



## Red Hat Enterprise Linux 8

# RHEL システムと Windows Active Directory を 直接統合

Active Directory と直接接続する RHEL システムの理解および設定



# Red Hat Enterprise Linux 8 RHEL システムと Windows Active Directory を直接統合

---

Active Directory と直接接続する RHEL システムの理解および設定

## 法律上の通知

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

本書は、SSSD を使用して RHEL システムを Windows Active Directory に直接統合する方法を説明します。

## 目次

多様性を受け入れるオープンソースの強化 .....	3
RED HAT ドキュメントへのフィードバック (英語のみ) .....	4
<b>第1章 SSSD を使用して RHEL システムを AD に直接接続 .....</b>	<b>5</b>
1.1. SSSD を使用した直接統合の概要	5
1.2. 直接統合でサポートされる WINDOWS プラットフォーム	6
1.3. AD および RHEL で一般的な暗号化タイプに対応	6
1.4. AD に直接接続	7
1.4.1. SSSD を使用した AD ドメインの検出および参加	7
1.4.2. AD との統合オプション: ID マッピングまたは POSIX 属性の使用	9
1.4.2.1. AD ユーザー用に新規 UID および GID を自動的に生成	10
1.4.2.2. AD で定義されている POSIX 属性の使用	10
1.4.3. Active Directory で定義された POSIX 属性を使用した AD への接続	10
1.4.4. SSSD を使用したさまざまな AD フォレストでの複数ドメインへの接続	12
1.5. AD プロバイダーが動的 DNS 更新を処理する方法	15
1.6. AD プロバイダーの動的 DNS 設定の変更	16
1.7. AD プロバイダーが信頼されるドメインを処理する方法	17
1.8. REALM コマンド	17
<b>第2章 SAMBA WINBIND を使用して RHEL システムを AD に直接接続 .....</b>	<b>19</b>
2.1. SAMBA WINBIND を使用した直接統合の概要	19
2.2. 直接統合でサポートされる WINDOWS プラットフォーム	19
2.3. AD および RHEL で一般的な暗号化タイプに対応	20
2.4. RHEL システムの AD ドメインへの参加	21
2.5. REALM コマンド	23
<b>第3章 AD への直接接続の管理 .....</b>	<b>25</b>
3.1. デフォルトの KERBEROS ホストのキータブの更新間隔を変更	25
3.2. AD ドメインからの RHEL システムの削除	25
3.3. SSSD でドメイン解決順序を設定して、AD ユーザーの短縮名を解決する手順	26
3.4. ドメインユーザーのログインパーミッションの管理	27
3.4.1. ドメイン内でユーザーのアクセス権を有効化	27
3.4.2. ドメイン内でユーザーのアクセス権を拒否	29
3.5. RHEL でのグループポリシーアクセス制御の適用	30
3.5.1. SSSD が GPO アクセス制御ルールを解釈する方法	30
3.5.1.1. ホストによるフィルタリングの制限	30
3.5.1.2. グループ別フィルタリングの制限	31
3.5.2. SSSD がサポートする GPO 設定の一覧	31
3.5.3. GPO 強制を制御する SSSD オプションの一覧	31
3.5.3.1. ad_gpo_access_control オプション	31
3.5.3.2. ad_gpo_implicit_deny オプション	32
3.5.4. GPO アクセス制御モードの変更	33
3.5.5. AD GUI での RHEL ホストの GPO の作成および設定	35
3.5.6. 関連情報	36



## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[弊社](#) の CTO、Chris Wright の [メッセージ](#) を参照してください。

Identity Management では、以下のような用語の置き換えが含まれます。

- **ブラックリストからブロックリスト**
- **ホワイトリストから許可リスト**
- **スレーブからセカンダリー**
- **単語 マスター は、コンテキストに応じて、より正確な言語に置き換えられます。**
  - **マスターからIdM サーバー**
  - **CA 更新マスター から CA 更新サーバー**
  - **CRL マスターからCRL パブリッシャーサーバー**
  - **マルチマスターからマルチサプライヤー**

## RED HAT ドキュメントへのフィードバック (英語のみ)

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。改善点を報告する場合は、以下のように行います。

- 特定の文章に簡単なコメントを記入する場合は、以下の手順を行います。
  1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上端に **Feedback** ボタンがあることを確認してください。
  2. マウスカーソルで、コメントを追加する部分を強調表示します。
  3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
  4. 表示される手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla のチケットを作成します。
  1. [Bugzilla](#) の Web サイトにアクセスします。
  2. Component で **Documentation** を選択します。
  3. **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記入してください。
  4. **Submit Bug** をクリックします。



## 第1章 SSSD を使用して RHEL システムを AD に直接接続

本セクションでは、SSSD (System Security Services Daemon) を使用して、RHEL システムを Active Directory (AD) に接続する方法を説明します。RHEL システムを Active Directory (AD) に接続するには、2つのコンポーネントが必要です。1つ目のコンポーネントは SSSD と呼ばれ、中央の ID および認証ソースと相互作用します。2つ目のコンポーネントは **realmd** と呼ばれ、利用可能なドメインを検出し、SSSD がドメインに接続するように基盤となる RHEL システムサービスを設定します。

- [SSSD を使用した直接統合の概要](#)
- [直接統合でサポートされる Windows プラットフォーム](#)
- [AD および RHEL で一般的な暗号化タイプに対応](#)
- [AD に直接接続](#)
- [AD プロバイダーが動的 DNS 更新を処理する方法](#)
- [AD プロバイダーの動的 DNS 設定の変更](#)
- [AD プロバイダーが信頼されるドメインを処理する方法](#)
- [realm コマンド](#)

### 1.1. SSSD を使用した直接統合の概要

オフラインのログインを許可するために、SSSD を使用して、ユーザーキャッシュを備えた共通のフレームワークを介して、ユーザーディレクトリーにアクセスして認証および認可を行います。SSSD は高度な設定が可能で、PAM (Pluggable Authentication Module) と NSS (Name Switch Service) の統合と、ローカルユーザーと、中央サーバーから取得した拡張ユーザーデータを保存するデータベースを提供します。SSSD は、RHEL システムを以下のいずれかの ID サーバーに接続するのに推奨されるコンポーネントです。

- Active Directory
- RHEL の ID 管理 (IdM)
- あらゆる汎用 LDAP または Kerberos サーバー



#### 注記

SSSD との直接統合は、デフォルトで1つの AD フォレスト内でのみ機能します。

SSSD が AD と Linux システムを直接統合するように設定する最も便利な方法は、**realmd** サービスを使用することです。これにより、呼び出し元はネットワーク認証およびドメインメンバーシップを標準的な方法で設定できます。**realmd** サービスは、アクセス可能なドメインおよびレルムに関する情報を自動的に検出し、ドメインまたはレルムに参加するのに高度な設定を必要としません。

SSSD は、AD との直接統合および間接統合の両方に使用でき、ある統合アプローチから別の統合アプローチに切り替えることができます。直接統合は、RHEL システムを AD 環境に導入する簡単な方法です。ただし、RHEL システムの共有が増えると、デプロイメントは通常、ホストベースのアクセス制御、sudo、SELinux ユーザーマッピングなどの ID 関連のポリシーをより集中管理する必要があります。最初に、ローカル設定ファイルで、RHEL システムのこのような設定を維持できます。ただし、システムの数が増えると、Red Hat Satellite などのプロビジョニングシステムでは、設定ファイルの配布と管理が簡単になります。直接統合がスケールされない場合は、間接統合を検討する必要があります。

す。直接統合 (RHEL クライアントは AD ドメインにあります) から間接統合 (AD と信頼関係がある IdM) への移行の詳細は、[「Moving RHEL clients from AD domain to IdM Server」](#) を参照してください。

ユースケースに適した統合のタイプは、[「間接統合と直接統合のどちらを選択すべきか」](#) を参照してください。

## 関連情報

- man ページの **realm(8)**
- man ページの **sssd-ad(5)**
- man ページの **sssd(8)**

## 1.2. 直接統合でサポートされる WINDOWS プラットフォーム

RHEL システムと、以下のフォレストおよびドメインの機能レベルを使用する Active Directory フォレストを直接統合できます。

- フォレスト機能レベルの範囲 - Windows Server 2008 ~ Windows Server 2016
- ドメイン機能レベルの範囲 - Windows Server 2008 ~ Windows Server 2016

