Red Hat Enterprise Linux 9

メールサーバーのデプロイ

メールサーバーサービスの設定および維持

Last Updated: 2024-05-02

Red Hat Enterprise Linux 9 メールサーバーのデプロイ

メールサーバーサービスの設定および維持

法律上の通知

概要

Red Hat Enterprise Linux では、メールトランスポートエージェント Postfix を SMTP サービスとして使用し、メール配信エージェント Dovecot を IMAP および POP3 サービスとして使用することで、お客様および内部ユーザーに信頼できる安全なメールサービスを提供できます。どちらのサービスも相互に統合され、アカウントデータを保存し、ユーザーを認証するための LDAP ディレクトリーなどの中央のバックエンドをサポートします。

目次

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 プロパティー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、Red Hat CTO である Chris Wright のメッセージ を参照してください。

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 DOVECOT IMAP および POP3 サーバーの設定と管理

Dovecot は、セキュリティーを重視する高パフォーマンスのメール配信エージェント (MDA) です。IMAP または POP3 互換の電子メールクライアントを使用して Dovecot サーバーに接続し、電子メールを読んだりダウンロードしたりできます。

Dovecot の主な機能:

- セキュリティーを重視する設計と実装
- 大規模環境でのパフォーマンスを向上させるために、高可用性を実現する双方向レプリケーションをサポート
- 高パフォーマンスの **dbox** メールボックス形式だけでなく、互換性の理由から **mbox** と **Maildir** もサポート
- 破損したインデックスファイルの修正などの自己修復機能
- IMAP 標準への準拠
- IMAP および POP3 クライアントのバグを回避するための回避策をサポート

1.1. PAM 認証を使用した DOVECOT サーバーのセットアップ

Dovecot は、ユーザーデータベースとして Name Service Switch (NSS) インターフェイスをサポートし、認証バックエンドとして Pluggable Authentication Module (PAM) フレームワークをサポートします。この設定により、Dovecot は、NSS を介してサーバー上でローカルに利用可能なユーザーにサービスを提供できます。

アカウントが次の場合に PAM 認証を使用します。

- **/etc/passwd** ファイルでローカルに定義されている。
- リモートデータベースに保存されているが、System Security Services Daemon (SSSD) またはその他の NSS プラグインを介してローカルで利用できる。

1.1.1. Dovecot のインストール

dovecot パッケージは以下を提供します。

- **dovecot** サービスとそれを管理するユーティリティー
- Dovecot がオンデマンドで開始するサービス (認証など)
- サーバーサイドメールフィルタリングなどのプラグイン
- **/etc/dovecot/** ディレクトリーの設定ファイル
- **/usr/share/doc/dovecot/** ディレクトリーのドキュメント

手順

- **dovecot** パッケージをインストールします。

```
# dnf install dovecot
```

注記

Dovecot がすでにインストールされていて、クリーンな設定ファイルが必要な場合は、**/etc/dovecot/** ディレクトリーを名前変更するか削除してください。その後、パッケージを再インストールします。設定ファイルを削除しないと、**dnf reinstall dovecot** コマンドは **/etc/dovecot/** 内の設定ファイルをリセットしません。

次のステップ

- Dovecot サーバーでの TLS 暗号化の設定。

1.1.2. Dovecot サーバーでの TLS 暗号化の設定

Dovecot はセキュアなデフォルト設定を提供します。たとえば、TLS はデフォルトで有効になっており、認証情報と暗号化されたデータをネットワーク経由で送信します。Dovecot サーバーで TLS を設定するには、証明書と秘密鍵ファイルへのパスを設定するだけです。さらに、Diffie-Hellman パラメーターを生成して使用し、Perfect Forward Secrecy (PFS) を提供することで、TLS 接続のセキュリティーを強化できます。

前提条件

- Dovecot がインストールされています。
- 次のファイルが、サーバー上のリストされた場所にコピーされました。
  - サーバー証明書: **/etc/pki/dovecot/certs/server.example.com.crt**
  - 秘密鍵: **/etc/pki/dovecot/private/server.example.com.key**
  - 認証局 (CA) 証明書: **/etc/pki/dovecot/certs/ca.crt**
- サーバー証明書の **Subject DN** フィールドのホスト名は、サーバーの完全修飾ドメイン名 (FQDN) と一致します。
- サーバーが RHEL 9.2 以降を実行し、FIPS モードが有効になっている場合、クライアントが Extended Master Secret (EMS) 拡張機能をサポートしているか、TLS 1.3 を使用している必要があります。EMS を使用しない TLS 1.2 接続は失敗します。詳細は、ナレッジベースの記事 TLS extension "Extended Master Secret" enforced を参照してください。

