domains** オプションを使用します。

sssd.conf に **pam_public_domains** が含まれている場合に、PAM 設定ファイルで **domains** を使用してドメインを指定するには、**pam_public_domains** でもドメインを指定する必要があります。必要なドメインを含まない **pam_public_domains** オプションを使用すると、信頼できないユーザーでサービスを実行している場合に、ドメインに対する PAM サービスの認証が失敗します。

注記

PAM 設定ファイルで定義するドメインの制限は、認証アクションにのみ適用され、ユーザーのルックアップには適用されません。

関連情報

- システム上の **sssd.conf(5)** man ページ
- システム上の **pam_sss(8)** man ページ

10.3. PAM サービスのドメインの制限

PAM サービス認証を指定ドメインに制限できます。

前提条件

- SSSD がインストールされ、実行している。

手順

1. 必要なドメインにアクセスするように SSSD を設定します。**/etc/sssd/sssd.conf** ファイルの **domains** オプションで、SSSD が認証できるドメインを定義します。

```
[sssd]
domains = <idm.example.com>, <ad.example.com>, <ldap.example.com>
```

2. PAM 設定ファイルで `domains` オプションを設定して、PAM サービスが認証できるドメインを指定します。以下に例を示します。

```
auth        sufficient    pam_sss.so forward_pass domains=<idm.example.com>
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
password    sufficient    pam_sss.so use_authtok
```

この設定により、PAM サービスの認証先が **<idm.example.com>** だけに制限されます。

検証

- **<idm.example.com>** に対して認証します。認証が正常に実行されるはずです。

10.4. PAM 設定ファイルについて

PAM 設定ファイルは、Red Hat Enterprise Linux のサービスの認証方法とポリシーを指定します。各 PAM 対応アプリケーションまたはサービスには、**/etc/pam.d/** ディレクトリー内に対応するファイルがあります。ファイルは、アクセスを制御するサービスにちなんで命名されます。たとえば、**login** プログラムには、**/etc/pam.d/login** という名前の、対応する PAM 設定ファイルがあります。

> **警告**
>
> PAM 設定ファイルを手動で編集すると、認証やアクセスの問題が発生する可能性があります。**authselect** ツールを使用して PAM を設定します。

PAM 設定ファイル形式

各 PAM 設定ファイルは、特定のモジュールの設定を定義するディレクティブで構成されています。PAM はいくつかのモジュール向けの認証中に引数を使用して情報をプラグ可能なモジュールに渡します。

```
module_type control_flag <module_name> <module_arguments>
```

以下に例を示します。

```
auth required pam_unix.so
```

PAM モジュールタイプ

PAM モジュールタイプは、モジュールが実行する認証タスクのタイプを指定します。モジュールは次のタスクを実行できます。

- **アカウント管理**
- **認証管理**
- **パスワード管理**
- **セッション管理**

個々のモジュールは、任意のモジュールタイプまたはすべてのモジュールタイプを提供できます。たとえば、**pam_unix.so** は 4 つのモジュールタイプをすべて提供します。

pam_unix.so などのモジュール名は、指定されたモジュールタイプを含むライブラリーの名前を PAM に提供します。ディレクトリー名は省略されています。アプリケーションが適切なバージョンの **libpam** にリンクされており、正しいバージョンのモジュールを検出できるためです。

モジュールタイプのディレクティブは積み重ねたり、重ねて配置したりできるため、複数のモジュールを 1 つの目的で一緒に使用できます。モジュールの順序は、制御フラグとともに重要であり、サービスに対するユーザーの認証という全体的な目標に対して、特定のモジュールの成功または失敗がどの程度重要であるかを決定します。

PAM モジュールをスタックして、ユーザーが認証される前に満たす必要がある特定の条件を適用できます。

PAM 制御フラグ
PAM モジュールが機能を実行すると、成功または失敗の結果が返されます。制御フラグは、この結果をどのように処理するかを PAM に指示します。

単純なフラグではキーワードを使用します。より複雑な構文は **[<value1>=<action1> <value2>=<action2> ...]** という形式に従います。

注記

モジュールの制御フラグが **sufficient** な値または **requisite** な値を使用する場合は、モジュールがリストされる順序は認証プロセスにとって重要になります。

オプションのリストを含む PAM 制御フラグの詳細な説明は、**pam.conf(5)** の man ページを参照してください。

関連情報

- システム上の **pam.conf(5)** man ページ
- システム上の **pam(8)** man ページ

10.5. PAM 設定例

詳細な説明を含む PAM 設定ファイルの例を参照してください。

アノテーション付き PAM 設定例

```
#%PAM-1.0
auth  required pam_securetty.so (1)
auth  required pam_unix.so nullok (2)
auth  required pam_nologin.so (3)
account  required pam_unix.so (4)
password required pam_pwquality.so retry=3 (5)
password required pam_unix.so shadow nullok use_authtok (6)
session  required pam_unix.so (7)
```

1. この行は、**/etc/securetty** ファイルが存在する場合は、そのファイルにリストされている端末からのみ root ログインが許可されることを保証します。ファイルに端末がリストされていない場合、root としてのログインは失敗し、**Login incorrect** メッセージが表示されます。

2. ユーザーにパスワードの入力を要求し、それを **/etc/passwd** に保存されている情報と照合し、存在する場合は **/etc/shadow** にも照合します。**nullok** 引数は空のパスワードを許可します。

3. **/etc/nologin** ファイルが存在するかどうかを確認します。ユーザーが存在して root でない場合は、認証に失敗します。

注記

この例では、最初の **auth** モジュールが失敗した場合でも、3 つの **auth** モジュールすべてがチェックされます。これは、認証がどの段階で失敗したかを潜在的な攻撃者に知られないようにする優れたセキュリティーアプローチです。

4 ユーザーのアカウントを確認します。たとえば、シャドウパスワードが有効になっていると、**pam_unix.so** モジュールのアカウントタイプは、期限切れのアカウントがあるかどうか、またはユーザーがパスワードを変更する必要があるかどうかをチェックします。

5 現在のパスワードの有効期限が切れた場合、またはユーザーが手動でパスワードの変更を要求した場合に、新しいパスワードの入力を求めます。次に、新しく作成されたパスワードの強度をチェックし、品質要件を満たしており、辞書ベースのパスワードクラッキングプログラムにより簡単に判別できないことを確認します。引数 **retry=3** は、ユーザーが強力なパスワードを作成するために 3 回試行できることを指定します。

6 **pam_unix.so** モジュールはパスワードの変更を管理します。**shadow** 引数は、ユーザーのパスワードを更新するときにモジュールにシャドウパスワードを作成するように指示します。**nullok** 引数を使用すると、ユーザーは空のパスワードを変更できます。それ以外の場合、null パスワードはアカウントロックとして扱われます。**use_authtok** 引数は、ユーザーに再度プロンプトを表示せずに、以前に入力されたパスワードを受け入れます。この方法では、すべての新しいパスワードは、受け入れられる前に、安全なパスワードであるかどうかの **pam_pwquality.so** チェックに合格する必要があります。

7 **pam_unix.so** モジュールはセッションを管理し、各セッションの開始時と終了時に、ユーザー名とサービスタイプをログに記録します。ログは **systemd-journald** サービスによって収集され、**journalctl** コマンドを使用して表示できます。ログは **/var/log/secure** にも保存されます。このモジュールは、追加機能のために他のセッションモジュールとスタックすることで補足できます。

第11章 NSLCD から SSSD への認証の移行

11.1. RHEL クライアントを NSLCD から SSSD に移行する

nss-pam-ldapd パッケージが RHEL から削除されたため、Red Hat は **SSSD** とその **ldap** プロバイダーに移行することを推奨します。これは **nslcd** サービスの機能を置き換えます。次の手順では、**nss-pam-ldap** 認証設定を使用するように以前に設定されたクライアントで LDAP ユーザーを認証するように **SSSD** を設定する方法について説明します。

前提条件

- RHEL クライアントは RHEL8 または RHEL 9 上にあります。
- 以前に、**nslcd** サービスを使用して LDAP ディレクトリーサーバーに対して認証するように RHEL クライアントを設定しました。
- LDAP ディレクトリーサービスは、RFC-2307 で定義されたスキーマを使用します。

手順

1. 現在の認証設定をバックアップします。

```
# authselect apply-changes -b --backup=ldap-configuration-backup
```

2. **SSSD** パッケージをインストールします。

```
# yum install sssd-ldap sssd-ad sssd-client \ sssd-common sssd-common-pac \ sssd-krb5 sssd-krb5-common
```

3. **nslcd** および **nscd** サービスを停止して無効にします。

```
# systemctl stop nslcd nscd
# systemctl disable nslcd nscd
```

4. **SSSD** で認証を設定します:

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

5. **SSSD** 設定ファイルに必要な所有権と権限を設定します。

```
# chown root:root /etc/sssd/sssd.conf
# chmod 600 /etc/sssd/sssd.conf
```

6. **/etc/sssd/sssd.conf** ファイルを開いて編集します。

7. 次の設定を入力し、**example.com** や **dc=example,dc=com** などの値をご使用の環境に適した値に置き換えます。

```
[sssd]
config_file_version = 2
services = nss, pam
domains = EXAMPLE.COM
```

```
debug_level = 6

[domain/EXAMPLE.COM]
id_provider = ldap
auth_provider = ldap
ldap_uri = ldap://server.example.com/
ldap_search_base = dc=example,dc=com
ldap_default_bind_dn = CN=binddn,DC=example,DC=com
ldap_default_authtok_type = password
ldap_default_authtok = <bind_account_password>
cache_credentials = True
```

> **注記**
>
> **SSSD** 設定で LDAP スキーマを指定する必要がある場合があります。
>
> ディレクトリーサーバーで RFC-2307bis スキーマを使用している場合は、**[domain/EXAMPLE.COM]** セクションに次の行を追加します。
>
> ```
> ldap_schema = rfc2307bis
> ```
>
> Microsoft Active Directory サーバーを使用している場合は、**[domain/EXAMPLE.COM]** セクションに次の行を追加して、LDAP ベースの認証を有効にします。
>
> ```
> ldap_schema = ad
> ```
>
> Kerberos 認証が必要な場合、Red Hat は、**SSSD** サービスを自動的に設定する **realm** コマンドを使用して RHEL クライアントを AD ドメインに参加させることを推奨します。

8. **SSSD** サービスを有効にして開始します。

```
# systemctl enable sssd
# systemctl start sssd
```

検証

1. LDAP ユーザーに関する情報を取得できることを確認します。

```
# id ldapuser
 uid=100424(ldapuser) gid=100424(ldapuser) groups=100424(ldapuser)

# getent passwd ldapuser
ldapuser:*: 100424: 100424:User, LDAP:/home/ldapuser:/bin/bash
```

2. LDAP ユーザーとしてログインできることを確認します。

```
# ssh -l ldapuser localhost
ldapuser@localhost's password:
Last login: Tue Dec 07 19:34:35 2021 from localhost
-sh-4.2$
```

> **注記**
>
> **nslcd** と **nscd** を使用して元の LDAP 設定を復元する必要がある場合は、次のコマンドを使用します。
>
> ```
> # authselect backup-restore=ldap-configuration-backup
> # systemctl stop sssd && systemctl disable sssd
> # systemctl start nslcd nscd
> # systemctl enable nslcd nscd
> ```

関連情報

- RFC-2307: ネットワーク情報サービスとして LDAP を使用するためのアプローチ
- インターネット-RFC-2307bis スキーマのドラフト

11.2. NSLCD.CONF オプションに相当する SSSD.CONF オプション

nslcd から **SSSD** への移行を支援するために、次の表に、**nslcd.conf** 設定ファイルの一般的なオプションと **sssd.conf** 設定ファイルの同等のオプションを示します。

表11.1 nslcd.conf オプションに相当する sssd.conf オプション

nslcd.conf オプション	sssd.conf オプション	説明
uid	該当なし	デーモンを実行するためのユーザー ID。デフォルトでは、SSSD は **sssd** ユーザーとして実行されます。
gid	該当なし	デーモンを実行する際に使用するグループ ID。デフォルトでは、SSSD は **sssd** プライベートグループとして実行されます。
uri	**ldap_uri**	次の形式の LDAP サーバーの URI: **ldap[s]://<host>[:port]**
base	**ldap_search_base**	検索ベースの識別名。
bindDN	**ldap_default_bind_dn**	LDAP 操作の実行に使用するデフォルトのバインド DN。
bindpw	**ldap_default_authtok**	デフォルトのバインド DN の認証トークン。現在、クリアテキストのパスワードのみがサポートされています。

nslcd.conf オプション	sssd.conf オプション	説明
ssl start_tls	**ldap_id_use_start_tls = true**	デフォルトのバインド DN の認証トークン。現在、クリアテキストのパスワードのみがサポートされています。
tls_reqcert	**ldap_tls_reqcert**	サーバー提供の証明書に対して実行するチェックを指定します。
tls_cacertfile	**ldap_tls_cacert**	すべての認証局の証明書を含むファイル。
tls_cacertdir	**ldap_tls_cacertdir**	個別のファイルに認証局証明書が含まれているディレクトリーのパス。
base passwd	**ldap_user_search_base**	ユーザーの LDAP 検索を制限するためのベース DN、検索範囲、LDAP フィルターをオプションで指定します。
base group	**ldap_group_search_base**	グループの LDAP 検索を制限するためのオプションのベース DN、検索範囲、および LDAP フィルター。