直接統合は、以下のサポート対象のオペレーティングシステムでテストされています。

- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2



### 注記

Windows Server 2019 では、新しい機能レベルが導入されていません。Windows Server 2019 が使用する機能レベルで最も高いものは Windows Server 2016 です。

## 1.3. AD および RHEL で一般的な暗号化タイプに対応

デフォルトでは、SSSD は RC4、AES-128、および AES-256 の Kerberos 暗号化タイプに対応します。

RC4 暗号化は、新しい暗号化タイプ AES-128 および AES-256 よりも安全ではないと見なされるため、RHEL 8 ではデフォルトで非推奨となり、無効にされています。一方、Active Directory (AD) ユーザーの認証情報と AD ドメイン間の信頼は RC4 暗号化をサポートしており、AES 暗号化タイプに対応していない可能性があります。

一般的な暗号化タイプがないと、RHEL ホストと AD ドメイン間の通信が機能しないか、一部の AD アカウントが認証できない可能性があります。この状況を修正するには、以下の設定のいずれかを変更します。

- **Active Directory で AES 暗号化サポートを有効にする (推奨オプション)** - AD フォレストの AD ドメイン間で信頼されるようにするには、Microsoft の記事 [「AD DS: Security: Kerberos "Unsupported etype" error when accessing a resource in a trusted domain」](#) を参照してください。

- **RC4 サポートを RHEL で有効化**: AD ドメインコントローラーに対する認証が行われるすべての RHEL ホストで次のコマンドを実行します。
  1. **update-crypto-policies** コマンドを使用して、**DEFAULT** 暗号化ポリシーに加え **AD-SUPPORT** 暗号化サブポリシーを有効にします。

```
[root@host ~]# update-crypto-policies --set DEFAULT:AD-SUPPORT
Setting system policy to DEFAULT:AD-SUPPORT
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies
to fully take place.
```

2. ホストを再起動します。

## 重要

**AD-SUPPORT** 暗号化サブポリシーは、RHEL 8.3 以降でのみ利用できます。

- RHEL 8.2 以前は RC4 のサポートを有効にするには、**cipher = RC4-128+** でカスタム暗号化モジュールポリシーを作成および有効にします。詳細は、[ポリシー修飾子を使用したシステム全体の暗号化ポリシーのカスタマイズ](#)を参照してください。
- RHEL 8.0 および RHEL 8.1 で RC4 のサポートを有効にするには、**/etc/crypto-policies/back-ends/krb5.config** ファイルの **permitted\_encyptypes** オプションに **+rc4** を追加します。

```
[[libdefaults]
permitted_encyptypes = aes256-cts-hmac-sha1-96 aes256-cts-hmac-sha384-
192 camellia256-cts-cmac aes128-cts-hmac-sha1-96 aes128-cts-hmac-
sha256-128 camellia128-cts-cmac +rc4
```

## 関連情報

- RHEL 暗号化ポリシーの使用方法は、『セキュリティ強化』ガイドの「[システム全体の暗号化ポリシーの使用](#)」を参照してください。

## 1.4. AD に直接接続

本セクションでは、ID マッピングまたは POSIX 属性のいずれかを使用して AD と直接統合する方法を説明します。

- [SSSD を使用した AD ドメインの検出および参加](#)
- [AD との統合オプション: ID マッピングまたは POSIX 属性の使用](#)
- [Active Directory で定義された POSIX 属性を使用した AD への接続](#)
- [SSSD を使用したさまざまな AD フォレストでの複数ドメインへの接続](#)

### 1.4.1. SSSD を使用した AD ドメインの検出および参加

この手順では、AD ドメインを検出し、SSSD を使用してこのドメインに RHEL システムを接続する方法を説明します。

## 前提条件

- RHEL ホストの以下のポートが開き、AD ドメインコントローラーからアクセスできることを確認します。

表1.1 SSSD を使用した Linux システムの AD への直接統合に必要なポート

サービス	ポート	プロトコル	備考
DNS	53	UDP および TCP	
LDAP	389	UDP および TCP	
Kerberos	88	UDP および TCP	
Kerberos	464	UDP および TCP	パスワードを設定または変更するために、kadmin により使用されます。
LDAP グローバルカタログ	3268	TCP	<b>id_provider = ad</b> オプションが使用されている場合
NTP	123	UDP	任意

- DNS に AD ドメインコントローラーサーバーが使用されていることを確認します。
- 両方のシステムのシステム時刻が同期していることを確認します。これにより、Kerberos が正常に機能できるようになります。

## 手順

1. 以下のパッケージをインストールします。

```
# yum install samba-common-tools realmd oddjob oddjob-mkhomedir sssd adcli krb5-workstation
```

2. 特定のドメインの情報を表示するには、**realm detect** を実行して、検出するドメイン名を追加します。

```
# realm discover ad.example.com
ad.example.com
type: kerberos
realm-name: AD.EXAMPLE.COM
domain-name: ad.example.com
configured: no
server-software: active-directory
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
```

```
required-package: sssd
required-package: adcli
required-package: samba-common
```

**realmd** システムは DNS SRV ルックアップを使用して、このドメイン内のドメインコントローラーを自動検索します。



### 注記

**realmd** システムは、Active Directory ドメインと Identity Management ドメインの両方を検出できます。両方のドメインが環境に存在する場合は、特定タイプのサーバーに検出結果を絞り込むには **--server-software=active-directory** オプションを使用します。

3. **realm join** コマンドを使用して、ローカルの RHEL システムを設定します。**realmd** スイートは、必要なすべての設定ファイルを自動的に編集します。たとえば、**ad.example.com** ドメインの場合は、次のコマンドを実行します。

```
# realm join ad.example.com
```

### 検証手順

- 管理者ユーザーなど、AD ユーザーの詳細を表示します。

```
# getent passwd administrator@ad.example.com
administrator@ad.example.com:*:1450400500:1450400513:Administrator:/home/administrator
@ad.example.com:/bin/bash
```

### 関連情報

- man ページの **realm(8)** を参照してください。
- man ページの **nmcli(1)** を参照してください。

## 1.4.2. AD との統合オプション: ID マッピングまたは POSIX 属性の使用

Linux システムおよび Windows システムは、ユーザーおよびグループに異なる識別子を使用します。

- Linux では、**ユーザー ID (UID)** と **グループ ID (GID)** が使用されます。『[基本的なシステム設定の構成](#)』の「[ユーザーアカウントおよびグループアカウントの管理](#)」を参照してください。Linux の UID および GID は、POSIX 標準に準拠します。
- Windows は、**セキュリティー ID (SID)** を使用します。



### 重要

RHEL システムを AD に接続すると、AD のユーザー名とパスワードを使用して認証できません。名前の重複が原因で競合が生じ、認証プロセスが中断される可能性があるため、Windows ユーザーと同じ名前の Linux ユーザーを作成しないでください。

RHEL システムに対して AD ユーザーとして認証するには、UID と GID が割り当てられている必要があります。SSSD は、ID マッピングまたは POSIX 属性のいずれかを使用して AD と統合するオプションを提供します。デフォルトでは、ID マッピングを使用します。

### 1.4.2.1. AD ユーザー用に新規 UID および GID を自動的に生成

SSSD は、AD ユーザーの SID を使用して、ID マッピングと呼ばれるプロセスにおいてアルゴリズムで POSIX ID を生成できます。ID マッピングは、AD の SID と Linux の ID との間にマップを作成します。

- SSSD が新しい AD ドメインを検出すると、利用可能な ID の範囲を新しいドメインに割り当てます。
- AD ユーザーが SSSD クライアントマシンに初めてログインすると、SSSD は、ユーザーの SID およびそのドメインの ID 範囲を基にした UID など、SSSD キャッシュにユーザーのエントリーを作成します。
- AD ユーザーの ID は、同じ SID から一貫した方法で生成されるため、Red Hat Enterprise Linux システムにログインする場合は、そのユーザーに同じ UID と GID が使用されます。