手順

1. 秘密鍵ファイルにセキュアな権限を設定します。

```
# chown root:root /etc/pki/dovecot/private/server.example.com.key
# chmod 600 /etc/pki/dovecot/private/server.example.com.key
```

2. Diffie-Hellman パラメーターを使用してファイルを生成します。

```
# openssl dhparam -out /etc/dovecot/dh.pem 4096
```

サーバーのハードウェアとエントロピーによっては、4096 ビットの Diffie-Hellman パラメーターを生成するのに数分かかる場合があります。

3. **/etc/dovecot/conf.d/10-ssl.conf** ファイルで証明書と秘密鍵ファイルへのパスを設定します。

a. **ssl_cert** および **ssl_key** パラメーターを更新し、サーバーの証明書と秘密鍵へのパスを使用するように設定します。

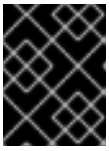
```
ssl_cert = </etc/pki/dovecot/certs/server.example.com.crt
ssl_key = </etc/pki/dovecot/private/server.example.com.key
```

b. **ssl_ca** パラメーターをコメント解除し、CA 証明書へのパスを使用するように設定します。

```
ssl_ca = </etc/pki/dovecot/certs/ca.crt
```

c. **ssl_dh** パラメーターをコメント解除し、Diffie-Hellman パラメーターファイルへのパスを使用するように設定します。

```
ssl_dh = </etc/dovecot/dh.pem
```



重要

Dovecot がファイルからパラメーターの値を確実に読み取るようにするには、パスの先頭に **<** 文字を付ける必要があります。

次のステップ

- 仮想ユーザーを使用するための Dovecot の準備

関連情報

- **/usr/share/doc/dovecot/wiki/SSL.DovecotConfiguration.txt**

1.1.3. 仮想ユーザーを使用するための Dovecot の準備

デフォルトでは、Dovecot はサービスを使用するユーザーとして、ファイルシステム上で多くのアクションを実行します。ただし、1 人のローカルユーザーを使用してこれらのアクションを実行するように Dovecot バックエンドを設定すると、複数の利点があります。

- Dovecot は、ユーザーの ID (UID) を使用する代わりに、特定のローカルユーザーとしてファイルシステムアクションを実行します。
- ユーザーは、サーバー上でローカルに利用できる必要はありません。
- すべてのメールボックスとユーザー固有のファイルを 1 つのルートディレクトリーに保存できます。
- ユーザーは UID とグループ ID (GID) を必要としないため、管理作業が軽減されます。
- サーバー上のファイルシステムにアクセスできるユーザーは、これらのファイルにアクセスできないため、メールボックスやインデックスを危険にさらす可能性はありません。
- レプリケーションのセットアップはより簡単です。

前提条件

- Dovecot がインストールされています。

手順

1. **vmail** ユーザーを作成します。

```
# useradd --home-dir /var/mail/ --shell /usr/sbin/nologin vmail
```

Dovecot は後でこのユーザーを使用してメールボックスを管理します。セキュリティー上の理由から、この目的で **dovecot** または **dovenull** システムユーザーを使用しないでください。

2. **/var/mail/** 以外のパスを使用する場合は、それに SELinux コンテキスト **mail_spool_t** を設定します。例:

```
# semanage fcontext -a -t mail_spool_t "<path>(/.*)?"
# restorecon -Rv <path>
```

3. **/var/mail/** への書き込み権限を **vmail** ユーザーにのみ付与します。

```
# chown vmail:vmail /var/mail/
# chmod 700 /var/mail/
```

4. **/etc/dovecot/conf.d/10-mail.conf** ファイルの **mail_location** パラメーターをコメント解除し、メールボックスの形式と場所を設定します。

```
mail_location = sdbox:/var/mail/%n/
```

この設定の場合:

- Dovecot は、**single** モードで高パフォーマンスの **dbox** メールボックス形式を使用します。このモードでは、サービスは、**maildir** 形式と同様に、各メールを個別のファイルに保存します。
- Dovecot はパス内の **%n** 変数をユーザー名に解決します。これは、各ユーザーがメールボックス用に個別のディレクトリーを持つようにするために必要です。

次のステップ

- PAM を Dovecot 認証バックエンドとして使用する。

関連情報

- **/usr/share/doc/dovecot/wiki/VirtualUsers.txt**
- **/usr/share/doc/dovecot/wiki/MailLocation.txt**
- **/usr/share/doc/dovecot/wiki/MailboxFormat.dbox.txt**
- **/usr/share/doc/dovecot/wiki/Variables.txt**

1.1.4. PAM を Dovecot 認証バックエンドとして使用する

デフォルトでは、Dovecot は Name Service Switch (NSS) インターフェイスをユーザーデータベースとして使用し、Pluggable Authentication Module (PAM) フレームワークを認証バックエンドとして使用します。

設定をカスタマイズして Dovecot を環境に適応させ、仮想ユーザー機能を使用して管理を簡素化します。

前提条件

- Dovecot がインストールされています。
- 仮想ユーザー機能が設定されています。

手順

1. **/etc/dovecot/conf.d/10-mail.conf** ファイルの **first_valid_uid** パラメーターを更新して、Dovecot に対して認証できる最小のユーザー ID (UID) を定義します。

```
first_valid_uid = 1000
```

デフォルトでは、**1000** 以上の UID を持つユーザーが認証を受けることができます。必要に応じて、**last_valid_uid** パラメーターを設定して、Dovecot がログインを許可する最大の UID を定義することもできます。

2. **/etc/dovecot/conf.d/auth-system.conf.ext** ファイルで、次のように **override_fields** パラメーターを **userdb** セクションに追加します。

```
userdb {
  driver = passwd
  override_fields = uid=vmail gid=vmail home=/var/mail/%n/
}
```

固定値のため、Dovecot は **/etc/passwd** ファイルからこれらの設定をクエリーしません。そのため、**/etc/passwd** に定義されたホームディレクトリーが存在する必要はありません。

次のステップ

- Dovecot 設定を完了します。

関連情報

- **/usr/share/doc/dovecot/wiki/PasswordDatabase.PAM.txt**
- **/usr/share/doc/dovecot/wiki/VirtualUsers.Home.txt**

1.1.5. Dovecot 設定の完了

Dovecot をインストールして設定したら、**firewalld** サービスで必要なポートを開き、サービスを有効にして開始します。その後、サーバーをテストできます。

前提条件

- 以下は Dovecot で設定されています。
  - TLS 暗号化
  - 認証バックエンド
- クライアントは認証局 (CA) 証明書を信頼します。

手順

1. IMAP または POP3 サービスのみをユーザーに提供する場合は、**/etc/dovecot/dovecot.conf** ファイルの **protocol** パラメーターをコメント解除し、必要なプロトコルに設定します。たとえば、POP3 を必要としない場合は、次のように設定します。

```
protocols = imap lmtp
```

   デフォルトでは、**imap**、**pop3**、および **lmtp** プロトコルが有効になっています。

2. ローカルファイアウォールでポートを開きます。たとえば、IMAPS、IMAP、POP3S、および POP3 プロトコルのポートを開くには、次のように入力します。

```
# firewall-cmd --permanent --add-service=imaps --add-service=imap --add-
service=pop3s --add-service=pop3
# firewall-cmd --reload
```

3. **dovecot** サービスを有効にして開始します。

```
# systemctl enable --now dovecot
```

検証

1. Dovecot に接続して電子メールを読むには、Mozilla Thunderbird などのメールクライアントを使用します。メールクライアントの設定は、使用するプロトコルによって異なります。

   表1.1 Dovecot サーバーへの接続設定

プロトコル	ポート	接続セキュリティー	認証方法
IMAP	143	STARTTLS	PLAIN[a]
IMAPS	993	SSL/TLS	PLAIN[a]
POP3	110	STARTTLS	PLAIN[a]
POP3S	995	SSL/TLS	PLAIN[a]