関連情報

- システム上の **nslcd.conf (5)** および **sssd-ldap (5)** の man ページ

第12章 ローカル SSSD 設定の誤字の排除

sssctl config-check コマンドを使用して、ホストの **/etc/sssd/sssd.conf** ファイルに誤字のエラーがあるかどうかをテストできます。

前提条件

- root でログインしている。
- **sssd-tools** パッケージがインストールされている。

手順

1. **sssctl config-check** コマンドを実行します。

```
# sssctl config-check

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

Messages generated during configuration merging: 0

Used configuration snippet files: 0
```

2. **/etc/sssd/sssd.conf** ファイルを開き、タイプミスを修正します。たとえば、前のステップでエラーメッセージが表示された場合は、**ldap_search** を **ldap_search_base** に置き換えます。

```
[...]
[domain/<domain_name>]
ldap_search_base = dc=<domain_component>,dc=<tld>
[...]
```

3. ファイルを保存します。
4. SSSD を再起動します。

```
# systemctl restart sssd
```

検証

- **sssctl config-check** コマンドを実行します。

```
# sssctl config-check
```

出力に問題が見つからなかったことが示されるはずです。

```
Issues identified by validators: 0

Messages generated during configuration merging: 0

Used configuration snippet files: 0
```

/etc/sssd/sssd.conf ファイルで、誤字がなくなりました。

第13章 IDM で SSSD を使用した認証のトラブルシューティング

Identity Management (IdM) 環境での認証には、以下に示す多くのコンポーネントが関係します。

IdM クライアント:

- SSSD サービス
- Name Services Switch (NSS)
- Pluggable Authentication Modules (PAM)

IdM サーバー:

- SSSD サービス
- IdM Directory Server
- IdM Kerberos Key Distribution Center (KDC)

Active Directory (AD) ユーザーとして認証する場合:

- AD ドメインコントローラー上の Directory Server
- AD ドメインコントローラー上の Kerberos サーバー

ユーザーを認証するには、SSSD サービスで以下の機能を実行できる必要があります。

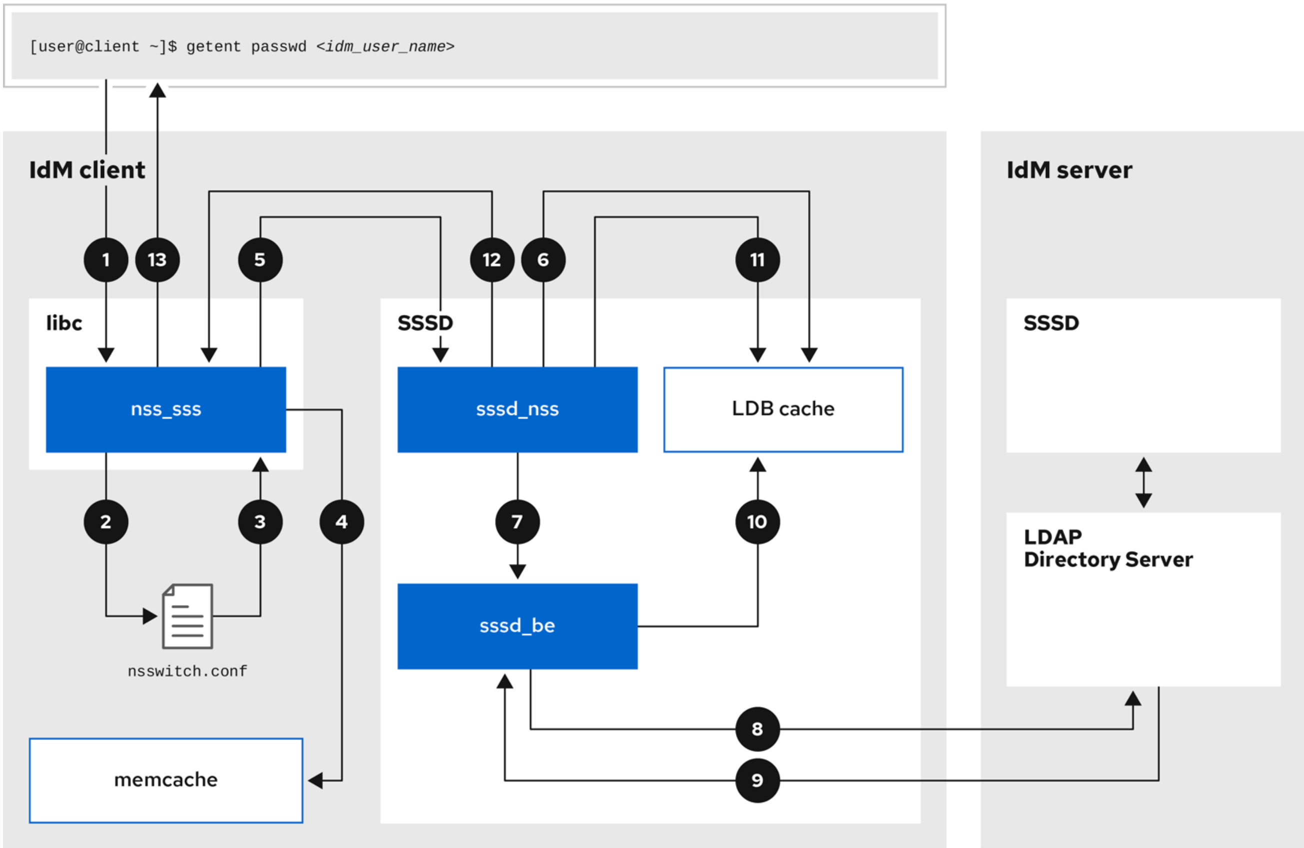
- 認証サーバーからユーザー情報を取得します。
- ユーザーに認証情報を要求し、それらの認証情報を認証サーバーに渡し、結果を処理します。

トラブルシューティングを実行するには、SSSD サービスとユーザー情報を保存するサーバー間で情報がどのように流れるかを把握する必要があります。これにより、問題が発生している場所を特定し、潜在的な原因を絞り込むことができます。

13.1. SSSD で IDM ユーザー情報を取得するデータフロー

以下の図は、**getent passwd <idm_user_name>** コマンドを使用して IdM ユーザー情報の要求時に IdM クライアントと IdM サーバーとの間の情報フローを簡単に説明します。

getent passwd を使用してユーザー情報を取得する場合の IdM クライアントとサーバー間の情報フロー



1. **getent** コマンドは、**libc** ライブラリーから **getpwnam** 呼び出しをトリガーします。
2. **libc** ライブラリーは、**/etc/nsswitch.conf** 設定ファイルを参照して、どのサービスがユーザー情報を提供するかを確認し、SSSD サービスのエントリー **sss** を検出します。
3. **libc** ライブラリーは、**nss_sss** モジュールを開きます。
4. nss_sss モジュールは、ユーザー情報のメモリーマップキャッシュを確認します。データがキャッシュに存在する場合は、**nss_sss** モジュールがそれを返します。
5. ユーザー情報が memory-mapped キャッシュにない場合、リクエストは SSSD **sssd_nss** レスポンダープロセスに渡されます。
6. SSSD サービスはキャッシュをチェックします。データがキャッシュに存在し、有効な場合は、**sssd_nss** レスポンダーがキャッシュからデータを読み取って、アプリケーションに返します。
7. データがキャッシュにない場合や期限切れである場合、**sssd_nss** レスポンダーは適切なバックエンドプロセスに対してクエリーを実行し、応答を待機します。SSSD サービスは、**sssd.conf** 設定ファイルで **id_provider=ipa** を設定して有効にした、IdM 環境の IPA バックエンドを使用します。
8. **sssd_be** バックエンドプロセスは IdM サーバーに接続して、IdM LDAP Directory Server から情報を要求します。
9. IdM サーバーの SSSD バックエンドは、IdM クライアントの SSSD バックエンドプロセスに対応します。

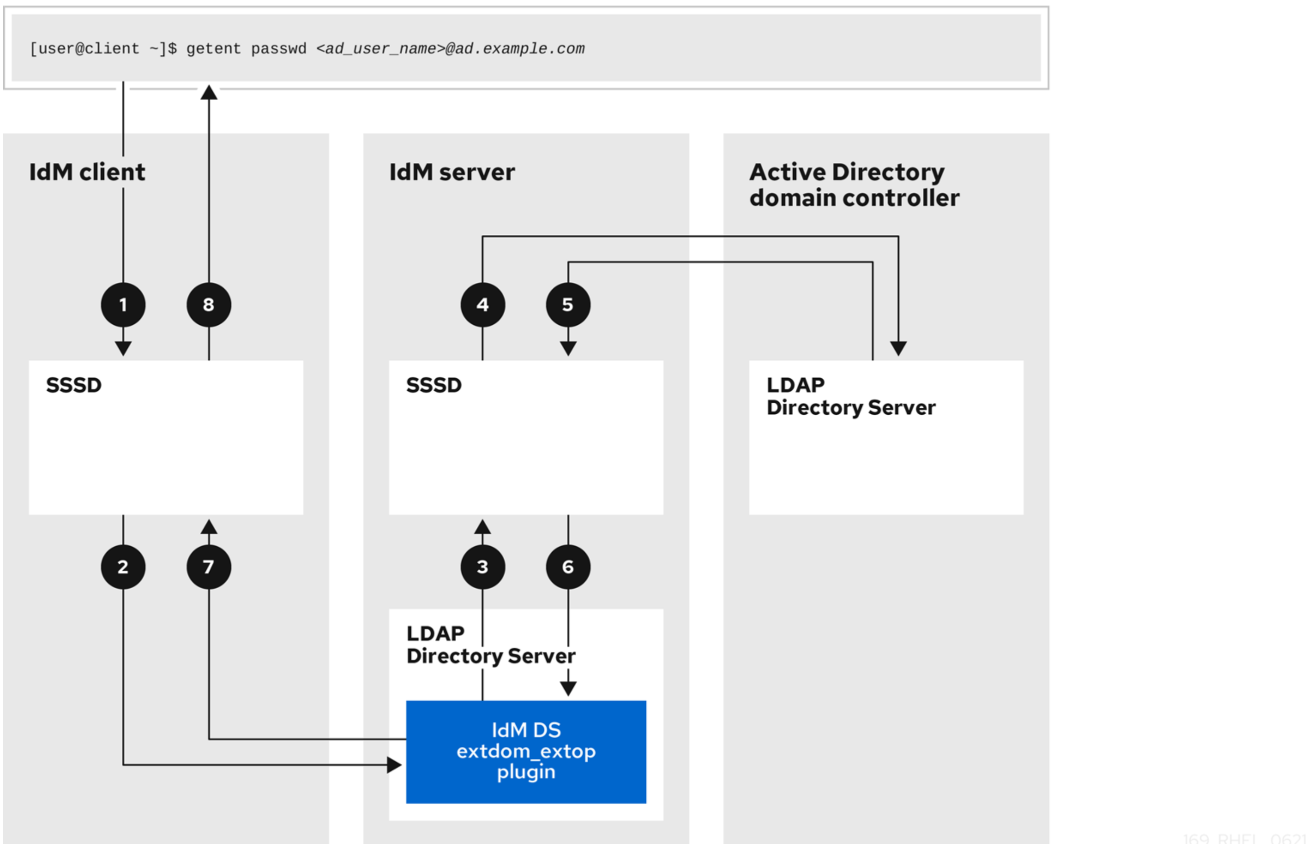
10. クライアントの SSSD バックエンドは、生成されるデータを SSSD キャッシュに保存し、キャッシュが更新されたレスポンダープロセスを警告します。
11. **sssd_nss** フロントエンドレスプロセスが SSSD キャッシュから情報を取得します。
12. **sssd_nss** レスポンダーは、**nss_sss** レスポンダーにユーザー情報を送信し、リクエストを完了します。
13. **libc** ライブラリーは、要求したアプリケーションにユーザー情報を返します。