「[SSSD を使用した AD ドメインの検出および参加](#)」を参照してください。



#### 注記

全クライアントシステムが SSSD を使用して SID を Linux ID にマッピングすると、マッピングは一貫性を維持します。一部のクライアントが別のソフトウェアを使用する場合は、以下のいずれかを選択します。

- すべてのクライアントで同じマッピングアルゴリズムが使用されていることを確認します。
- AD に定義されている明示的な POSIX 属性を使用します。

### 1.4.2.2. AD で定義されている POSIX 属性の使用

AD は、**uidNumber**、**gidNumber**、**unixHomeDirectory**、**loginShell** などの POSIX 属性を作成して保存できます。

上記の ID マッピングを使用すると、SSSD は新しい UID と GID を作成し、AD で定義された値を上書きします。AD 定義の値を維持するには、SSSD で ID マッピングを無効にする必要があります。

「[Active Directory で定義された POSIX 属性を使用した AD への接続](#)」を参照してください。

### 1.4.3. Active Directory で定義された POSIX 属性を使用した AD への接続

最適なパフォーマンスを得るには、POSIX 属性を AD グローバルカタログに公開します。POSIX 属性がグローバルカタログにない場合、SSSD は LDAP ポート上の個々のドメインコントローラーに直接接続します。

#### 前提条件

- RHEL ホストの以下のポートが開き、AD ドメインコントローラーからアクセスできることを確認します。

表1.2 SSSD を使用した Linux システムの AD への直接統合に必要なポート

サービス	ポート	プロトコル	備考
DNS	53	UDP および TCP	

サービス	ポート	プロトコル	備考
LDAP	389	UDP および TCP	
Kerberos	88	UDP および TCP	
Kerberos	464	UDP および TCP	パスワードを設定または変更するために、kadmin により使用されます。
LDAP グローバルカタログ	3268	TCP	<b>id_provider = ad</b> オプションが使用されている場合
NTP	123	UDP	任意

- DNS に AD ドメインコントローラーサーバーが使用されていることを確認します。
- 両方のシステムのシステム時刻が同期していることを確認します。これにより、Kerberos が正常に機能できるようになります。

## 手順

1. 以下のパッケージをインストールします。

```
# yum install realmd oddjob oddjob-mkhomedir sssd adcli krb5-workstation
```

2. **realm join** コマンドに **--automatic-id-mapping=no** オプションを付けて実行して、ローカルの RHEL システムで ID マッピングを無効にします。**realmd** スイートは、必要な設定ファイルをすべて自動的に編集します。たとえば、**ad.example.com** ドメインの場合は、次のコマンドを実行します。

```
# realm join --automatic-id-mapping=no ad.example.com
```

3. ドメインに参加している場合は、SSSD で ID マッピングを手動で無効にできます。
  - a. **/etc/sss/sss.conf** ファイルを開きます。
  - b. AD ドメインセクションで、**ldap\_id\_mapping = false** 設定を追加します。
  - c. SSSD キャッシュを削除します。

```
rm -f /var/lib/sss/db/*
```

- d. SSSD を再起動します。

```
systemctl restart sssd
```

SSSD は、ローカルで作成するのではなく、AD の POSIX 属性を使用するようになりました。



## 注記

AD のユーザーに関連する POSIX 属性 (**uidNumber**、**gidNumber**、**unixHomeDirectory**、および **loginShell**) を設定する必要があります。

## 検証手順

- 管理者ユーザーなど、AD ユーザーの詳細を表示します。

```
# getent passwd administrator@ad.example.com
administrator@ad.example.com:*:10000:10000:Administrator:/home/Administrator:/bin/bash
```

## 関連情報

- ID マッピングおよび **ldap\_id\_mapping** パラメーターの詳細は、man ページの **sssd-ldap(8)** を参照してください。

### 1.4.4. SSSD を使用したさまざまな AD フォレストでの複数ドメインへの接続

この手順では、異なるフォレストで複数の Active Directory (AD) ドメインに参加して認証する方法を説明します。このドメインには信頼関係がありません。

この例では、**addomain1.com** と **addomain2.com** の2つのドメインに参加させる方法を説明します。**realmd** を使用して最初のドメインに参加し、そのドメインの SSSD、Kerberos、およびその他のユーティリティを自動的に設定します。**adcli** を使用して追加のドメインに参加し、手動で設定ファイルにそのドメインを追加します。

## 前提条件

- RHEL ホストの以下のポートが開き、AD ドメインコントローラーからアクセスできることを確認します。

表1.3 SSSD を使用した Linux システムの AD への直接統合に必要なポート

サービス	ポート	プロトコル	備考
DNS	53	UDP および TCP	
LDAP	389	UDP および TCP	
Kerberos	88	UDP および TCP	
Kerberos	464	UDP および TCP	パスワードを設定または変更するためには、 <b>kadmin</b> により使用されます。
LDAP グローバルカタログ	3268	TCP	<b>id_provider = ad</b> オプションが使用されている場合



サービス	ポート	プロトコル	備考
NTP	123	UDP	任意

- DNS に AD ドメインコントローラーサーバーが使用されていることを確認します。
- 両方のシステムのシステム時刻が同期していることを確認します。これにより、Kerberos が正常に機能できるようになります。
- マシンをそのドメインに参加させる権限を持つ各 AD ドメインに、AD 管理者アカウントの認証情報があることを確認します。

## 手順

1. 必要なパッケージをインストールします。

```
# yum install sssd realmd adcli samba-common-tools oddjob oddjob-mkhomedir
```

2. **realmd** を使用して、最初の AD ドメイン **addomain1.com** に参加します。

```
# realm join ADDOMAIN1.COM
```

3. システムの **keytab** を一意の名前に変更します。

```
# mv /etc/krb5.keytab /etc/addomain1.com.krb5.keytab
```

4. **adcli** を使用して、2 番目の AD ドメインと追加のドメインに参加します。**-K** オプションを使用して、ホストの認証情報が書き込まれる Kerberos キータブの固有のパスを指定します。

```
# adcli join -D dc2.addomain2.com -K /etc/addomain2.com.krb5.keytab
```

5. **/etc/krb5.conf** を変更します。

- **includedir** オプションを追加して、SSSD 設定ファイルを追加します。
- **dns\_lookup\_kdc** オプションを使用して、AD ドメインコントローラーの DNS ルックアップを有効にします。

```
includedir /var/lib/sss/pubconf/krb5.include.d/

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

[libdefaults]
default_realm = ADDOMAIN1.COM
dns_lookup_realm = false
dns_lookup_kdc = true
ticket_lifetime = 24h
renew_lifetime = 7d
```

```
forwardable = true
```

```
...
```

6. `/etc/sss/sss.conf` を変更して、使用中のすべての AD ドメインに関する情報を追加します。

```
[sss]
services = nss, pam
config_file_version = 2
domains = addomain1.com, addomain2.com

[domain/addomain1.com]
id_provider = ad
access_provider = ad
krb5_keytab = /etc/addomain1.com.krb5.keytab
ldap_krb5_keytab = /etc/addomain1.com.krb5.keytab
ad_server = dc1.addomain1.com
ad_maximum_machine_account_password_age = 0
use_fully_qualified_names = true
default_shell=/bin/bash
override_homedir=/home/%d/%u

[domain/addomain2.com]
id_provider = ad
access_provider = ad
krb5_keytab = /etc/addomain2.com.krb5.keytab
ldap_krb5_keytab = /etc/addomain2.com.krb5.keytab
ad_server = dc2.addomain2.com
ad_maximum_machine_account_password_age = 0
use_fully_qualified_names = true
default_shell=/bin/bash
override_homedir=/home/%d/%u

[nss]

[pam]
```