[a] クライアントは、TLS 接続を介して暗号化されたデータを送信します。したがって、認証情報は開示されません。

   デフォルトでは、Dovecot は TLS を使用しない接続ではプレーンテキスト認証を受け入れないため、この表には暗号化されていない接続の設定がリストされていないことに注意してください。

2. デフォルト以外の値を含む設定を表示します。

```
# doveconf -n
```

関連情報

- **firewall-cmd(1)** man ページ

1.2. LDAP 認証を使用した DOVECOT サーバーのセットアップ

インフラストラクチャーが LDAP サーバーを使用してアカウントを保存している場合、それに対して Dovecot ユーザーを認証できます。この場合、アカウントをディレクトリーで集中管理するため、ユーザーは Dovecot サーバー上のファイルシステムにローカルでアクセスする必要はありません。

複数の Dovecot サーバーをレプリケーションでセットアップして、メールボックスを高可用性にする予定がある場合にも、集中管理されたアカウントは利点があります。

1.2.1. Dovecot のインストール

dovecot パッケージは以下を提供します。

- **dovecot** サービスとそれを管理するユーティリティー
- Dovecot がオンデマンドで開始するサービス (認証など)
- サーバーサイドメールフィルタリングなどのプラグイン
- **/etc/dovecot/** ディレクトリーの設定ファイル
- **/usr/share/doc/dovecot/** ディレクトリーのドキュメント

手順

- **dovecot** パッケージをインストールします。

```
# dnf install dovecot
```

注記

Dovecot がすでにインストールされていて、クリーンな設定ファイルが必要な場合は、**/etc/dovecot/** ディレクトリーを名前変更するか削除してください。その後、パッケージを再インストールします。設定ファイルを削除しないと、**dnf reinstall dovecot** コマンドは **/etc/dovecot/** 内の設定ファイルをリセットしません。

次のステップ

- Dovecot サーバーでの TLS 暗号化の設定。

1.2.2. Dovecot サーバーでの TLS 暗号化の設定

Dovecot はセキュアなデフォルト設定を提供します。たとえば、TLS はデフォルトで有効になっており、認証情報と暗号化されたデータをネットワーク経由で送信します。Dovecot サーバーで TLS を設定するには、証明書と秘密鍵ファイルへのパスを設定するだけです。さらに、Diffie-Hellman パラメーターを生成して使用し、Perfect Forward Secrecy (PFS) を提供することで、TLS 接続のセキュリティーを強化できます。

前提条件

- Dovecot がインストールされています。

- 次のファイルが、サーバー上のリストされた場所にコピーされました。
  - サーバー証明書: **/etc/pki/dovecot/certs/server.example.com.crt**
  - 秘密鍵: **/etc/pki/dovecot/private/server.example.com.key**
  - 認証局 (CA) 証明書: **/etc/pki/dovecot/certs/ca.crt**
- サーバー証明書の **Subject DN** フィールドのホスト名は、サーバーの完全修飾ドメイン名 (FQDN) と一致します。
- サーバーが RHEL 9.2 以降を実行し、FIPS モードが有効になっている場合、クライアントが Extended Master Secret (EMS) 拡張機能をサポートしているか、TLS 1.3 を使用している必要があります。EMS を使用しない TLS 1.2 接続は失敗します。詳細は、ナレッジベースの記事 TLS extension "Extended Master Secret" enforced を参照してください。

手順

1. 秘密鍵ファイルにセキュアな権限を設定します。

```
# chown root:root /etc/pki/dovecot/private/server.example.com.key
# chmod 600 /etc/pki/dovecot/private/server.example.com.key
```

2. Diffie-Hellman パラメーターを使用してファイルを生成します。

```
# openssl dhparam -out /etc/dovecot/dh.pem 4096
```

サーバーのハードウェアとエントロピーによっては、4096 ビットの Diffie-Hellman パラメーターを生成するのに数分かかる場合があります。

3. **/etc/dovecot/conf.d/10-ssl.conf** ファイルで証明書と秘密鍵ファイルへのパスを設定します。
   a. **ssl_cert** および **ssl_key** パラメーターを更新し、サーバーの証明書と秘密鍵へのパスを使用するように設定します。

```
ssl_cert = </etc/pki/dovecot/certs/server.example.com.crt
ssl_key = </etc/pki/dovecot/private/server.example.com.key
```

   b. **ssl_ca** パラメーターをコメント解除し、CA 証明書へのパスを使用するように設定します。

```
ssl_ca = </etc/pki/dovecot/certs/ca.crt
```

   c. **ssl_dh** パラメーターをコメント解除し、Diffie-Hellman パラメーターファイルへのパスを使用するように設定します。

```
ssl_dh = </etc/dovecot/dh.pem
```