13.2. SSSD で AD ユーザー情報を取得する際のデータフロー

IdM 環境と Active Directory(AD) ドメインとの間でフォレスト間の信頼を確立した場合は、IdM クライアントの AD ユーザー情報を取得する際に情報フローが、AD ユーザーデータベースへの追加の手順とともに、IdM クライアントの AD ユーザー情報の取得時に非常に似ています。

次の図は、ユーザーが **getent passwd <user_name@ad.example.com>** コマンドを使用して AD ユーザーに関する情報を要求した場合の情報フローを簡略化したものです。この図には、SSSD で IdM ユーザー情報を取得するデータフロー が含まれません。IdM クライアントの SSSD サービス、IdM サーバーの SSSD サービス、および AD ドメインコントローラー上の LDAP データベースとの間の通信にフォーカスします。

IdM クライアント、IdM サーバー、および AD ドメインコントローラー間の AD ユーザー情報を取得する場合の情報フロー



1. IdM クライアントは、AD ユーザー情報に関するローカルの SSSD キャッシュを検索します。

2. IdM クライアントにユーザー情報がない場合や、情報が古い場合に、クライアントの SSSD サービスが IdM サーバーの **extdom_extop** プラグインに問い合わせて、LDAP 拡張操作を実行し、情報を要求します。

3. IdM サーバーの SSSD サービスは、ローカルキャッシュで AD ユーザー情報を検索します。

4. IdM サーバーに SSSD キャッシュにユーザー情報がない場合や、その情報が古い場合は、LDAP 検索を実行して、AD ドメインコントローラーからユーザー情報を要求します。

5. IdM サーバーの SSSD サービスは、AD ドメインコントローラーから AD ユーザー情報を受け取り、キャッシュに保存します。

6. **extdom_extop** プラグインは、LDAP 拡張操作を完了する IdM サーバーの SSSD サービスから情報を受信します。

7. IdM クライアントの SSSD サービスは、LDAP 拡張操作から AD ユーザー情報を受信します。

8. IdM クライアントは、AD ユーザー情報を SSSD キャッシュに保存し、要求したアプリケーションに情報を返します。

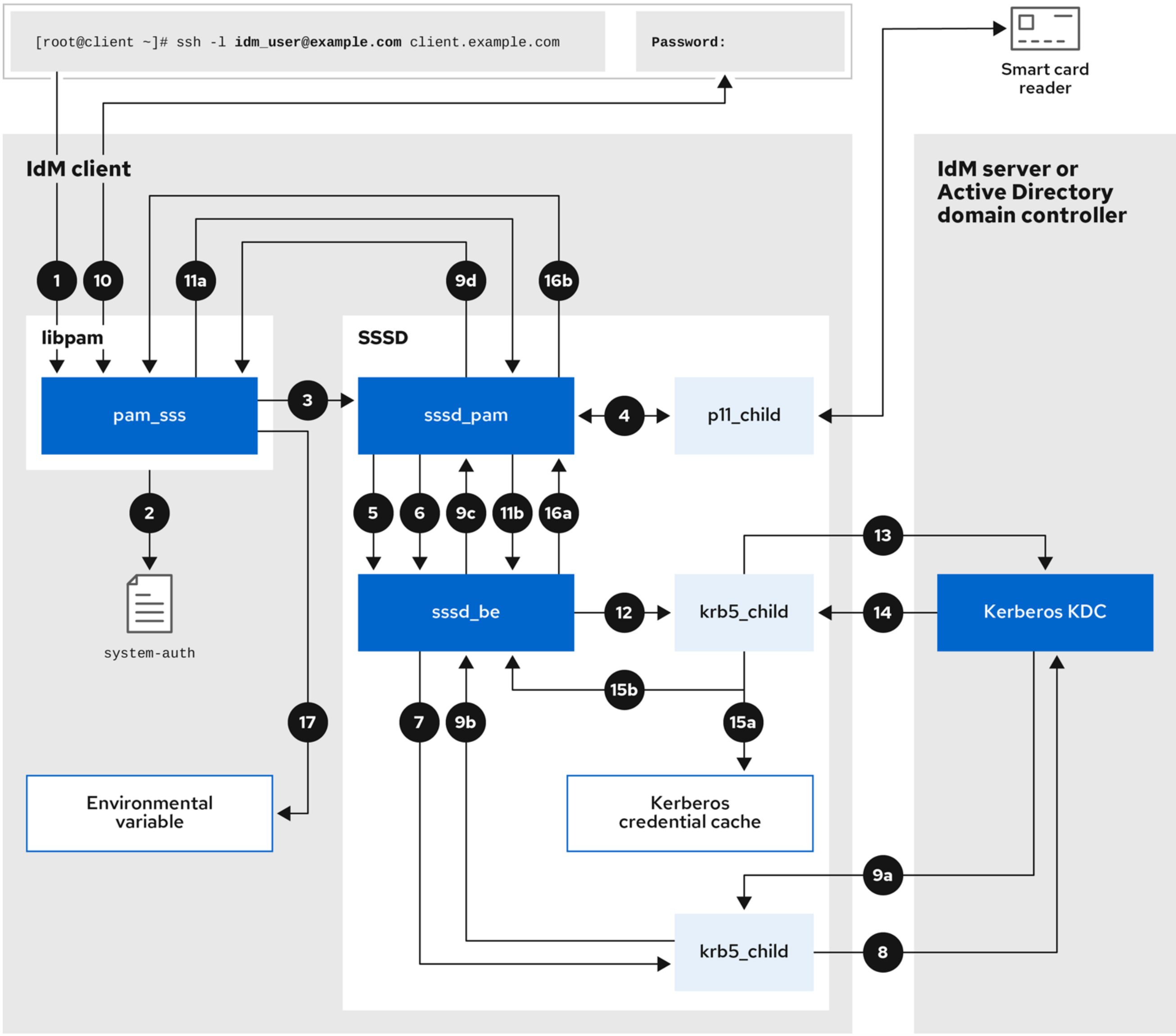
13.3. IDM で SSSD を使用してユーザーとして認証する場合にデータフロー

IdM サーバーまたはクライアントでユーザーとして認証するには、以下のコンポーネントが必要です。

- **sshd** サービスなど、認証要求を開始するサービス。
- Pluggable Authentication Module (PAM) ライブラリーとそのモジュール。
- SSSD サービス、そのレスポンダー、およびバックエンド。
- スマートカード認証が設定されている場合は、スマートカードリーダー。
- 認証サーバー:
  - IdM ユーザーは、IdM Kerberos Key Distribution Center (KDC) に対して認証されます。
  - Active Directory (AD) ユーザーは、AD ドメインコントローラー (DC) に対して認証されます。

以下の図は、コマンドラインの SSH サービスを介してホストにローカルでログインしようとすると、情報フローを簡単に認証する必要がある場合を示しています。

認証試行中の IdM クライアントと IdM サーバーまたは AD ドメインコントローラー間の情報の流れ



1. **ssh** コマンドで認証を試みると、**libpam** ライブラリーがトリガーされます。

2. **libpam** ライブラリーは、認証の試行を要求するサービスに対応する **/etc/pam.d/** ディレクトリーの PAM ファイルを参照します。この例では、ローカルホストの SSH サービス経由で認証されている例では、**libpam** ライブラリーは **/etc/pam.d/system-auth** 設定ファイルを確認し、SSSD PAM の **pam_sss.so** エントリーを検出します。

   ```
   auth    sufficient    pam_sss.so
   ```

3. 利用可能な認証方法を判断するため、**libpam** ライブラリーは **pam_sss** モジュールを開き、SSSD サービスの **sssd _pam** PAM レスポンダーに **SSS_PAM_PREAUTH** リクエストを送信します。

4. スマートカード認証が設定されていると、SSSD サービスは一時的な **p11_child** プロセスを生成し、スマートカードを確認し、そこから証明書を取得します。

5. ユーザーにスマートカード認証が設定されていると、**sssd_pam** レスポンダーは、スマートカードの証明書とユーザーを照合します。**sssd_pam** レスポンダーは、グループメンバーシップがアクセス制御に影響する可能性があるため、ユーザーが属するグループの検索も実行しま

す。

6. **sssd_pam** レスポンダーは、**SSS_PAM_PREAUTH** 要求を **sssd_be** バックエンドレスに送信し、パスワードや 2 要素認証などのサーバーがサポートする認証方法を表示します。SSSD サービスが IPA レスポンダーを使用する IdM 環境では、デフォルトの認証方法は Kerberos です。この例では、ユーザーは簡単な Kerberos パスワードで認証されます。

7. **sssd_be** レスポンダーは一時的な **krb5_child** プロセスを起動します。

8. **krb5_child** プロセスは、IdM サーバーの KDC に連絡して、利用可能な認証方法を確認します。

9. KDC はリクエストに応答します。

    a. **krb5_child** プロセスは応答を評価し、結果を **sssd_be** バックエンドプロセスに送信します。

    b. **sssd_be** バックエンドプロセスが結果を受け取ります。

    c. **sssd_pam** レスポンダーは結果を受け取ります。

    d. **pam_sss** モジュールは結果を受け取ります。

10. ユーザーにパスワード認証が設定されていると、**pam_sss** モジュールにより、パスワードの入力が求められます。スマートカード認証が設定されていると、**pam_sss** モジュールにより、スマートカードの PIN の入力が求められます。

11. モジュールがユーザー名とパスワードを使用して **SSS_PAM_AUTHENTICATE** 要求を送信します。これは次の宛先に送信されます。

    a. **sssd_pam** レスポンダー。

    b. **sssd_be** バックエンドプロセス。

12. **sssd_be** プロセスは、KDC に問い合わせる一時的な **krb5_child** プロセスを起動します。

13. **krb5_child** process は、ユーザー名とパスワードを使用して KDC から Kerberos チケット保証チケット (TGT) の取得を試みます。

14. **krb5_child** プロセスは、認証の試行の結果を受け取ります。

15. **krb5_child** プロセス:

    a. TGT を認証情報キャッシュに保存します。

    b. **sssd_be** バックエンドプロセスに認証結果を返します。

16. 認証結果は **sssd_be** プロセスから以下を行います。

    a. **sssd_pam** レスポンダー。

    b. **pam_sss** モジュール。

17. **pam_sss** モジュールは、その他のアプリケーションが参照できるように、環境変数をユーザーの TGT の場所で設定します。

13.4. 認証問題の範囲の制限

ユーザーを正常に認証するには、ユーザー情報を保存するデータベースから SSSD サービスでユーザー情報を取得できる必要があります。以下の手順では、認証プロセスの異なるコンポーネントをテストする手順を説明します。これにより、ユーザーがログインできない場合に認証の問題のスコープを制限する方法を説明します。

手順

1. SSSD サービスおよびそのプロセスが実行していることを確認します。

```
[root@client ~]# pstree -a | grep sssd
  |-sssd -i --logger=files
  |   |-sssd_be --domain implicit_files --uid 0 --gid 0 --logger=files
  |   |-sssd_be --domain <domain_name> --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. クライアントが IP アドレスを使用してユーザーデータベースサーバーに接続できることを確認します。

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

この手順が失敗した場合は、ネットワークとファイアウォール設定が、IdM クライアントとサーバー間の直接通信が許可されていることを確認してください。firewalld の使用および設定を参照してください。

3. クライアントが、完全修飾ホスト名を使用して IdM LDAP サーバー (IdM ユーザー用) または AD ドメインコントローラー (AD ユーザーの場合) を検出して連絡できることを確認します。