- 各ドメインセクションで、`krb5_keytab` オプションおよび `ldap_krb5_keytab` オプションで各ドメインに対応する Kerberos キータブへのパスを指定します。
  - `ad_maximum_machine_account_password_age = 0` を設定して、ホストの Kerberos キーの更新を無効にします。
  - `use_fully_qualified_names = true` を設定して、異なるドメインのユーザーを区別します。
  - `override_homedir = /home/%d/%u` を設定し、異なるドメイン (`%d`) の各ユーザー (`%u`) がそれぞれ一意のホームディレクトリを受け取るようにします。たとえば、ユーザー `linuxuser@addomain1.com` のホームディレクトリは、`/home/addomain1.com/linuxuser` です。
7. SSH は、システムキータブからホストキーを取得し、GSSAPI/Kerberos を介してシングルサインオン機能を提供します。シングルサインオンを使用する場合は、現在の Kerberos ホストキーをすべてシステムキータブ `/etc/KRB5.keytab` にコピーします。

```
# ktutil
```

```
ktutil: rkt /etc/addomain1.com.krb5.keytab
ktutil: rkt /etc/addomain2.com.krb5.keytab
ktutil: wkt /etc/krb5.keytab
```

8. SSSD サービスを再起動して有効にします。

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

## 検証手順

1. 各 AD ドメインのユーザーの詳細を表示します。

```
# id administrator@addomain1.com
uid=1240800500(administrator@addomain1.com) gid=1240800513(domain
users@addomain1.com) groups=1240800513(domain
users@addomain1.com),1240800512(domain
admins@addomain1.com),1240800518(schema
admins@addomain1.com),1240800520(group policy creator
owners@addomain1.com),1240800572(denied rodc password replication
group@addomain1.com),1240800519(enterprise admins@addomain1.com)

# id administrator@addomain2.com
uid=1013800500(administrator@addomain2.com)
gid=1013800500(administrator@addomain2.com)
groups=1013800500(administrator@addomain2.com),1013800513(domain
users@addomain2.com)
```

2. 各ドメインのユーザーとしてログインし、ユーザーに正しいホームディレクトリが作成されていることを確認します。

```
# ssh administrator@addomain1.com@localhost
administrator@addomain1.com@localhost's password:
Creating directory '/home/addomain1.com/administrator'.
```

```
$ pwd
/home/addomain1.com/administrator
```

```
# ssh administrator@addomain2.com@localhost
administrator@addomain2.com@localhost's password:
Creating directory '/home/addomain2.com/administrator'.
```

```
$ pwd
/home/addomain2.com/administrator
```

## 1.5. AD プロバイダーが動的 DNS 更新を処理する方法

Active Directory (AD) は、アクティブではないレコードをタイムアウト (**aging**) および削除 (**scavenging**) して、DNS レコードをアクティブに管理します。

デフォルトでは、SSSD サービスは、RHEL クライアントの DNS レコードを以下の間隔で更新しません。

- アイデンティティプロバイダーがオンラインになるタイミング。
- RHEL システムが再起動したタイミング。
- `/etc/sss/sss.conf` 設定ファイルの `dyndns_refresh_interval` オプションで指定される間隔。デフォルト値は **86400** 秒 (24 時間) です。



### 注記

`dyndns_refresh_interval` オプションを DHCP リースと同じ間隔に設定すると、IP リースの更新後に DNS レコードを更新できます。

SSSD は、Kerberos/GSSAPI for DNS (GSS-TSIG) を使用して動的 DNS 更新を AD サーバーに送信します。そのため、必要な操作は、AD へのセキュアな接続を有効にするだけです。

### 関連情報

- `sss-ad(5)` の man ページ

## 1.6. AD プロバイダーの動的 DNS 設定の変更

以下の手順では、SSSD サービス内の設定を調節して、Active Directory 環境に参加している RHEL ホストの DNS レコードを自動更新する間隔を変化させます。

### 前提条件

- RHEL ホストが SSSD サービスを使用して Active Directory 環境に追加している。
- `/etc/sss/sss.conf` 設定ファイルを編集するには、**root** パーミッションが必要です。

### 手順

1. テキストエディターで `/etc/sss/sss.conf` 設定ファイルを開きます。
2. AD ドメインの `[domain]` セクションに以下のオプションを追加して、DNS レコードの更新間隔を 12 時間に設定し、PTR レコードの更新を無効にして DNS レコード Time To Live (TTL) を 1 時間に設定します。

```
[domain/ad.example.com]
id_provider = ad
...
dyndns_refresh_interval = 43200
dyndns_update_ptr = false
dyndns_ttl = 3600
```

3. `/etc/sss/sss.conf` 設定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、設定の変更を読み込みます。

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



## 注記

**sssd.conf** ファイルの **dyndns\_update** オプションを **false** に設定すると、動的 DNS 更新を無効にできます。

```
[domain/ad.example.com]
id_provider = ad
...
dyndns_update = false
```

## 関連情報

- の **sssd-ad(5)** の man ページ

## 1.7. AD プロバイダーが信頼されるドメインを処理する方法

本セクションでは、**/etc/sss/sss.conf** 設定ファイルで **id\_provider = ad** オプションを設定した場合に、どのように信頼できるドメインを SSSD が処理するかを説明します。

- SSSD は、AD フォレストのドメインを1つだけサポートします。SSSD が複数のフォレストから複数のドメインにアクセスする必要がある場合は、SSSD の代わりに信頼 (推奨) または **winbindd** サービスで IPA を使用することを検討してください。
- デフォルトでは、SSSD はフォレスト内のすべてのドメインを検出し、信頼されるドメイン内のオブジェクトの要求が到達すると、SSSD はこれを解決しようとします。信頼できるドメインに到達できない、または地理的に離れているために遅くなる場合は、**/etc/sss/sss.conf** に **ad\_enabled\_domains** パラメーターを設定して、どの信頼ドメインから SSSD がオブジェクトを解決するかを制限できます。
- デフォルトでは、完全修飾ユーザー名を使用して信頼されるドメインのユーザーを解決する必要があります。

## 関連情報

- **sss.conf(5)** の man ページ

## 1.8. REALM コマンド

**realmd** システムの主要なタスク領域は、以下の2つになります。

- ドメインでのシステム登録の管理
- ローカルシステムリソースへのアクセスが許可されるドメインユーザーの制御

**realmd** では、コマンドラインツールの **realm** を使用してコマンドを実行します。ほとんどの **realm** コマンドでは、ユーティリティーが実行するアクションと、アクションを実行するドメインやユーザーアカウントなどのエンティティーを指定する必要があります。

### 表1.4 realmd コマンド

コマンド	説明
<b>レルムコマンド</b>	
discover	ネットワーク上にあるドメインの検出スキャンを実行します。
join	指定したドメインにシステムを追加します。
leave	指定したドメインからシステムを削除します。
list	システムに設定したすべてのドメイン、または検出され設定されているすべてのドメインを表示します。
<b>ログインコマンド</b>	
permit	設定されているドメイン内の特定のユーザーまたはすべてのユーザーによるローカルシステムへのアクセスを有効にします。
deny	設定されているドメイン内の特定のユーザーまたはすべてのユーザーがローカルシステムにアクセスするのを制限します。

**realm** コマンドの詳細は、man ページの **realm(8)** を参照してください。

## 第2章 SAMBA WINBIND を使用して RHEL システムを AD に直接接続

本セクションでは、Samba Winbind を使用して RHEL システムを Active Directory (AD) に接続する方法を説明します。RHEL システムを AD に接続するには、2つのコンポーネントが必要です。1つのコンポーネント Samba Winbind は AD のアイデンティティおよび認証ソースと対話し、もう1つのコンポーネントである **realmd** は利用可能なドメインを検出し、基盤となる RHEL システムサービス (この場合は Samba Winbind) が AD ドメインに接続するように設定します。

- [Samba Winbind を使用した直接統合の概要](#)
- [直接統合でサポートされる Windows プラットフォーム](#)
- [AD および RHEL で一般的な暗号化タイプに対応](#)
- [RHEL システムの AD ドメインへの参加](#)
- [realm コマンド](#)

### 2.1. SAMBA WINBIND を使用した直接統合の概要

Samba Winbind は、Linux システムで Windows クライアントをエミュレートし、AD サーバーと通信します。

**realmd** サービスを使用すると、以下を実行して Samba Winbind を設定できます。

- ネットワーク認証およびドメインメンバーシップの標準的な設定。
- アクセス可能なドメインおよびレルムに関する情報を自動的に検出します。
- ドメインまたはレルムに参加するために高度な設定を必要としません。