重要

Dovecot がファイルからパラメーターの値を確実に読み取るようにするには、パスの先頭に **<** 文字を付ける必要があります。

次のステップ

- 仮想ユーザーを使用するための Dovecot の準備

関連情報

- **/usr/share/doc/dovecot/wiki/SSL.DovecotConfiguration.txt**

1.2.3. 仮想ユーザーを使用するための Dovecot の準備

デフォルトでは、Dovecot はサービスを使用するユーザーとして、ファイルシステム上で多くのアクションを実行します。ただし、1 人のローカルユーザーを使用してこれらのアクションを実行するように Dovecot バックエンドを設定すると、複数の利点があります。

- Dovecot は、ユーザーの ID (UID) を使用する代わりに、特定のローカルユーザーとしてファイルシステムアクションを実行します。
- ユーザーは、サーバー上でローカルに利用できる必要はありません。
- すべてのメールボックスとユーザー固有のファイルを 1 つのルートディレクトリーに保存できます。
- ユーザーは UID とグループ ID (GID) を必要としないため、管理作業が軽減されます。
- サーバー上のファイルシステムにアクセスできるユーザーは、これらのファイルにアクセスできないため、メールボックスやインデックスを危険にさらす可能性はありません。
- レプリケーションのセットアップはより簡単です。

前提条件

- Dovecot がインストールされています。

手順

1. **vmail** ユーザーを作成します。

```
# useradd --home-dir /var/mail/ --shell /usr/sbin/nologin vmail
```

   Dovecot は後でこのユーザーを使用してメールボックスを管理します。セキュリティー上の理由から、この目的で **dovecot** または **dovenull** システムユーザーを使用しないでください。

2. **/var/mail/** 以外のパスを使用する場合は、それに SELinux コンテキスト **mail_spool_t** を設定します。例:

```
# semanage fcontext -a -t mail_spool_t "<path>(/.*)?"
# restorecon -Rv <path>
```

3. **/var/mail/** への書き込み権限を **vmail** ユーザーにのみ付与します。

```
# chown vmail:vmail /var/mail/
# chmod 700 /var/mail/
```

4. **/etc/dovecot/conf.d/10-mail.conf** ファイルの **mail_location** パラメーターをコメント解除し、メールボックスの形式と場所を設定します。

```
mail_location = sdbox:/var/mail/%n/
```

この設定の場合:

- Dovecot は、**single** モードで高パフォーマンスの **dbox** メールボックス形式を使用します。このモードでは、サービスは、**maildir** 形式と同様に、各メールを個別のファイルに保存します。
- Dovecot はパス内の **%n** 変数をユーザー名に解決します。これは、各ユーザーがメールボックス用に個別のディレクトリーを持つようにするために必要です。

次のステップ

- LDAP を Dovecot 認証バックエンドとして使用する。

関連情報

- **/usr/share/doc/dovecot/wiki/VirtualUsers.txt**
- **/usr/share/doc/dovecot/wiki/MailLocation.txt**
- **/usr/share/doc/dovecot/wiki/MailboxFormat.dbox.txt**
- **/usr/share/doc/dovecot/wiki/Variables.txt**

1.2.4. LDAP を Dovecot 認証バックエンドとして使用する

通常、LDAP ディレクトリー内のユーザーは、ディレクトリーサービスに対して自分自身を認証できます。Dovecot は、これを使用して、ユーザーが IMAP または POP3 サービスにログインする場合、ユーザーを認証できます。この認証方法には、次のとおり、多くの利点があります。

- 管理者は、ディレクトリーでユーザーを集中管理できます。
- LDAP アカウントには、特別な属性は必要ありません。LDAP サーバーから認証を受けることができれば十分です。したがって、この方法は、LDAP サーバーで使用されるパスワード保存方式とは無関係です。
- ユーザーは、Name Service Switch (NSS) インターフェイスと Pluggable Authentication Module (PAM) フレームワークを介して、サーバー上でローカルに利用できる必要はありません。