```
[user@client ~]$ dig -t SRV ldap._tcp.<domain>@<server_name>
[user@client ~]$ ping <fully_qualified_host_name_of_the_server>
```

この手順が失敗した場合は、**/etc/resolv.conf** ファイルを含む Dynamic Name Service (DNS) の設定を確認してください。DNS サーバーの順序の設定 を参照してください。

注記

デフォルトでは、SSSD サービスは DNS サービス (SRV) レコードを介して LDAP サーバーと AD DC を自動的に検出しようとします。SSSD を特定のサーバーに制限するには、**sssd.conf** 設定ファイルで次のオプションを使用してサーバーを定義します。

- **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>**

このオプションを使用する場合は、そのオプションに記載されているサーバーと通信できることを確認します。

4. クライアントが LDAP サーバーに対して認証でき、**ldapsearch** コマンドでユーザー情報を取得できることを確認します。

   a. LDAP サーバーが **server.example.com** などの IdM サーバーである場合は、ホストの Kerberos チケットを取得し、ホスト Kerberos プリンシパルで認証されるデータベース検索を実行します。

   ```
   [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=<idm_user>
   ```

   b. LDAP サーバーが **server.example.com** などの Active Directory (AD) Domain Controller (DC) サーバーである場合は、ホストの Kerberos チケットを取得し、ホスト Kerberos プリンシパルで認証されるデータベース検索を実行します。

   ```
   [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=<idm_user>
   ```

   c. LDAP サーバーがプレーン LDAP サーバーであり、**sssd.conf** ファイルに **ldap_default_bind_dn** および **ldap_default_authtok** オプションを設定した場合は、同じ **ldap_default_bind_dn** アカウントとして認証されます。

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

   この手順が失敗した場合は、データベース設定で、ホストが LDAP サーバーを検索できることを確認します。

5. SSSD サービスは Kerberos 暗号化を使用するため、ログインできないユーザーとして Kerberos チケットを取得できます。

   a. LDAP サーバーが IdM サーバーの場合:

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

   b. LDAP サーバーデータベースが AD サーバーの場合:

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

この手順が失敗した場合は、Kerberos サーバーが適切に動作し、すべてのサーバーが同期され、ユーザーアカウントがロックされていないことを確認します。

6. コマンドラインに関するユーザー情報を取得できることを確認します。

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

この手順が失敗した場合は、クライアントの SSSD サービスがユーザーデータベースから情報を受信できることを確認します。

a. **/var/log/messages** ログファイルのエラーを確認します。

b. SSSD サービスで詳細なロギングを有効にし、デバッグログを収集して、問題のソースに関するログを確認します。

c. オプション: Red Hat テクニカルサポートケースを作成し、収集したトラブルシューティング情報を提供します。

7. ホスト上で **sudo** 権限を持っている場合は、**sssctl** ユーティリティーを使用して、ユーザーがログインできるかどうかを確認します。

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

この手順が失敗した場合は、PAM 設定、IdM HBAC ルール、IdM RBAC ルールなどの認可設定を確認します。

a. ユーザーの UID が、**/etc/login.defs** ファイルで定義されている **UID_MIN** 以上であることを確認してください。

b. **/var/log/secure** ログファイルおよび **/var/log/messages** ログファイルで認可エラーを確認します。

c. SSSD サービスで詳細なロギングを有効にし、デバッグログを収集して、問題のソースに関するログを確認します。

d. オプション: Red Hat テクニカルサポートケースを作成し、収集したトラブルシューティング情報を提供します。

関連情報

- sssd.conf ファイルで SSSD の詳細なロギングの有効化
- sssctl コマンドを使用した SSSD の詳細なロギングの有効化
- IdM サーバーの認証問題をトラブルシューティングするために SSSD サービスからデバッグログを収集する
- IdM クライアントの認証問題をトラブルシューティングするために SSSD サービスからデバッグログを収集する

13.5. SSSD ログファイルおよびログレベル

それぞれの SSSD サービスは、**/var/log/sssd/** ディレクトリーに独自のログファイルを記録します。**example.com** IdM ドメインの IdM サーバーのログファイルは、以下のようになります。

```
[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
```

SSSD ログファイルの目的

krb5_child.log

Kerberos 認証に関連する有効期限の短いヘルパープロセスのログファイル。

ldap_child.log

LDAP サーバーとの通信用の Kerberos チケットの取得に関連する短期ヘルパープロセスのログファイル。

sssd_<domain_name>.log

sssd.conf ファイルのドメインセクションごとに、SSSD サービスは LDAP サーバーとの通信に関する情報を別のログファイルに記録します。たとえば、**example.com** という名前の IdM ドメインがある環境では、SSSD サービスは **sssd_example.com.log** という名前のファイルのログにその情報を記録します。ホストが **ad.example.com** という名前の AD ドメインと直接統合されている場合は、**sssd_ad.example.com.log** という名前のファイルのログに情報が記録されます。

> **注記**
>
> IdM 環境と、AD ドメインを持つフォレスト間の信頼があると、AD ドメインに関する情報は引き続き IdM ドメインのログファイルに記録されます。
>
> 同様に、ホストが AD ドメインに直接統合されている場合は、プライマリードメインのログファイルに、子ドメインに関する情報が書き込まれます。

selinux_child.log

SELinux 情報を取得および設定する短期間ヘルパープロセスのログファイル。

sssd.log

SSSD を監視して、レスポンダーおよびバックエンドプロセスと通信するためのログファイル。

sssd_ifp.log

InfoPipe レスポンダーのログファイル。システムバスからアクセス可能なパブリック D-Bus インターフェイスを提供します。

sssd_nss.log

ユーザーおよびグループ情報を取得する Name Services Switch (NSS) レスポンダーのログファイル。

sssd_pac.log

AD Kerberos チケットから PAC を収集する Microsoft Privilege Attribute Certificate (PAC) レスポンダー用のログファイルは、PAC から PAC に関する情報を取得します。これにより、AD ユーザーを直接要求しないようにします。

sssd_pam.log

PAM (Pluggable Authentication Module) レスポンダー用のログファイルです。

sssd_ssh.log

SSH レスポンダープロセスのログファイル。

SSSD ロギングレベル

デバッグレベルを設定すると、それ下のすべてのデバッグレベルが有効になります。たとえば、debug レベルを 6 に設定すると、デバッグレベル 0 から 5 も有効になります。

表13.1 SSSD ロギングレベル

レベル	説明
0	**致命的な障害が発生しました**。SSSD サービスが起動しなかったり、終了しないようにするエラー。これは、RHEL 8.3 以前のデフォルトのデバッグログレベルです。
1	**重大なエラー**SSSD サービスを終了しないものの、主要な機能は 1 つ以上正しく機能しません。
2	**深刻なエラー**。特定の要求または操作が失敗したことを示すエラー。これは、RHEL 8.4 以降のデフォルトのデバッグログレベルです。
3	**マイナーな障害が発生しました**。レベル 2 で操作の失敗がキャプチャーされたエラー。
4	**設定**。
5	**関数** データ。
6	**操作** 関数のトレースメッセージ。
7	**内部制御** 関数のトレースメッセージ。
8	**関数内部** 変数の内容。
9	非常に **低いレベルのトレース** 情報。

13.6. SSSD.CONF ファイルで SSSD の詳細なロギングの有効化

デフォルトでは、RHEL 8.4 以降の SSSD サービスは、重大な失敗 (デバッグレベル 2) のみをログに記録しますが、認証問題のトラブルシューティングに必要な詳細レベルではログに記録されません。

SSSD サービスの再起動時に詳細なロギングを有効にするには、**/etc/sssd/sssd.conf** 設定ファイルの各セクションに **debug_level=<integer>** オプションを追加します。ここで、**<integer>** の値は 0 から 9 の数字になります。デバッグレベルは最大 3 つのログで、最大 3 つのログで、レベル 8 以上では、多くの詳細なログメッセージを提供します。**レベル 6** は、認証の問題のデバッグに役立ちます。

前提条件

- **sssd.conf** 設定ファイルを編集し、SSSD サービスを再起動するには、root パスワードが必要です。

手順

1. テキストエディターで **/etc/sssd/sssd.conf** ファイルを開きます。
2. **debug_level** オプションをファイルのすべてのセクションに追加し、デバッグレベルを、選択した詳細に設定します。

```
[domain/<domain_name>]
debug_level = 6
id_provider = ipa
...

[sssd]
debug_level = 6
services = nss, pam, ifp, ssh, sudo
domains = <domain_name>

[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. **sssd.conf** ファイルを保存して閉じます。
4. SSSD サービスを再起動して、新しい設定を読み込みます。

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

関連情報

- SSSD ログファイルおよびログレベル

13.7. SSSCTL コマンドを使用した SSSD の詳細なロギングの有効化

デフォルトでは、RHEL 8.4 以降の SSSD サービスは、重大な失敗 (デバッグレベル 2) のみをログに記録しますが、認証問題のトラブルシューティングに必要な詳細レベルではログに記録されません。

sssctl debug-level <integer> コマンドを使用して、コマンドラインで SSSD サービスのデバッグレベルを変更できます。ここで、**<integer>** の値は 0 から 9 の数字になります。デバッグレベルは最大 3 つのログで、最大 3 つのログで、レベル 8 以上では、多くの詳細なログメッセージを提供します。レベル 6 は、認証の問題のデバッグに役立ちます。

前提条件

- **sssctl** コマンドを実行するには、root パスワードが必要です。

手順

- **sssctl debug-level** コマンドを使用して、デバッグレベルを目的の詳細度に設定します。以下に例を示します。

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

関連情報

- SSSD ログファイルおよびログレベル

13.8. IDM サーバーの認証問題をトラブルシューティングするために SSSD サービスからデバッグログを収集する

IdM ユーザーが IdM サーバーへの認証を試行する際に問題が発生した場合は、サーバー上の SSSD サービスで詳細なデバッグロギングを有効にし、ユーザーに関する情報の取得を試行するログを収集します。

前提条件

- **root** 権限がある。

手順

1. IdM サーバーで詳細な SSSD デバッグロギングを有効にします。

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

2. 認証問題が発生しているユーザーの SSSD キャッシュでオブジェクトを無効にするため、LDAP サーバーを省略し、SSSD がすでにキャッシュされている情報を取得しません。

   ```
   [root@server ~]# sssctl cache-expire -u <idm_user>
   ```

3. 古い SSSD ログを削除して、トラブルシューティングのデータセットを最小限に抑える。

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

4. 認証問題が発生し、試行前後にタイムスタンプを収集する際に、ユーザーが認証問題が発生しようと試みます。これらのタイムスタンプは、データセットのスコープをさらに絞り込むことができます。

```
[root@server sssd]# date; su <idm_user>; date
Mon Mar 29 15:33:48 EDT 2021
su: user idm_user does not exist
Mon Mar 29 15:33:49 EDT 2021
```

5. オプション: 詳細な SSSD ログの収集を継続しない場合は、デバッグレベルを下げます。

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

6. 障害のある要求に関する情報を SSSD ログで確認します。たとえば、**/var/log/sssd/sssd_example.com.log** ファイルを確認すると、SSSD サービスが **cn=accounts,dc=example,dc=com** LDAP サブツリーでユーザーを見つけられなかったことを示しています。これは、ユーザーが存在しないか、別の場所に存在することを示しています。

```
(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=idm_user@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=idm_user)(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. 認証問題の原因を判断できない場合は、以下を行います。

   a. 最近生成した SSSD ログを収集します。

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

   b. Red Hat テクニカルサポートケースを作成し、以下を提供します。

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

      ii. ログに対応する要求のタイムスタンプおよびユーザー名を含むコンソールの出力。

      ```
      [root@server sssd]# date; id <idm_user>; date
      Mon Mar 29 15:33:48 EDT 2021
      id: 'idm_user': no such user
      Mon Mar 29 15:33:49 EDT 2021
      ```

13.9. IDM クライアントの認証問題をトラブルシューティングするために SSSD サービスからデバッグログを収集する

IdM クライアントに IdM ユーザーとして認証を試行する際に問題が発生した場合は、IdM サーバーでユーザー情報を取得できることを確認します。IdM サーバーでユーザー情報を取得できない場合は、(IdM サーバーから情報を取得する) IdM クライアントでそれを取得できなくなります。

認証の問題が IdM サーバーから生成されていないことを確認したら、IdM サーバーと IdM クライアントの両方から SSSD デバッグログを収集していました。

前提条件

- IdM サーバーではなく、IdM クライアントで認証の問題のみがあります。
- **sssctl** コマンドを実行して SSSD サービスを再起動するには、root パスワードが必要です。

手順

1. **クライアントで**、テキストエディターで **/etc/sssd/sssd.conf** ファイルを開きます。
2. **クライアントで**、ファイルの **[domain]** セクションに **ipa_server** オプションを追加し、このオプションを完全修飾ドメイン名 (FQDN) を使用して IdM サーバーに設定します。これにより、IdM クライアントは他の IdM サーバーの自動検出を避け、このテストを 1 つのクライアントおよびサーバー 1 台だけに制限します。

```
[domain/<idm_domain_name>]
ipa_server = <idm_server_fqdn>
...
```

3. **クライアントで sssd.conf** ファイルを保存して閉じます。
4. **クライアントで** SSSD サービスを再起動して、設定の変更を読み込みます。

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

5. **サーバーおよびクライアントで**、詳細な SSSD デバッグロギングを有効にします。

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

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

6. **サーバーおよびクライアントで**、認証問題が発生しているユーザーの SSSD キャッシュの検証オブジェクトでは、LDAP データベースを迂回せず、SSSD がすでにキャッシュされています。