以下の点に留意してください。

- マルチフォレストの AD 設定における Winbind との直接統合は、双方向の信頼が必要になります。
- **idmap\_ad** プラグインがリモートフォレストユーザーを正常に処理するには、リモートフォレストがローカルフォレストを信頼する必要があります。

Samba の **winbindd** サービスは、Name Service Switch (NSS) のインターフェースを提供し、ローカルシステムにログインする際にドメインユーザーが AD に対して認証できるようにします。

**winbindd** を使用すると、追加のソフトウェアをインストールしなくてもディレクトリーとプリンターを共有する設定が強化されます。詳細は、『さまざまな種類のサーバーのデプロイメント』の「[Samba をサーバーとして使用](#)」を参照してください。

#### 関連情報

- man ページの **realmd** を参照してください。
- man ページの **winbindd** を参照してください。

### 2.2. 直接統合でサポートされる WINDOWS プラットフォーム

RHEL システムと、以下のフォレストおよびドメインの機能レベルを使用する Active Directory フォレストを直接統合できます。

- フォレスト機能レベルの範囲 - Windows Server 2008 ~ Windows Server 2016
- ドメイン機能レベルの範囲 - Windows Server 2008 ~ Windows Server 2016

直接統合は、以下のサポート対象のオペレーティングシステムでテストされています。

- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2



### 注記

Windows Server 2019 では、新しい機能レベルが導入されていません。Windows Server 2019 が使用する機能レベルで最も高いものは Windows Server 2016 です。

## 2.3. AD および RHEL で一般的な暗号化タイプに対応

デフォルトでは、Samba Winbind は RC4、AES-128、および AES-256 の Kerberos 暗号化タイプに対応します。

RC4 暗号化は、新しい暗号化タイプ AES-128 および AES-256 よりも安全ではないと見なされるため、RHEL 8 ではデフォルトで非推奨となり、無効にされています。一方、Active Directory (AD) ユーザーの認証情報と AD ドメイン間の信頼は RC4 暗号化をサポートしており、AES 暗号化タイプに対応していない可能性があります。

一般的な暗号化タイプがないと、RHEL ホストと AD ドメイン間の通信が機能しないか、一部の AD アカウントが認証できない可能性があります。この状況を修正するには、以下の設定のいずれかを変更します。

- **Active Directory で AES 暗号化サポートを有効にする (推奨オプション)** - AD フォレストの AD ドメイン間で信頼されるようにするには、Microsoft の記事 [「AD DS: Security: Kerberos "Unsupported etype" error when accessing a resource in a trusted domain」](#) を参照してください。
- **RC4 サポートを RHEL で有効化**: AD ドメインコントローラーに対する認証が行われるすべての RHEL ホストで次のコマンドを実行します。
  1. **update-crypto-policies** コマンドを使用して、**DEFAULT** 暗号化ポリシーに加え **AD-SUPPORT** 暗号化サブポリシーを有効にします。

```
[root@host ~]# update-crypto-policies --set DEFAULT:AD-SUPPORT
Setting system policy to DEFAULT:AD-SUPPORT
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies
to fully take place.
```

2. ホストを再起動します。



## 重要

**AD-SUPPORT** 暗号化サブポリシーは、RHEL 8.3 以降でのみ利用できます。

- RHEL 8.2 以前は RC4 のサポートを有効にするには、**cipher = RC4-128+** でカスタム暗号化モジュールポリシーを作成および有効にします。詳細は、[ポリシー修飾子を使用したシステム全体の暗号化ポリシーのカスタマイズ](#)を参照してください。
- RHEL 8.0 および RHEL 8.1 で RC4 のサポートを有効にするには、`/etc/crypto-policies/back-ends/krb5.config` ファイルの **permitted\_enctypes** オプションに **+rc4** を追加します。

```
[libdefaults]
permitted_enctypes = aes256-cts-hmac-sha1-96 aes256-cts-hmac-sha384-192
camellia256-cts-cmac aes128-cts-hmac-sha1-96 aes128-cts-hmac-sha256-128
camellia128-cts-cmac +rc4
```

## 関連情報

- RHEL 暗号化ポリシーの使用方法は、『セキュリティ強化』ガイドの「[システム全体の暗号化ポリシーの使用](#)」を参照してください。

## 2.4. RHEL システムの AD ドメインへの参加

本セクションでは、**realmd** を使用して Samba Winbind を設定することで、Red Hat Enterprise Linux システムを AD ドメインに参加させる方法を説明します。

### 手順

1. AD で Kerberos 認証に非推奨の RC4 暗号化タイプが必要な場合は、RHEL でこの暗号のサポートを有効にします。

```
# update-crypto-policies --set DEFAULT:AD-SUPPORT
```

2. 以下のパッケージをインストールします。

```
# yum install realmd oddjob-mkhomedir oddjob samba-winbind-clients \
samba-winbind samba-common-tools samba-winbind-krb5-locator
```

3. ドメインメンバーでディレクトリーまたはプリンターを共有するには、**samba** パッケージをインストールします。

```
# yum install samba
```

4. 既存の Samba 設定ファイル `/etc/samba/smb.conf` をバックアップします。

```
# mv /etc/samba/smb.conf /etc/samba/smb.conf.bak
```

5. ドメインに参加します。たとえば、ドメイン **ad.example.com** に参加するには、以下のコマンドを実行します。

```
# realm join --membership-software=samba --client-software=winbind ad.example.com
```

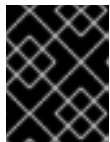
上記のコマンドを使用すると、**realm** ユーティリティーが自動的に以下を実行します。

- **ad.example.com** ドメインのメンバーシップに **/etc/samba/smb.conf** ファイルを作成します。
  - ユーザーおよびグループの検索用の **winbind** モジュールを、**/etc/nsswitch.conf** ファイルに追加します。
  - **/etc/pam.d/** ディレクトリーの PAM (プラグ可能な認証モジュール) 設定ファイルを更新します。
  - **winbind** サービスを起動し、システムの起動時にサービスを起動できるようにします。
6. 必要に応じて、**/etc/samba/smb.conf** ファイルの別の ID マッピングバックエンド、またはカスタマイズした ID マッピングを設定します。詳細は、『[さまざまな種類のサーバーのデプロイメント](#)』の「[Samba ID マッピングの概要](#)」セクションを参照してください。
  7. **/etc/krb5.conf** ファイルを編集し、以下のセクションを追加します。

```
[plugins]
  localauth = {
    module = winbind:/usr/lib64/samba/krb5/winbind_krb5_localauth.so
    enable_only = winbind
  }
```

8. **winbind** サービスが稼働していることを確認します。

```
# systemctl status winbind
...
Active: active (running) since Tue 2018-11-06 19:10:40 CET; 15s ago
```



### 重要

Samba がドメインのユーザーおよびグループの情報をクエリーできるようにするには、**smb** を起動する前に **winbind** サービスを実行する必要があります。

9. **samba** パッケージをインストールしてディレクトリーおよびプリンターを共有している場合は、**smb** サービスを有効化して開始します。

```
# systemctl enable --now smb
```

## 検証手順

1. AD ドメインの AD 管理者アカウントなど、AD ユーザーの詳細を表示します。

```
# getent passwd "AD\administrator"
AD\administrator:*:10000:10000::/home/administrator@AD:/bin/bash
```

2. AD ドメイン内のドメインユーザーグループのメンバーをクエリーします。

```
# getent group "AD\Domain Users"
AD\domain users:x:10000:user1,user2
```

3. オプションで、ファイルやディレクトリーに権限を設定する際に、ドメインのユーザーおよびグループを使用できることを確認します。たとえば、`/srv/samba/example.txt` ファイルの所有者を **AD\administrator** に設定し、グループを **AD\Domain Users** に設定するには、以下のコマンドを実行します。

```
# chown "AD\administrator":"AD\Domain Users" /srv/samba/example.txt
```

4. Kerberos 認証が期待どおりに機能することを確認します。
  - a. AD ドメインメンバーで、**administrator@AD.EXAMPLE.COM** プリンシパルのチケットを取得します。

```
# kinit administrator@AD.EXAMPLE.COM
```