前提条件

- Dovecot がインストールされています。
- 仮想ユーザー機能が設定されています。
- LDAP サーバーへの接続は、TLS 暗号化をサポートします。
- Dovecot サーバー上の RHEL は、LDAP サーバーの認証局 (CA) 証明書を信頼します。
- ユーザーが LDAP ディレクトリーの異なるツリーに保存されている場合、ディレクトリーを検索するための Dovecot 専用の LDAP アカウントが存在します。このアカウントには、他のユーザーの識別名 (DN) を検索する権限が必要です。
- MariaDB サーバーが RHEL 9.2 以降を実行し、FIPS モードが有効になっている場合、この

Dovecot サーバーは Extended Master Secret (EMS) 拡張機能をサポートするか、TLS 1.3 を使用します。EMS を使用しない TLS 1.2 接続は失敗します。詳細は、ナレッジベースの記事 TLS extension "Extended Master Secret" enforced を参照してください。

手順

1. **/etc/dovecot/conf.d/10-auth.conf** ファイルで認証バックエンドを設定します。

   a. 不要な **auth-*.conf.ext** 認証バックエンド設定ファイルの **include** ステートメントをコメントアウトします。次に例を示します。

   ```
   #!include auth-system.conf.ext
   ```

   b. 次の行をコメント解除して、LDAP 認証を有効にします。

   ```
   !include auth-ldap.conf.ext
   ```

2. **/etc/dovecot/conf.d/auth-ldap.conf.ext** ファイルを編集し、次のように **override_fields** パラメーターを **userdb** セクションに追加します。

   ```
   userdb {
     driver = ldap
     args = /etc/dovecot/dovecot-ldap.conf.ext
     override_fields = uid=vmail gid=vmail home=/var/mail/%n/
   }
   ```

   固定値のため、Dovecot は LDAP サーバーからこれらの設定をクエリーしません。したがって、これらの属性も存在する必要はありません。

3. 次の設定で **/etc/dovecot/dovecot-ldap.conf.ext** ファイルを作成します。

   a. LDAP 構造に応じて、次のいずれかを設定します。

   - ユーザーが LDAP ディレクトリーの異なるツリーに保存されている場合は、動的 DN 検索を設定します。

     ```
     dn = cn=dovecot_LDAP,dc=example,dc=com
     dnpass = password
     pass_filter = (&(objectClass=posixAccount)(uid=%n))
     ```

     Dovecot は、指定された DN、パスワード、およびフィルターを使用して、ディレクトリー内の認証ユーザーの DN を検索します。この検索では、Dovecot はフィルター内の **%n** をユーザー名に置き換えます。LDAP 検索で返される結果は 1 つだけであることに注意してください。

   - すべてのユーザーが特定のエントリーに保存されている場合は、DN テンプレートを設定します。

     ```
     auth_bind_userdn = cn=%n,ou=People,dc=example,dc=com
     ```

   b. LDAP サーバーへの認証バインドを有効にして、Dovecot ユーザーを確認します。

   ```
   auth_bind = yes
   ```

c. URL を LDAP サーバーに設定します。

```
uris = ldaps://LDAP-srv.example.com
```

セキュリティー上の理由から、LDAP プロトコル上で LDAPS または **STARTTLS** コマンドを使用した暗号化された接続のみを使用してください。後者の場合は、さらに **tls = yes** を設定に追加します。

証明書の検証を機能させるには、LDAP サーバーのホスト名が TLS 証明書で使用されているホスト名と一致する必要があります。

d. LDAP サーバーの TLS 証明書の検証を有効にします。

```
tls_require_cert = hard
```

e. ベース DN には、ユーザーの検索を開始する DN を設定します。

```
base = ou=People,dc=example,dc=com
```

f. 検索範囲を設定します。

```
scope = onelevel
```

Dovecot は、指定されたベース DN のみを **onelevel** スコープで検索し、サブツリーも **subtree** スコープで検索します。

4. **/etc/dovecot/dovecot-ldap.conf.ext** ファイルにセキュアな権限を設定します。

```
# chown root:root /etc/dovecot/dovecot-ldap.conf.ext
# chmod 600 /etc/dovecot/dovecot-ldap.conf.ext
```