```
[root@server ~]# sssctl cache-expire -u <idm_user>
```

```
[root@client ~]# sssctl cache-expire -u <idm_user>
```

7. **サーバーおよびクライアントで**、古い SSSD ログを削除して、トラブルシューティングのデータセットを最小限に抑える。

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

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

8. **クライアントで**、認証問題が発生し、試行前後にタイムスタンプを収集する際に、ユーザーが認証問題が発生しようと試みます。これらのタイムスタンプは、データセットのスコープをさらに絞り込むことができます。

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

9. オプション: **サーバーおよびクライアントで**、詳細な SSSD ログの収集を継続しない場合は、デバッグレベルを下げます。

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

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

10. **サーバーおよびクライアントで**、失敗した要求に関する情報を SSSD ログを確認します。
    a. クライアントログのクライアントからの要求を確認します。
    b. サーバーログのクライアントからの要求を確認します。
    c. サーバーログでリクエストの結果を確認します。
    d. サーバーからリクエストの結果を受信するクライアントの結果を確認します。
11. 認証問題の原因を判断できない場合は、以下を行います。
    a. IdM サーバーおよび IdM クライアントで最近生成した SSSD ログを収集します。ホスト名またはロールに応じてラベルを付けます。

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

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

    b. Red Hat テクニカルサポートケースを作成し、以下を提供します。
        i. SSSD デバッグログ:
            A. サーバーから **sssd-logs-server-Mar29.tar**
            B. クライアントからの **sssd-logs-client-Mar29.tar**
        ii. ログに対応する要求のタイムスタンプおよびユーザー名を含むコンソールの出力。

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

13.10. SSSD バックエンドでのクライアント要求の追跡

SSSD はリクエストを非同期的に処理し、別々の要求のメッセージを同じログファイルに追加します。バックエンドログでクライアント要求を追跡するには、一意の要求識別子 **RID#<integer> とクライアント ID [CID #<integer>]** を使用できます。これらの識別子は、複数の SSSD コンポーネントからのログファイルにわたる個々の要求を最初から最後まで追跡するために、ログを分離するのに役立ちます。

前提条件

- デバッグロギングを有効にし、IdM クライアントから要求が送信されている。
- **root** 権限がある。

手順

1. SSSD のログファイルを確認するために、**less** ユーティリティーを使用してログファイル **/var/log/sssd/sssd_<domain_name>.log** を開きます。以下に例を示します。

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

2. SSSD ログでクライアント要求に関する情報を確認します。

   ```
   (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 ログファイルからのこの出力例は、2 つの異なる要求について一意の識別子の **RID#3** および **RID#4** を示しています。

ただし、SSSD クライアントインターフェイスへの 1 つのクライアント要求が、バックエンドで複数の要求をトリガーすることが多いため、クライアント要求とバックエンドの要求との間に 1 対 1 の相関関係がなくなります。バックエンド内の複数のリクエストには異なる RID 番号がありますが、最初の各バックエンドリクエストには一意のクライアント ID が含まれているため、管理者は単一のクライアントリクエストに対して複数の RID 番号を追跡できます。