- b. キャッシュされた Kerberos チケットを表示します。

```
# klist
Ticket cache: KCM:0
Default principal: administrator@AD.EXAMPLE.COM

Valid starting    Expires          Service principal
01.11.2018 10:00:00 01.11.2018 20:00:00
krbtgt/AD.EXAMPLE.COM@AD.EXAMPLE.COM
renew until 08.11.2018 05:00:00
```

5. 利用可能なドメインの表示:

```
# wbinfo --all-domains
BUILTIN
SAMBA-SERVER
AD
```

## 関連情報

- 非推奨の RC4 暗号化を使用しない場合は、AD で AES 暗号化タイプを有効にすることができます。『さまざまな種類のサーバーのデプロイメント』の「[GPO を使用した Active Directory で AES 暗号化タイプの有効化](#)」を参照してください。
- **realm** ユーティリティーの詳細は、man ページの **realm(8)** を参照してください。

## 2.5. REALM コマンド

**realmd** システムの主要なタスク領域は、以下の 2 つになります。

- ドメインでのシステム登録の管理
- ローカルシステムリソースへのアクセスが許可されるドメインユーザーの制御

**realmd** では、コマンドラインツールの **realm** を使用してコマンドを実行します。ほとんどの **realm** コマンドでは、ユーティリティーが実行するアクションと、アクションを実行するドメインやユーザーアカウントなどのエンティティーを指定する必要があります。

表2.1 **realmd** コマンド

コマンド	説明
<b>レルムコマンド</b>	
discover	ネットワーク上にあるドメインの検出スキャンを実行します。
join	指定したドメインにシステムを追加します。
leave	指定したドメインからシステムを削除します。
list	システムに設定したすべてのドメイン、または検出され設定されているすべてのドメインを表示します。
<b>ログインコマンド</b>	
permit	設定されているドメイン内の特定のユーザーまたはすべてのユーザーによるローカルシステムへのアクセスを有効にします。
deny	設定されているドメイン内の特定のユーザーまたはすべてのユーザーがローカルシステムにアクセスするのを制限します。

**realm** コマンドの詳細は、man ページの **realm(8)** を参照してください。

## 第3章 AD への直接接続の管理

本セクションでは、Active Directory への接続を変更および管理する方法を説明します。

### 前提条件

- RHEL システムを Active Directory ドメインに接続している。

### 3.1. デフォルトの KERBEROS ホストのキータブの更新間隔を変更

SSSD は、**adcli** パッケージがインストールされていると、AD 環境で Kerberos ホストキータブファイルを自動的に更新します。デーモンは、マシンアカウントのパスワードが設定されている値よりも古いかどうかを毎日確認し、必要に応じてそのパスワードを更新します。

デフォルトの更新間隔は 30 日です。デフォルトを変更するには、以下の手順に従います。

#### 手順

1. 以下のパラメーターを `/etc/sss/sss.conf` ファイルの AD プロバイダーに追加します。

```
ad_maximum_machine_account_password_age = value_in_days
```

2. SSSD を再起動します。

```
# systemctl restart sssd
```

3. Kerberos ホストのキータブの自動更新を無効にするには、**ad\_maximum\_machine\_account\_password\_age = 0** を設定します。

#### 関連情報

- man ページの **adcli(8)**
- **sss.conf(5)** の man ページ

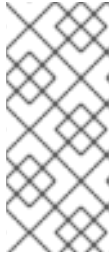
### 3.2. AD ドメインからの RHEL システムの削除

この手順では、Active Directory (AD) ドメインから RHEL システムを削除する方法を説明します。

#### 手順

1. **realm leave** コマンドを使用して、ID ドメインからシステムを削除します。このコマンドは、SSSD およびローカルシステムからドメイン設定を削除します。

```
# realm leave ad.example.com
```



## 注記

クライアントがドメインからいなくなっても、アカウントは AD から削除されず、ローカルクライアントの設定のみが削除されます。AD アカウントを削除する場合は、**--remove** オプションを指定してコマンドを実行します。ユーザーパスワードの入力が求められ、Active Directory からアカウントを削除する権限が必要になります。

2. **realm leave** コマンドに **-U** オプションを付けて実行し、システムを ID ドメインから削除する別のユーザーを指定します。  
デフォルトでは、**realm leave** コマンドはデフォルトの管理者として実行されます。AD の場合は、管理者アカウントは **Administrator** と呼ばれます。ドメインに参加するために別のユーザーを使用していた場合は、そのユーザーとして削除を実行しないといけない場合があります。

```
# realm leave [ad.example.com] -U [AD.EXAMPLE.COM\user]
```

コマンドは最初に認証情報なしで接続を試みますが、必要に応じてパスワードが要求されます。

## 検証手順

- ドメインが設定されていないことを確認します。

```
# realm discover [ad.example.com]
ad.example.com
  type: kerberos
  realm-name: EXAMPLE.COM
  domain-name: example.com
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: oddjob
  required-package: oddjob-mkhomedir
  required-package: sssd
  required-package: adcli
  required-package: samba-common-tools
```

## 関連情報

- **realm(8)** の man ページを参照してください。

## 3.3. SSSD でドメイン解決順序を設定して、AD ユーザーの短縮名を解決する手順

デフォルトでは、**ad\_username@ad.example.com** および **group@ad.example.com** など完全修飾ユーザー名を指定して、SSSD サービスで AD に接続された RHEL ホスト上にある Active Directory (AD) ユーザーおよびグループを解決する必要があります。

この手順では、SSSD 設定でドメイン解決の順序を設定し、**ad\_username** などの短縮名を使用して AD ユーザーおよびグループを解決できるようにします。この設定例では、以下の順序でユーザーおよびグループを検索します。

1. Active Directory (AD) 子ドメイン **subdomain2.ad.example.com**

2. AD 子ドメイン **subdomain1.ad.example.com**
3. AD root ドメイン **ad.example.com**

### 前提条件

- SSSD サービスを使用して、RHEL ホストを直接 AD に接続している。

### 手順

1. テキストエディターで **/etc/sss/sss.conf** ファイルを開きます。
2. このファイルの **[sss]** セクションに **domain\_resolution\_order** オプションを設定します。

```
domain_resolution_order = subdomain2.ad.example.com, subdomain1.ad.example.com,
ad.example.com
```

3. ファイルを保存してから閉じます。
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@ad-client ~]# systemctl restart sssd
```

### 検証手順

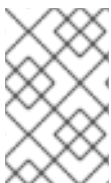
- 短縮名だけを使用して、最初のドメインからユーザー情報を取得できることを確認します。

```
[root@ad-client ~]# id <user_from_subdomain2>
uid=1916901142(user_from_subdomain2) gid=1916900513(domain users)
groups=1916900513(domain users)
```

## 3.4. ドメインユーザーのログインパーミッションの管理

デフォルトでは、ドメイン側のアクセス制御が適用されます。これは、Active Directory (AD) ユーザーのログインポリシーが AD ドメイン自体に定義されることを意味します。クライアント側のアクセス制御を使用できるように、このデフォルトの動作は上書きできます。クライアント側のアクセス制御では、ログインパーミッションはローカルポリシーでのみ定義されます。

ドメインがクライアント側のアクセス制御を適用する場合は、**realmd** を使用して、そのドメインのユーザーの基本的なアクセスルールである **allow** または **deny** を設定できます。

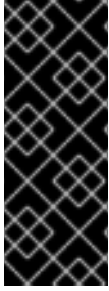


### 注記

アクセスルールは、システムにあるすべてのサービスへのアクセスを許可または拒否します。特定のシステムリソースまたはドメインに、より具体的なアクセスルールを設定する必要があります。

### 3.4.1. ドメイン内でユーザーのアクセス権を有効化

本セクションでは、ドメイン内のユーザーのアクセス権を有効にする方法を説明します。



## 重要

特定のユーザーまたはグループのアクセスのみを許可する方が、一部のユーザーへのアクセスを拒否して、他のすべてのユーザーにアクセスを許可するよりも安全です。したがって、デフォルトで全ユーザーにアクセスを許可し、レルムの許可 `-x` を使用して特定のユーザーのみを拒否することは推奨されません。Red Hat では、代わりに、すべてのユーザーに対してデフォルトのアクセス禁止ポリシーを維持し、レルム許可を使用して選択したユーザーのアクセスのみを許可することが推奨されます。

## 前提条件

- RHEL システムが Active Directory ドメインのメンバーである。

## 手順

1. すべてのユーザーにアクセス権を付与します。