次のステップ

- Dovecot 設定を完了します。

関連情報

- **/usr/share/doc/dovecot/example-config/dovecot-ldap.conf.ext**
- **/usr/share/doc/dovecot/wiki/UserDatabase.Static.txt**
- **/usr/share/doc/dovecot/wiki/AuthDatabase.LDAP.txt**
- **/usr/share/doc/dovecot/wiki/AuthDatabase.LDAP.AuthBinds.txt**
- **/usr/share/doc/dovecot/wiki/AuthDatabase.LDAP.PasswordLookups.txt**

1.2.5. Dovecot 設定の完了

Dovecot をインストールして設定したら、**firewalld** サービスで必要なポートを開き、サービスを有効にして開始します。その後、サーバーをテストできます。

前提条件

- 以下は Dovecot で設定されています。
  - TLS 暗号化
  - 認証バックエンド
- クライアントは認証局 (CA) 証明書を信頼します。

手順

1. IMAP または POP3 サービスのみをユーザーに提供する場合は、**/etc/dovecot/dovecot.conf** ファイルの **protocol** パラメーターをコメント解除し、必要なプロトコルに設定します。たとえば、POP3 を必要としない場合は、次のように設定します。

   ```
   protocols = imap lmtp
   ```

   デフォルトでは、**imap**、**pop3**、および **lmtp** プロトコルが有効になっています。

2. ローカルファイアウォールでポートを開きます。たとえば、IMAPS、IMAP、POP3S、および POP3 プロトコルのポートを開くには、次のように入力します。

   ```
   # firewall-cmd --permanent --add-service=imaps --add-service=imap --add-service=pop3s --add-service=pop3
   # firewall-cmd --reload
   ```

3. **dovecot** サービスを有効にして開始します。

   ```
   # systemctl enable --now dovecot
   ```

検証

1. Dovecot に接続して電子メールを読むには、Mozilla Thunderbird などのメールクライアントを使用します。メールクライアントの設定は、使用するプロトコルによって異なります。

   表1.2 Dovecot サーバーへの接続設定

プロトコル	ポート	接続セキュリティー	認証方法
IMAP	143	STARTTLS	PLAIN[a]
IMAPS	993	SSL/TLS	PLAIN[a]
POP3	110	STARTTLS	PLAIN[a]
POP3S	995	SSL/TLS	PLAIN[a]

[a] クライアントは、TLS 接続を介して暗号化されたデータを送信します。したがって、認証情報は開示されません。

デフォルトでは、Dovecot は TLS を使用しない接続ではプレーンテキスト認証を受け入れないため、この表には暗号化されていない接続の設定がリストされていないことに注意してください。

2. デフォルト以外の値を含む設定を表示します。

```
# doveconf -n
```

関連情報

- **firewall-cmd(1)** man ページ

1.3. MARIADB SQL 認証を使用した DOVECOT サーバーのセットアップ

ユーザーとパスワードを MariaDB SQL サーバーに保存する場合、それをユーザーデータベースと認証バックエンドとして使用するように、Dovecot を設定できます。この設定では、アカウントをデータベースで集中管理するため、ユーザーは Dovecot サーバー上のファイルシステムにローカルアクセスできません。

複数の Dovecot サーバーをレプリケーションでセットアップして、メールボックスを高可用性にする予定がある場合にも、集中管理されたアカウントは利点があります。

1.3.1. Dovecot のインストール

dovecot パッケージは以下を提供します。

- **dovecot** サービスとそれを管理するユーティリティー
- Dovecot がオンデマンドで開始するサービス (認証など)
- サーバーサイドメールフィルタリングなどのプラグイン
- **/etc/dovecot/** ディレクトリーの設定ファイル
- **/usr/share/doc/dovecot/** ディレクトリーのドキュメント

手順

- **dovecot** パッケージをインストールします。

```
# dnf install dovecot
```

注記

Dovecot がすでにインストールされていて、クリーンな設定ファイルが必要な場合は、**/etc/dovecot/** ディレクトリーを名前変更するか削除してください。その後、パッケージを再インストールします。設定ファイルを削除しないと、**dnf reinstall dovecot** コマンドは **/etc/dovecot/** 内の設定ファイルをリセットしません。