以下の例は、1 つのクライアントリクエスト **[sssd.nss CID #1]** と、バックエンドで生成された複数のリクエスト (**[RID#5]** から **[RID#13]**) を示しています。

```
(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. ログ解析ツールを使用したクライアント要求の追跡

System Security Services Daemon (SSSD) には、複数の SSSD コンポーネントからのログファイルにわたり、要求を最初から最後まで追跡するに使用できるログ解析ツールが組み込まれています。

13.11.1. ログ解析ツールの仕組み

ログ解析ツールを使用すると、複数の SSSD コンポーネントからのログファイルにわたり、SSSD の要求を最初から最後まで追跡できます。**sssctl analyze** コマンドを使用してアナライザーツールを実行します。

ログ解析ツールは、SSSD の NSS および PAM の問題をトラブルシューティングし、SSSD デバッグログの確認を容易化するのに役立ちます。SSSD プロセス全体の特定のクライアントリクエストにのみ関連する SSSD ログを抽出して出力できます。

SSSD は、ユーザー認証 (**su**、**ssh**) 情報とは別に、ユーザーおよびグループの ID 情報 (**id**、**getent**) を追跡します。NSS レスポンダのクライアント ID(CID) は PAM レスポンダーの CID とは独立しており、NSS と PAM のリクエストを解析すると重複した数値が表示されます。**--pam** オプションを **sssctl analyze** コマンドとともに使用して、PAM リクエストを確認します。

> **注記**
>
> SSSD メモリーキャッシュから返された要求はログに記録されず、ログ解析ツールで追跡できません。

関連情報

- **sudo sssctl analyze request --help**
- **sudo sssctl analyze --help**
- システム上の **sssd.conf** および **sssctl** man ページ

13.11.2. ログ解析ツールの実行

ログ解析ツールを使用して、SSSD 内のクライアント要求を追跡および分析します。

前提条件

- ログ解析機能を有効にするには、**/etc/sssd/sssd.conf** ファイルの **[$responder]** セクションと **[domain/$domain]** セクションで **debug_level** を 7 以上に設定する必要があります。
- 分析するログは、**libtevent** チェーン ID をサポートする互換性のあるバージョンの SSSD、つまり RHEL 8.5 以降の SSSD からのものである必要がある。

手順

1. ログ解析ツールを **list** モードで実行して、追跡対象の要求のクライアント ID を特定し、**-v** オプションを追加して詳細な出力を表示します。

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

    SSSD に対して行われた最近のクライアント要求の詳細なリストが表示されます。

    > **注記**
    >
    > PAM リクエストを分析する場合は、**sssctl analyze request list** コマンドを **-pam** オプション付きで実行します。

2. **show [unique client ID]** オプションを指定してログ解析ツールを実行し、指定したクライアント ID 番号に関連するログを表示します。

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

3. 必要に応じて、ログファイルに対してログ解析ツールを実行できます。次に例を示します。

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

関連情報

- **sssctl analyze request list --help**
- **sssctl analyze request show --help**
- システム上の **sssctl** man ページ

13.12. 関連情報

- General SSSD Debugging Procedures (Red Hat ナレッジベース)

第14章 シングルサインオン用のアプリケーションの設定

シングルサインオン (SSO) は、1 回のログイン手順で複数のシステムにログインできる認証スキームです。ユーザーの認証手段として、Kerberos チケット、SSL 認定、またはトークンを使用するように、ブラウザーとメールクライアントを設定できます。