```
# realm permit --all
```

2. 特定のユーザーにアクセス権を付与します。

```
$ realm permit aduser01@example.com
$ realm permit 'AD.EXAMPLE.COM\aduser01'
```

現在、アクセスを許可できるのはプライマリードメインのユーザーのみで、信頼できるドメインのユーザーには許可できません。これは、ユーザーログインにドメイン名を含める必要があり、SSSD は、現在、**realmd** に利用可能な子ドメインに関する情報を提供できないためです。

## 検証手順

1. SSH を使用して、[aduser01@example.com](mailto:aduser01@example.com) ユーザーとしてサーバーにログインします。

```
$ ssh aduser01@example.com@server_name
[aduser01@example.com@server_name ~]$
```

2. ssh コマンドをもう一度使用して、[aduser02@example.com](mailto:aduser02@example.com) ユーザーと同じサーバーにアクセスします。

```
$ ssh aduser02@example.com@server_name
Authentication failed.
```

[aduser02@example.com](mailto:aduser02@example.com) がシステムへのアクセスを拒否する方法に注目してください。[aduser01@example.com](mailto:aduser01@example.com) ユーザーにのみ、システムにログインするパーミッションを付与しています。指定したログインポリシーが原因で、その Active Directory ドメインの他のユーザーはすべて拒否されます。



## 注記

**sssd.conf** ファイルで **use\_fully\_qualified\_names** を `true` に設定すると、すべての要求で完全修飾ドメイン名を使用する必要があります。ただし、**use\_fully\_qualified\_names** を `false` に設定すると、要求で完全修飾名を使用できますが、出力には簡略化されたバージョンのみが表示されます。

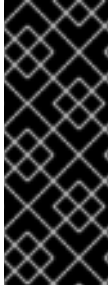


## 関連情報

- **realm(8)** の man ページを参照してください。

### 3.4.2. ドメイン内でユーザーのアクセス権を拒否

本セクションでは、ドメイン内のすべてのユーザーへのアクセスを拒否する方法を説明します。



#### 重要

特定のユーザーまたはグループのアクセスのみを許可する方が、一部のユーザーへのアクセスを拒否して、他のすべてのユーザーにアクセスを許可するよりも安全です。したがって、デフォルトで全ユーザーにアクセスを許可し、レルムの許可 `-x` を使用して特定のユーザーのみを拒否することは推奨されません。Red Hat では、代わりに、すべてのユーザーに対してデフォルトのアクセス禁止ポリシーを維持し、レルム許可を使用して選択したユーザーのアクセスのみを許可することが推奨されます。

#### 前提条件

- RHEL システムが Active Directory ドメインのメンバーである。

#### 手順

1. ドメイン内のすべてのユーザーへのアクセスを拒否します。

```
# realm deny --all
```

このコマンドは、**realm** アカウントがローカルマシンにログインできないようにします。**realm permit** を使用して、ログインを特定アカウントに制限します。

2. ドメインユーザーの **login-policy** が **deny-any-login** に設定されていることを確認します。

```
[root@replica1 ~]# realm list
example.net
  type: kerberos
  realm-name: EXAMPLE.NET
  domain-name: example.net
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: oddjob
  required-package: oddjob-mkhomedir
  required-package: sssd
  required-package: adcli
  required-package: samba-common-tools
  login-formats: %U@example.net
  login-policy: deny-any-login
```

3. `-x` オプションを使用して特定のユーザーへのアクセスを拒否します。

```
$ realm permit -x 'AD.EXAMPLE.COM\aduser02'
```

#### 検証手順

- SSH を使用して、**aduser01@example.net** ユーザーとしてサーバーにログインします。

```
$ ssh aduser01@example.net@server_name
Authentication failed.
```



### 注記

**sssd.conf** ファイルで **use\_fully\_qualified\_names** を true に設定すると、すべての要求で完全修飾ドメイン名を使用する必要があります。ただし、**use\_fully\_qualified\_names** を false に設定すると、要求で完全修飾名を使用できますが、出力には簡略化されたバージョンのみが表示されます。

### 関連情報

- **realm(8)** の man ページを参照してください。

## 3.5. RHEL でのグループポリシーアクセス制御の適用

**Group Policy Object (GPO)** は、AD 環境のコンピューターおよびユーザーに適用可能な Microsoft Active Directory (AD) に保存されているアクセス制御設定の集合です。管理者は、AD で GPO を指定することで、AD に参加している Windows クライアントと Red Hat Enterprise Linux (RHEL) ホストの両方が許可するログインポリシーを定義できます。

以下のセクションでは、環境で GPO を管理する方法を説明します。

- [「SSSD が GPO アクセス制御ルールを解釈する方法」](#)
- [「SSSD がサポートする GPO 設定の一覧」](#)
- [「GPO 強制を制御する SSSD オプションの一覧」](#)
- [「GPO アクセス制御モードの変更」](#)
- [「AD GUI での RHEL ホストの GPO の作成および設定」](#)

### 3.5.1. SSSD が GPO アクセス制御ルールを解釈する方法

デフォルトでは、SSSD は Active Directory (AD) ドメインコントローラーからグループポリシーオブジェクト (GPO) を取得し、ユーザーが AD に参加している特定の RHEL ホストにログインできるかどうかを判断します。

SSSD は AD **Windows Logon Rights** を Pluggable Authentication Module (PAM) サービス名にマッピングし、GNU/Linux 環境でこれらのパーミッションを強制します。

AD 管理者として、**セキュリティフィルター**にリストすることで、GPO ルールのスコープを特定のユーザー、グループ、またはホストに制限できます。

#### 3.5.1.1. ホストによるフィルタリングの制限

SSSD の古いバージョンは、AD GPO セキュリティフィルター内のホストを評価しません。

- **RHEL 8.3.0 以降:** SSSD は、セキュリティフィルター内のユーザー、グループ、およびホストをサポートします。
- **8.3.0 よりも古い RHEL バージョン:** SSSD はホストエントリを無視し、セキュリティフィ

ルターでユーザーおよびグループのみをサポートします。  
SSSD が GPO ベースのアクセス制御を特定のホストに適用するには、AD ドメインで新しい組織単位(OU)を作成し、システムを新しい OU に移動してから GPO をこの OU にリンクします。

### 3.5.1.2. グループ別フィルタリングの制限

SSSD は現在、セキュリティ識別子 (SID) **S-1-5-32-544** を持つ **Administrators** など、Active Directory の組み込みグループをサポートしていません。Red Hat は、RHEL ホストを対象とする AD GPO で AD の組み込みグループを使用することを推奨しています。

#### 関連情報

- Windows GPO オプションとそれに対応する SSSD オプションの一覧は、「[SSSD がサポートする GPO 設定の一覧](#)」を参照してください。

### 3.5.2. SSSD がサポートする GPO 設定の一覧

以下の表は、Windows の **グループポリシー管理エディター** で指定される Active Directory GPO オプションに対応する SSSD オプションを示しています。

表3.1 SSSD が取得した GPO アクセス制御オプション

GPO オプション	対応する <code>sssd.conf</code> オプション
ローカルでのログオンの許可 ローカルでのログオンの拒否	<code>ad_gpo_map_interactive</code>
リモートデスクトップサービスを介したログオンの許可 リモートデスクトップサービスを介したログオンの拒否	<code>ad_gpo_map_remote_interactive</code>
ネットワークからこのコンピューターへのアクセス ネットワークからこのコンピューターへのアクセスを拒否	<code>ad_gpo_map_network</code>
バッチジョブとしてのログオンの許可 バッチジョブとしてのログオンの拒否	<code>ad_gpo_map_batch</code>
サービスとしてのログオンの許可 サービスとしてのログオンの拒否	<code>ad_gpo_map_service</code>

- GPO オプションにマップする PAM (プラグ可能な認証モジュール) サービスなど、この `sssd.conf` 設定の詳細は、`sssd-ad(5)` の man ページを参照してください。

### 3.5.3. GPO 強制を制御する SSSD オプションの一覧

#### 3.5.3.1. `ad_gpo_access_control` オプション

`/etc/sss/sss.conf` ファイルに `ad_gpo_access_control` オプションを設定して、GPO ベースのアクセス制御が動作する 3 種類のモードを選択できます。

表3.2 `ad_gpo_access_control` の値の表

<code>ad_gpo_access_control</code> の値	動作
<b>enforcing</b>	GPO ベースのアクセス制御ルールが評価され、適用されます。これは、RHEL 8 ではデフォルトの設定です。
<b>permissive</b>	GPO ベースのアクセス制御ルールは評価されますが、強制されません。 <b>syslog</b> メッセージは、アクセスが拒否される度に記録されます。これは、RHEL 7 ではデフォルトの設定です。このモードは、ユーザーがログインを継続できるように、ポリシーの調整をテストするのに適しています。
<b>disabled</b>	GPO ベースのアクセス制御ルールは、評価も強制もされません。

### 3.5.3.2. `ad_gpo_implicit_deny` オプション

`ad_gpo_implicit_deny` オプションは、デフォルトで **False** に設定されます。このデフォルトの状態では、適用可能な GPO が見つからない場合にユーザーがアクセスが許可されます。このオプションを **True** に設定する場合は、GPO ルールを使用したユーザーアクセスを明示的に許可する必要があります。

この機能を使用してセキュリティを強化することはできますが、アクセスを意図せずに拒否しないように注意してください。Red Hat は、`ad_gpo_access_control` が **permissive** に設定されている間に、この機能をテストすることを推奨します。

以下の表では、AD サーバー側で定義したログイン権限と `ad_gpo_implicit_deny` の値に基づいてユーザーがアクセスを許可または拒否されるタイミングを表しています。

表3.3 `ad_gpo_implicit_deny` が **False** (デフォルト) に設定されているログイン動作

allow-rules	deny-rules	結果
なし	なし	すべてのユーザーが許可
なし	present	deny-rules でないユーザーのみが許可
present	なし	allow-rules のユーザーのみを許可
あり	あり	allow-rules のユーザーのみが許可されますが、拒否ルールでは許可されません

表3.4 `ad_gpo_implicit_deny` が **True** に設定されているログイン動作

allow-rules	deny-rules	結果
なし	なし	すべてのユーザーを拒否

allow-rules	deny-rules	結果
なし	あり	すべてのユーザーを拒否
あり	なし	allow-rules のユーザーのみを許可
あり	あり	allow-rules のユーザーのみが許可されますが、拒否ルールでは許可されません

### 関連情報

- SSSD で GPO 強制モードを変更する手順は、「[GPO アクセス制御モードの変更](#)」を参照してください。
- さまざまな GPO モードの詳細は、`sssd-ad(5) man` ページの `ad_gpo_access_control` のエントリーを参照してください。

#### 3.5.4. GPO アクセス制御モードの変更

この手順では、GPO ベースのアクセス制御ルールが Active Directory (AD) 環境に参加している RHEL ホストでどのように評価されるかを変更します。

この例では、テスト目的で GPO 操作モードを **Enforcing** (デフォルト) から **Permissive** に変更します。

## 重要

以下のエラーが表示された場合には、GPO ベースのアクセス制御により Active Directory ユーザーはログインできません。

- `/var/log/secure:`

```
Oct 31 03:00:13 client1 sshd[124914]: pam_sss(sshd:account): Access denied for user aduser1: 6 (Permission denied)
Oct 31 03:00:13 client1 sshd[124914]: Failed password for aduser1 from 127.0.0.1 port 60509 ssh2
Oct 31 03:00:13 client1 sshd[124914]: fatal: Access denied for user aduser1 by PAM account configuration [preauth]
```

- `/var/log/sss/sss__example.com_.log:`

```
(Sat Oct 31 03:00:13 2020) [sss[be[example.com]]]
[ad_gpo_perform_hbac_processing] (0x0040): GPO access check failed: [1432158236](Host Access Denied)
(Sat Oct 31 03:00:13 2020) [sss[be[example.com]]] [ad_gpo_cse_done] (0x0040): HBAC processing failed: [1432158236](Host Access Denied)
(Sat Oct 31 03:00:13 2020) [sss[be[example.com]]] [ad_gpo_access_done] (0x0040): GPO-based access control failed.
```

これが望ましくない動作の場合は、AD で正しい GPO 設定のトラブルシューティング中に、この手順で説明されているように、`ad_gpo_access_control` を **Permissive** に設定できます。

## 前提条件

- SSSD を使用して RHEL ホストを AD 環境に追加している。
- `/etc/sss/sss.conf` 設定ファイルの編集には、**root** 権限が必要になります。

## 手順

1. SSSD サービスを停止します。

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

2. テキストエディターで `/etc/sss/sss.conf` ファイルを開きます。
3. AD ドメインの **domain** セクションで、`ad_gpo_access_control` を **Permissive** に設定します。

```
[domain/example.com]
ad_gpo_access_control=permissive
...
```

4. `/etc/sss/sss.conf` ファイルを保存します。
5. SSSD サービスを再起動して、設定の変更を読み込みます。

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

## 関連情報

- さまざまな GPO アクセス制御モードの一覧は、「[GPO の適用を制御する SSSD オプションの一覧](#)」を参照してください。

### 3.5.5. AD GUI での RHEL ホストの GPO の作成および設定

以下の手順では、RHEL ホストへのログオンアクセスを制御するために、Active Directory (AD) グラフィカルユーザーインターフェース(GUI)で Group Policy Object (GPO) を作成します。

#### 前提条件

- SSSD を使用して RHEL ホストを AD 環境に追加している。
- GUI を使用して AD に変更を加えるための AD 管理者権限がある。

#### 手順

1. **Active Directory ユーザーおよびコンピューター** 内で、新しい GPO に関連付ける組織単位 (OU) を作成します。
  - a. ドメインを右クリックします。
  - b. **New** を選択します。
  - c. **Organizational Unit** を選択します。
2. Active Directory に参加しているときに) RHEL ホストを表す Computer オブジェクトの名前をクリックし、新しい OU にドラッグします。独自の OU に RHEL ホストがあると、GPO はこのホストをターゲットとします。
3. **Group Policy Management Editor** で、作成した OU の GPO を新規作成します。
  - a. **Forest** を展開します。
  - b. **Domain** を展開します。
  - c. ドメインを展開します。
  - d. 新しい OU をダブルクリックします。
  - e. **Create a GPO in this domain** を選択します。
4. **Allow SSH access** または **Allow Console/GUI access** など、新規 GPO の名前を指定して **OK** をクリックします。
5. 新規 GPO を編集します。
  - a. **Group Policy Management** エディター内で OU を選択します。
  - b. 右クリックして **Edit** を選択します。
  - c. **User Queuing Assignment** を選択します。
  - d. **Computer Configuration** を選択します
  - e. **Policies** を選択します。

- f. **Windows Settings** を選択します。
  - g. **セキュリティー設定** を選択します。
  - h. **Local Policies** を選択します。
  - i. **User Queuing Assignment** を選択します。
6. ログインパーミッションを割り当てます。
    - a. **Allow log on locally** をダブルクリックしてローカルコンソール/GUI アクセスを付与します。
    - b. **Allow log on through Remote Desktop Services** をダブルクリックして、SSH アクセスを付与します。
  7. これらのポリシーのいずれかにアクセスするユーザーをポリシー自体に追加します。
    - a. **Add User or Group** をクリックします。
    - b. 空白フィールドにユーザー名を入力します。
    - c. **OK** をクリックします。

#### 関連情報

- Group Policy Objects の詳細は、Microsoft ドキュメントの「[Group Policy Objects](#)」を参照してください。

#### 3.5.6. 関連情報

- RHEL ホストを Active Directory 環境にジョインする方法は、「[SSSD を使用した RHEL システムを AD に直接接続](#)」を参照してください。