アプリケーションによって設定が異なる場合があります。本章では、Mozilla Thunderbird メールクライアントおよび Mozilla Firefox Web ブラウザーに SSO 認証スキーマを設定する方法を例として説明します。

14.1. 前提条件

- 以下のアプリケーションをインストールしている。
  - Mozilla Firefox **バージョン 88**
  - Mozilla Thunderbird **バージョン 78**

14.2. シングルサインオンに KERBEROS を使用するように FIREFOX を設定する

Firefox は、イントラネットサイトやその他の保護された Web サイトへのシングルサインオン (SSO) に Kerberos を使用するように設定できます。これを行うには、最初に、Kerberos 認証情報を適切な鍵配布センター (KDC) に送信するように Firefox を設定する必要があります。

注記

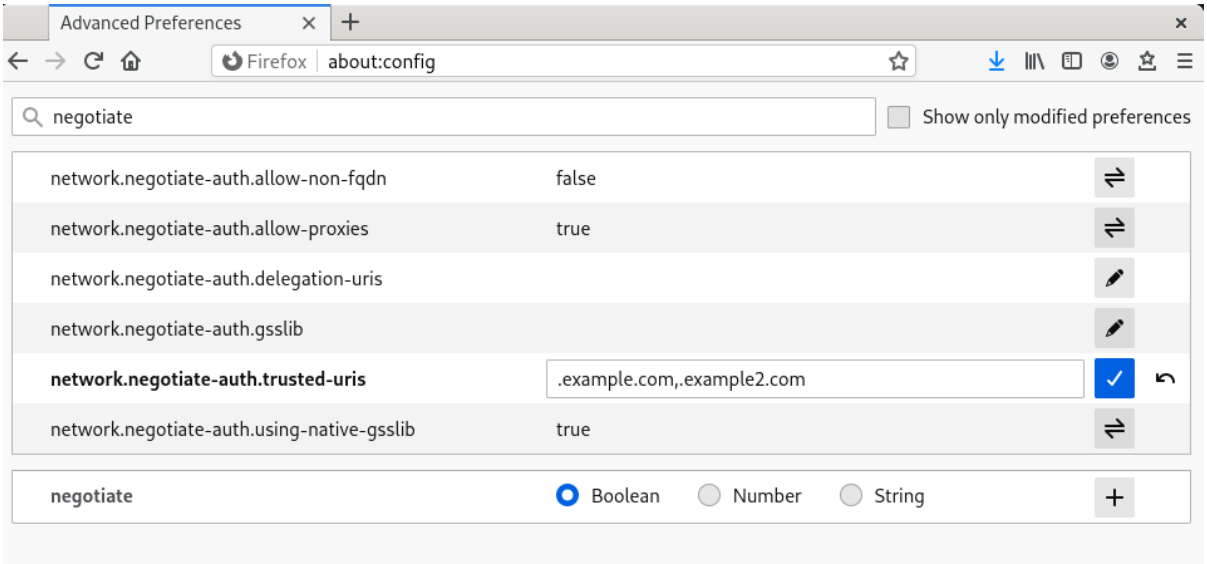
Kerberos 認証情報を渡すように Firefox を設定した後でも、有効な Kerberos チケットが必要です。Kerberos チケットを生成するには、**kinit** コマンドを使用し、KDC 上のユーザーのユーザーパスワードを入力します。

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

手順

1. Firefox のアドレスバーに **about:config** と入力し、現在の設定オプションのリストを表示します。
2. **Filter** フィールドに **negotiate** と入力して、オプションのリストを制限します。
3. **network.negotiate-auth.trusted-uris** エントリーをダブルクリックします。
4. 先行ピリオド (.) を含む、認証に使用するドメイン名を入力します。複数のドメインを追加する場合は、ドメインをコンマ区切りで入力します。

図14.1 Firefox の手動設定



関連情報

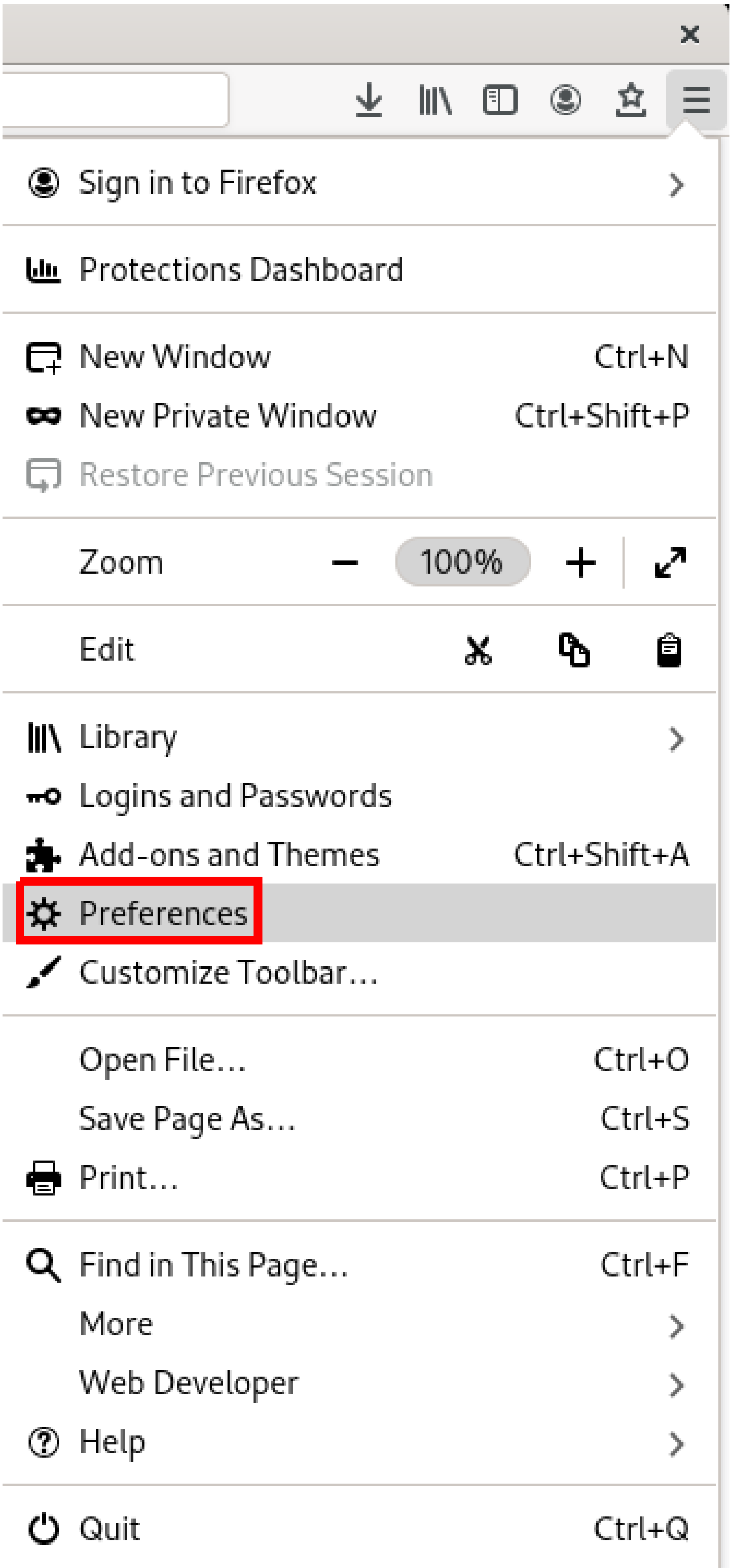
- Kerberos 認証用のブラウザーの設定

14.3. FIREFOX での証明書の表示

Mozilla Firefox に保存されている証明書を表示して、認証設定を確認できます。

手順

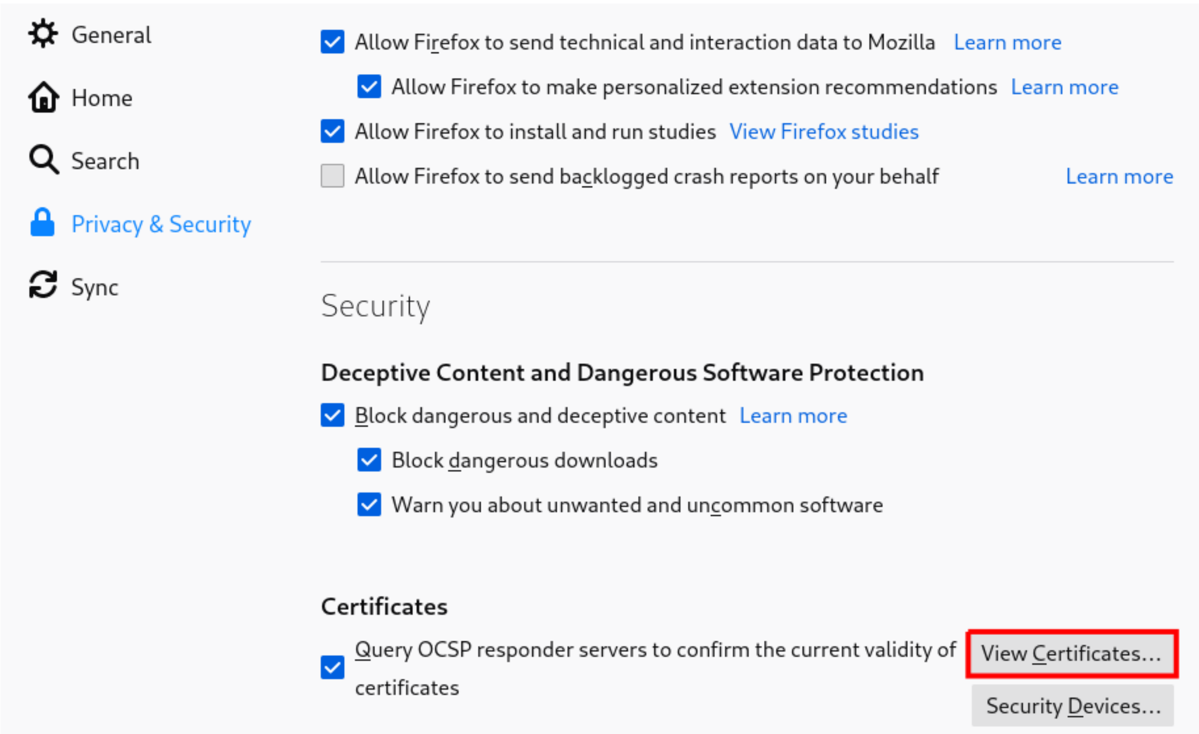
1. Mozilla Firefox で Firefox メニューを開き、**Preferences** を選択します。



2. 左側のパネルで、**Privacy & Security** セクションを選択します。

3. **Certificates** までスクロールします。

4. **View Certificates** をクリックして、**Certificate Manager** を開きます。



14.4. FIREFOX での CA 証明書のインポート

証明書を Mozilla Firefox にインポートして、その証明書をセキュアな接続のために使用する Web サイト、サーバー、またはアプリケーションとの信頼を確立できます。

前提条件

- デバイスに CA 証明書がある。

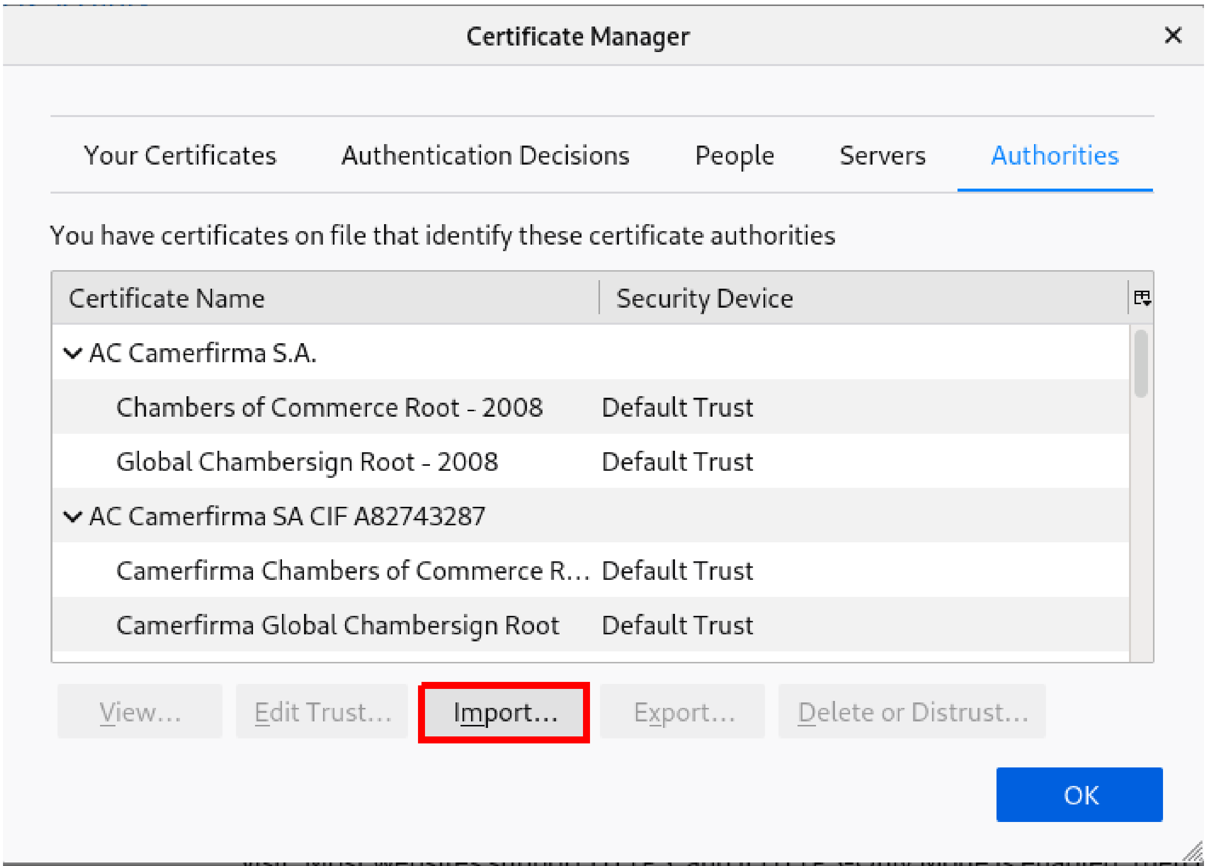
手順

手順

1. **Certificate Manager** を開きます。

2. **Authorities** タブを選択し、**Import** をクリックします。

図14.2 Firefox での CA 証明書のインポート



3. デバイスからダウンロードした CA 証明書を選択します。

14.5. FIREFOX での証明書信頼設定の編集

Mozilla Firefox が証明書を信頼する方法を変更できます。

前提条件

1. 証明書が正常にインポートされました。

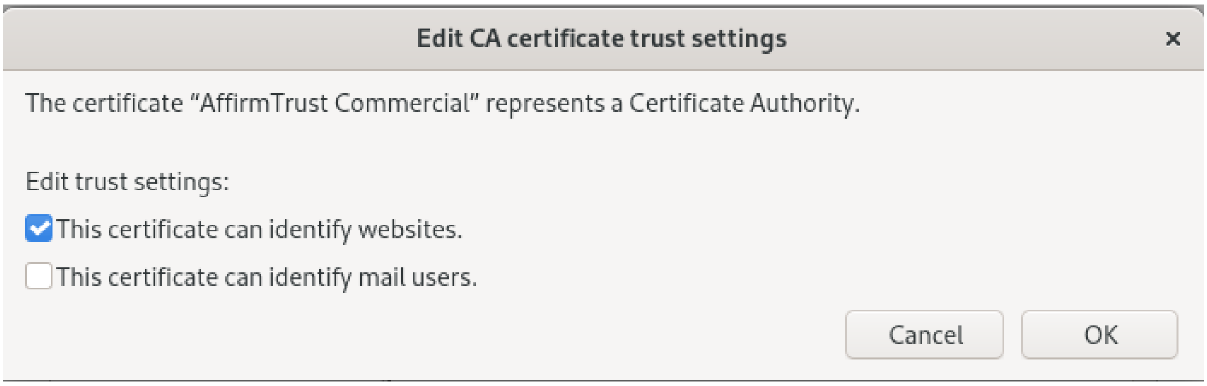
手順

1. **Certificate Manager** を開きます。

2. **Authorities** タブで、適切な証明書を選択し、**Edit Trust** をクリックします。

3. 証明書信頼設定を編集します。

図14.3 Firefox での証明書信頼設定の編集



14.6. FIREFOX での認証用個人証明書のインポート

Web サイトやサービスへの認証用の個人証明書をインポートできます。

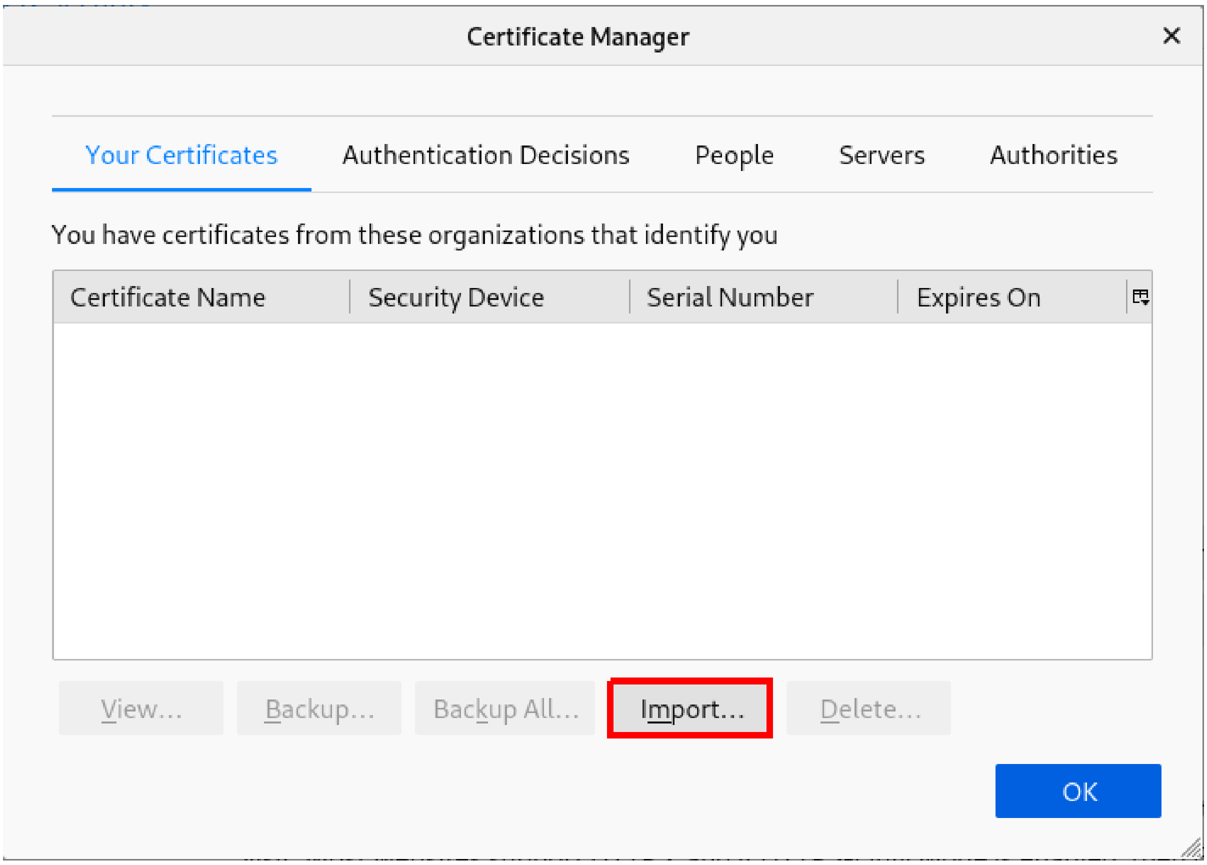
前提条件

1. デバイスに個人証明書が保存されている。

手順

1. **Certificate Manager** を開きます。
2. **Your Certificates** タブを選択し、**Import** をクリックします。

図14.4 Firefox での認証用個人証明書のインポート



3. コンピューターから適切な証明書をインポートします。

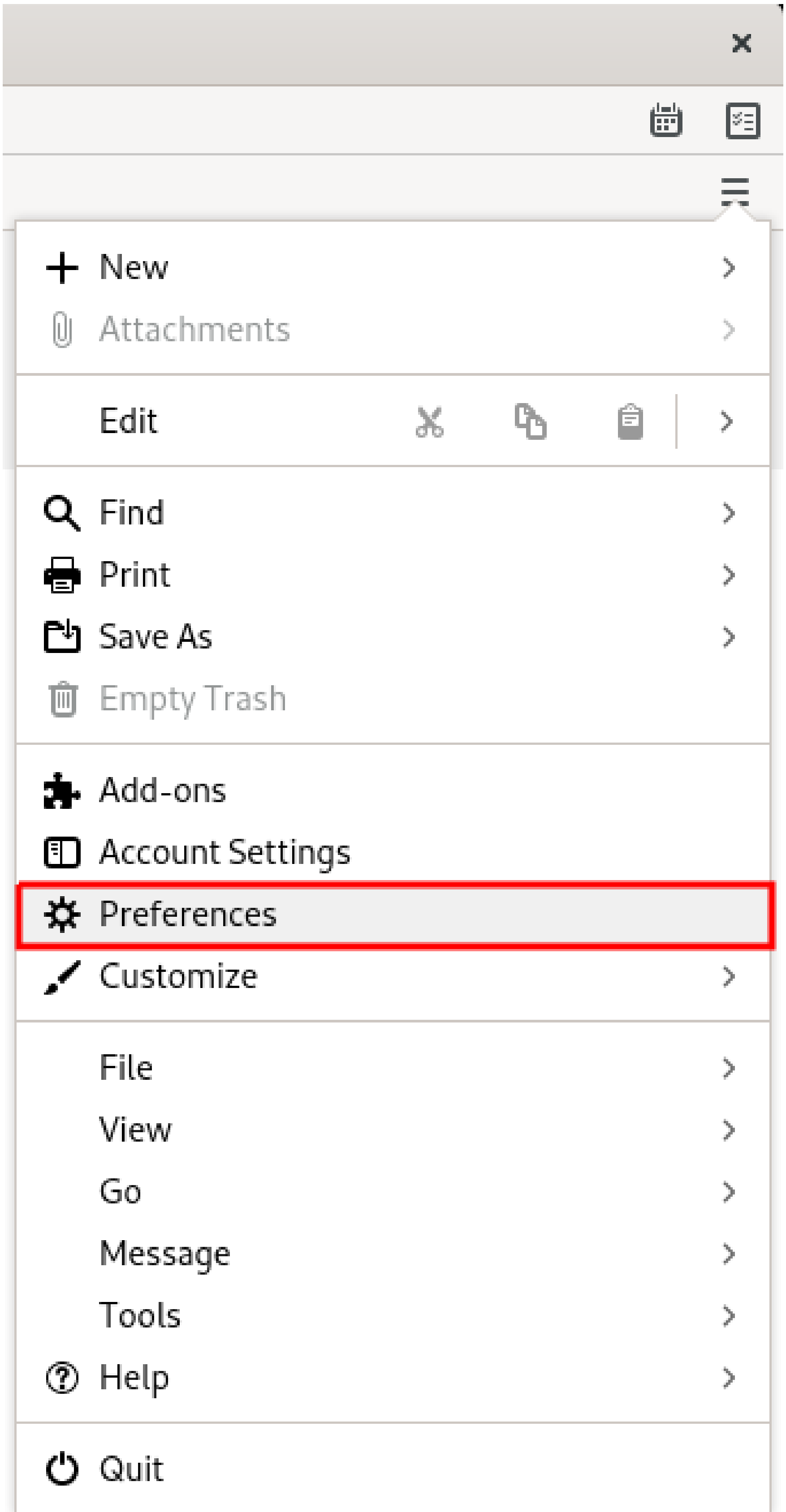
14.7. THUNDERBIRD での証明書の表示

Mozilla Thunderbird で証明書を表示して、メールクライアントのセキュリティー設定を管理できます。

手順

1. Thunderbird でメインメニューを開き、**Preferences** を選択します。

図14.5 メニューから Preferences を選択する



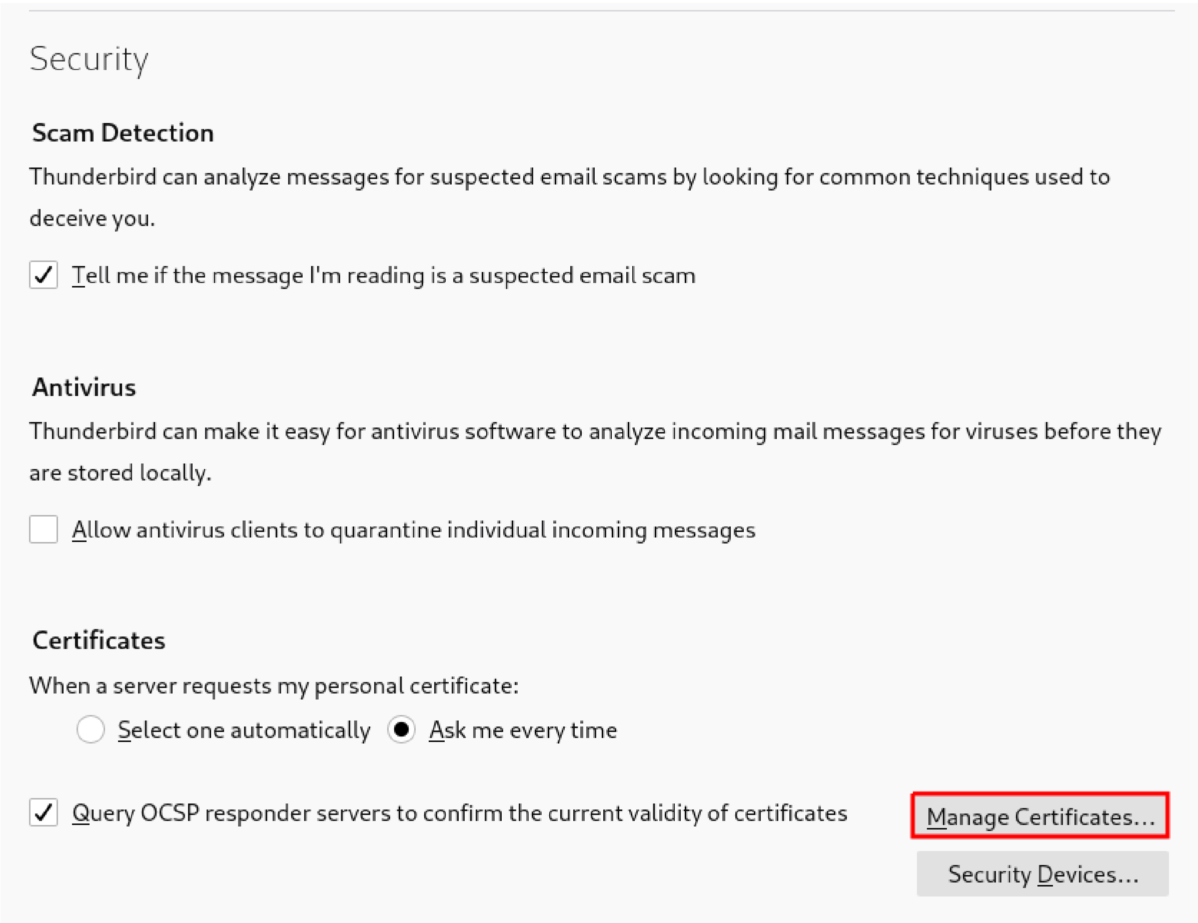
2. 左側のパネルで、**Privacy & Security** セクションを選択します。

図14.6 セキュリティーセクションの選択

3. **Certificates** までスクロールします。

4. **Manage Certificates** をクリックして、**Certificate Manager** を開きます。

図14.7 Certificate Manager の起動



14.8. THUNDERBIRD での証明書のインポート

Mozilla Thunderbird メールクライアントに証明書をインポートできます。

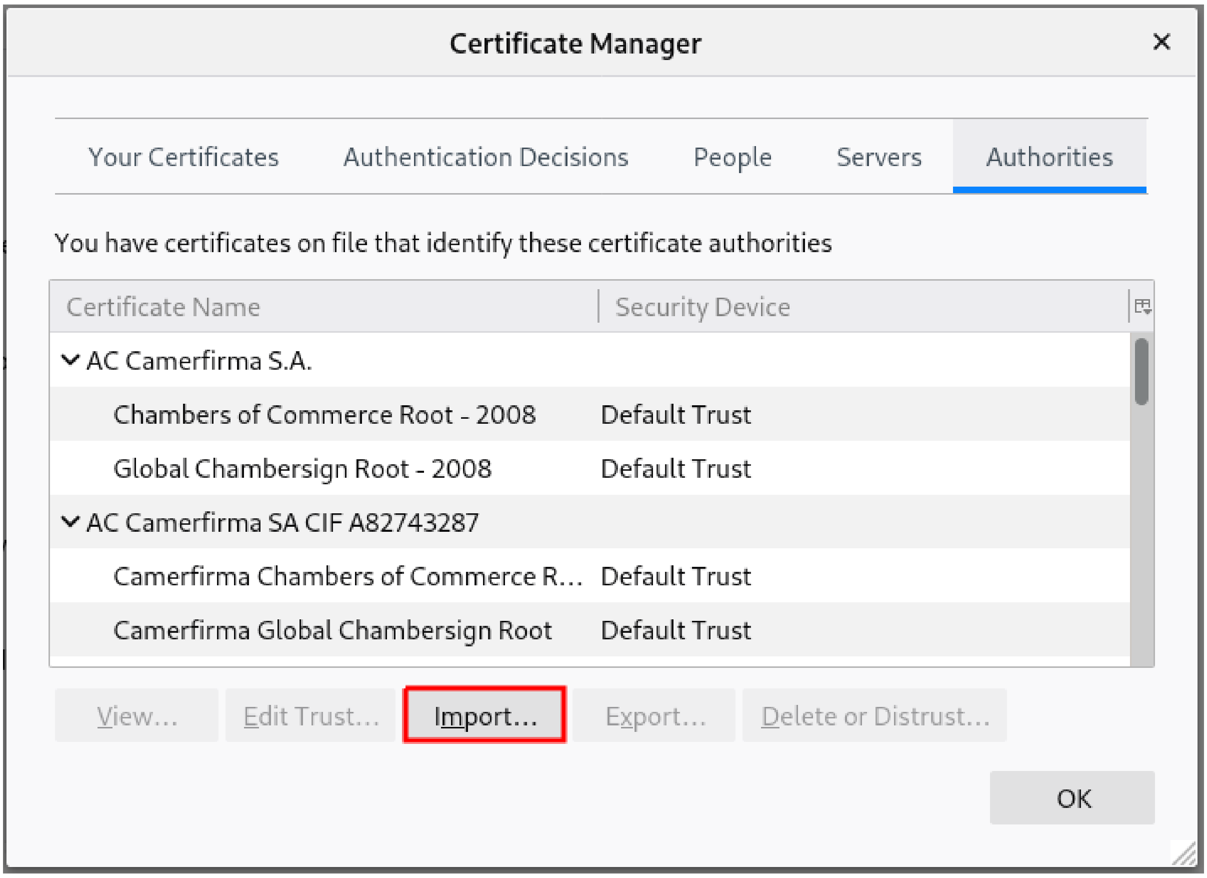
前提条件

- デバイスに CA 証明書が保存されている。

手順

1. **Certificate Manager** を開きます。
2. **Authorities** タブを選択し、**Import** をクリックします。

図14.8 Thunderbird で CA 証明書のインポート



3. ダウンロードした CA 証明書を選択します。

14.9. THUNDERBIRD での証明書信頼設定の編集

Mozilla Thunderbird メールクライアントで証明書信頼設定を編集できます。

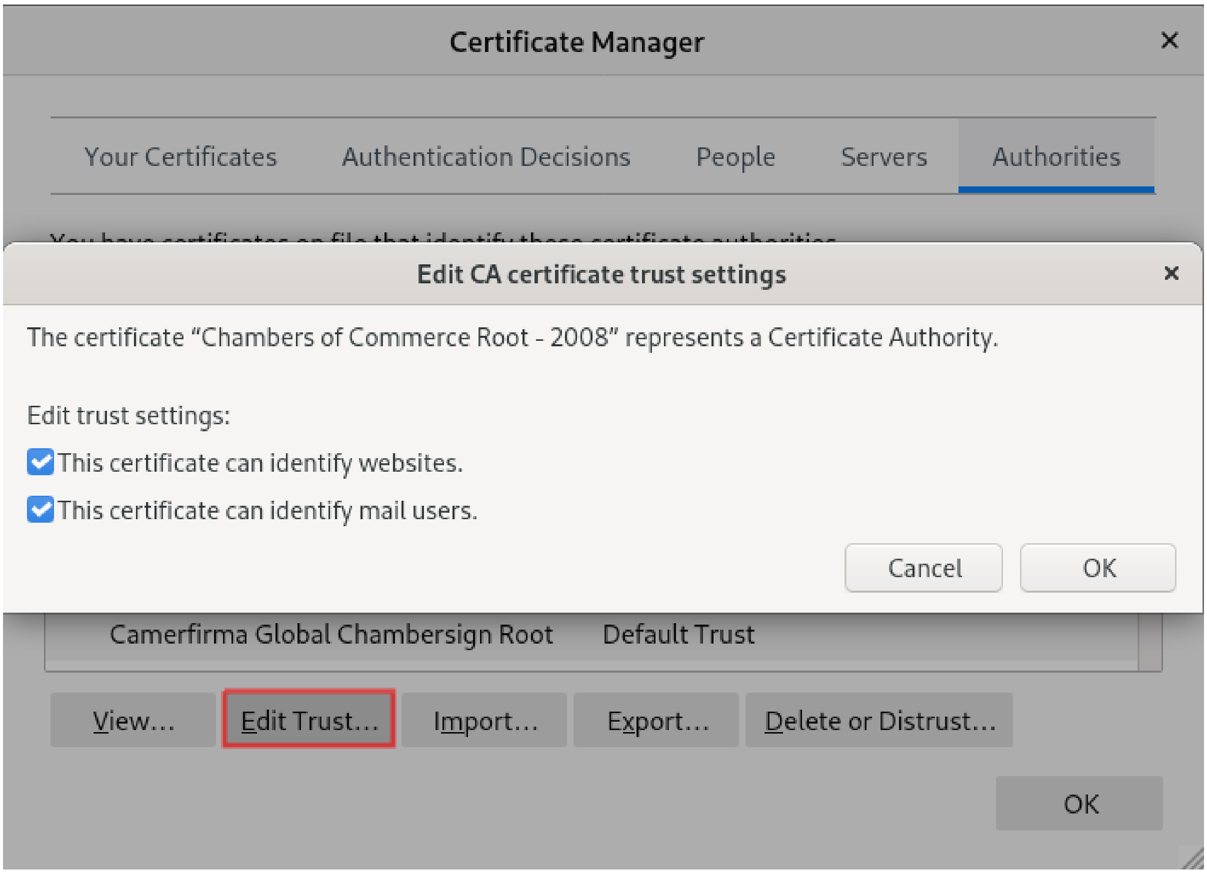
前提条件

- 証明書が正常にインポートされました。

手順

1. **Certificate Manager** を開きます。
2. **Authorities** タブで、適切な証明書を選択し、**Edit Trust** をクリックします。
3. 証明書信頼設定を編集します。

図14.9 Thunderbird で証明書の信頼設定の編集



14.10. THUNDERBIRD での個人証明書のインポート

Mozilla Thunderbird メールクライアントに個人認証用の証明書をインポートできます。

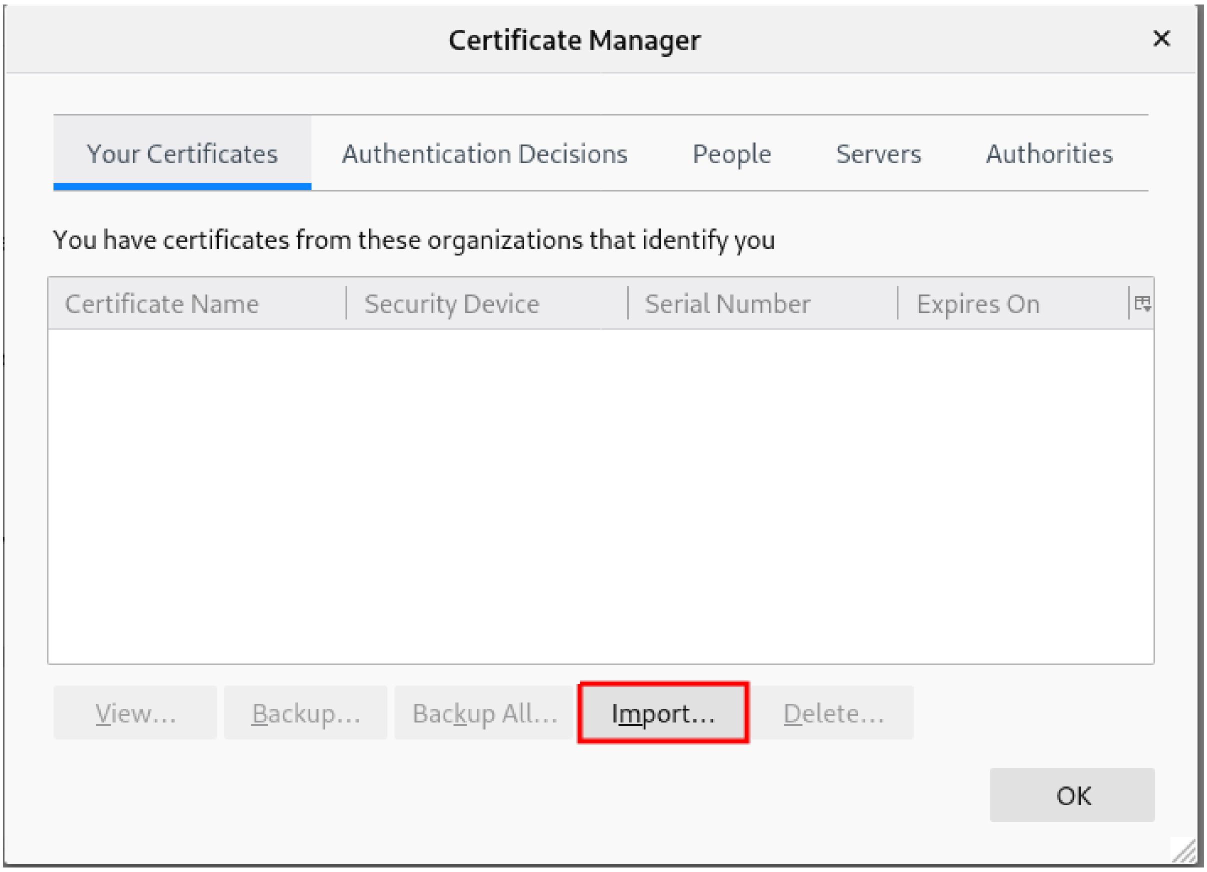
前提条件

1. デバイスに個人証明書が保存されている。

手順

1. **Certificate Manager** を開きます。
2. **Your Certificates** タブで、**Import** をクリックします。

図14.10 Thunderbird での認証用個人証明書のインポート

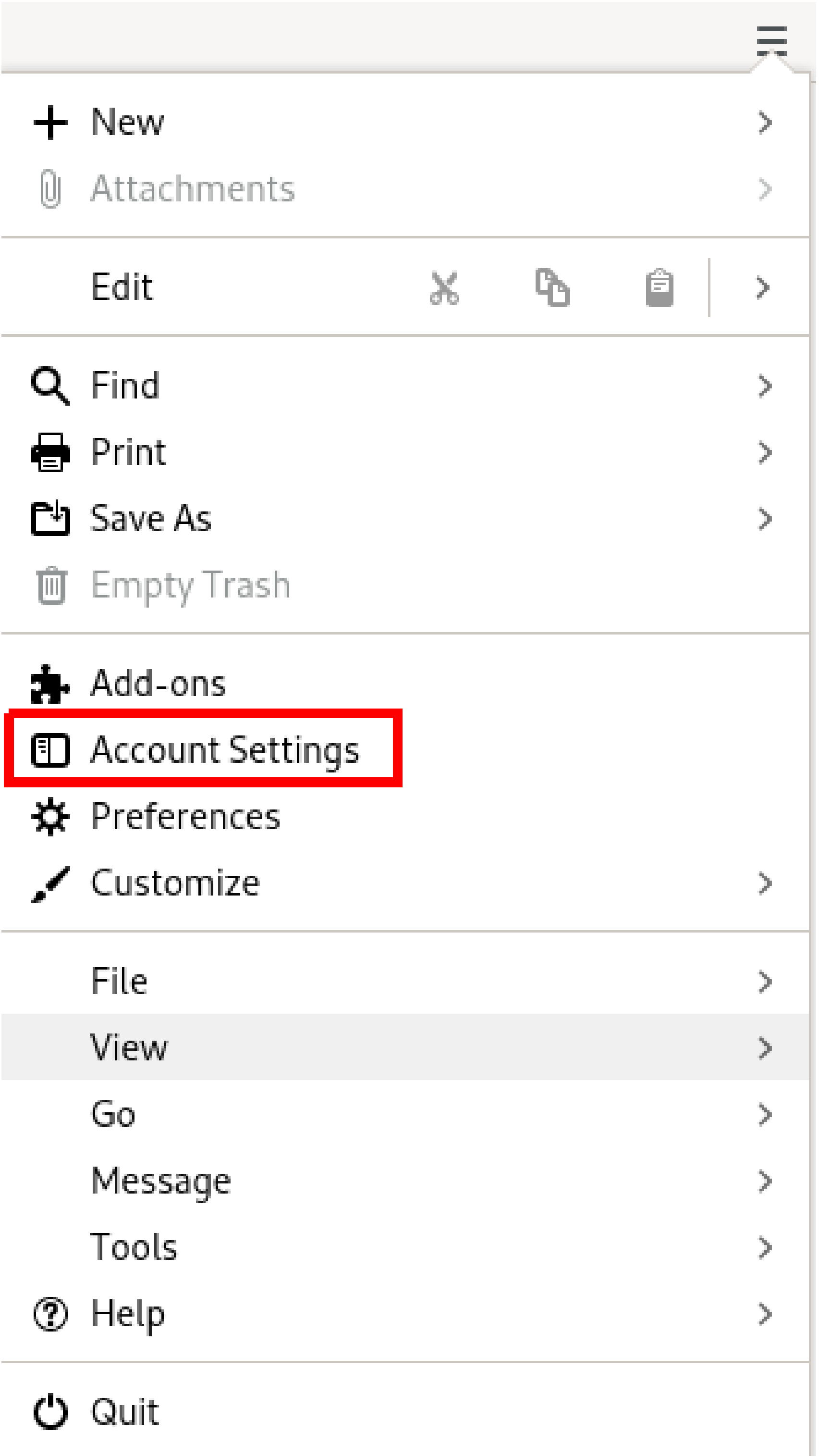


3. 必要な証明書をコンピューターからインポートします。

4. **Certificate Manager** を閉じます。

5. メインメニューを開き、**Account Settings** を選択します。

図14.11 メニューからアカウント設定を選択する



6. アカウントのメールアドレスの下の左側のパネルで **End-To-End Encryption** を選択します。
   End-To-End Encryption セクションの選択

7. **S/MIME** セクションで、1 つ目の **Select** ボタンをクリックして、メッセージの署名に使用する個人証明書を選択します。

8. **S/MIME** セクションで、2 つ目の **Select** ボタンをクリックして、メッセージの暗号化と復号に使用する個人証明書を選択します。
署名および暗号化/復号に使用する証明書の選択

注記

有効な証明書をインポートし忘れた場合は、**Manage S/MIME certificate** オプションを使用すると、**Certificate Manager** を直接開くことができます。