次のステップ

- Dovecot サーバーでの TLS 暗号化の設定。

1.3.2. Dovecot サーバーでの TLS 暗号化の設定

Dovecot はセキュアなデフォルト設定を提供します。たとえば、TLS はデフォルトで有効になっており、認証情報と暗号化されたデータをネットワーク経由で送信します。Dovecot サーバーで TLS を設定するには、証明書と秘密鍵ファイルへのパスを設定するだけです。さらに、Diffie-Hellman パラメーターを生成して使用し、Perfect Forward Secrecy (PFS) を提供することで、TLS 接続のセキュリティーを強化できます。

前提条件

- Dovecot がインストールされています。
- 次のファイルが、サーバー上のリストされた場所にコピーされました。
  - サーバー証明書: **/etc/pki/dovecot/certs/server.example.com.crt**
  - 秘密鍵: **/etc/pki/dovecot/private/server.example.com.key**
  - 認証局 (CA) 証明書: **/etc/pki/dovecot/certs/ca.crt**
- サーバー証明書の **Subject DN** フィールドのホスト名は、サーバーの完全修飾ドメイン名 (FQDN) と一致します。
- サーバーが RHEL 9.2 以降を実行し、FIPS モードが有効になっている場合、クライアントが Extended Master Secret (EMS) 拡張機能をサポートしているか、TLS 1.3 を使用している必要があります。EMS を使用しない TLS 1.2 接続は失敗します。詳細は、ナレッジベースの記事 TLS extension "Extended Master Secret" enforced を参照してください。

手順

1. 秘密鍵ファイルにセキュアな権限を設定します。

```
# chown root:root /etc/pki/dovecot/private/server.example.com.key
# chmod 600 /etc/pki/dovecot/private/server.example.com.key
```

2. Diffie-Hellman パラメーターを使用してファイルを生成します。

```
# openssl dhparam -out /etc/dovecot/dh.pem 4096
```

   サーバーのハードウェアとエントロピーによっては、4096 ビットの Diffie-Hellman パラメーターを生成するのに数分かかる場合があります。

3. **/etc/dovecot/conf.d/10-ssl.conf** ファイルで証明書と秘密鍵ファイルへのパスを設定します。
   a. **ssl_cert** および **ssl_key** パラメーターを更新し、サーバーの証明書と秘密鍵へのパスを使用するように設定します。

```
ssl_cert = </etc/pki/dovecot/certs/server.example.com.crt
ssl_key = </etc/pki/dovecot/private/server.example.com.key
```

   b. **ssl_ca** パラメーターをコメント解除し、CA 証明書へのパスを使用するように設定します。

```
ssl_ca = </etc/pki/dovecot/certs/ca.crt
```

c. **ssl_dh** パラメーターをコメント解除し、Diffie-Hellman パラメーターファイルへのパスを使用するように設定します。

```
ssl_dh = </etc/dovecot/dh.pem
```

重要

Dovecot がファイルからパラメーターの値を確実に読み取るようにするには、パスの先頭に **<** 文字を付ける必要があります。

次のステップ

- 仮想ユーザーを使用するための Dovecot の準備

関連情報

- **/usr/share/doc/dovecot/wiki/SSL.DovecotConfiguration.txt**

1.3.3. 仮想ユーザーを使用するための Dovecot の準備

デフォルトでは、Dovecot はサービスを使用するユーザーとして、ファイルシステム上で多くのアクションを実行します。ただし、1 人のローカルユーザーを使用してこれらのアクションを実行するように Dovecot バックエンドを設定すると、複数の利点があります。

- Dovecot は、ユーザーの ID (UID) を使用する代わりに、特定のローカルユーザーとしてファイルシステムアクションを実行します。
- ユーザーは、サーバー上でローカルに利用できる必要はありません。
- すべてのメールボックスとユーザー固有のファイルを 1 つのルートディレクトリーに保存できます。
- ユーザーは UID とグループ ID (GID) を必要としないため、管理作業が軽減されます。
- サーバー上のファイルシステムにアクセスできるユーザーは、これらのファイルにアクセスできないため、メールボックスやインデックスを危険にさらす可能性はありません。
- レプリケーションのセットアップはより簡単です。

前提条件

- Dovecot がインストールされています。

手順

1. **vmail** ユーザーを作成します。

```
# useradd --home-dir /var/mail/ --shell /usr/sbin/nologin vmail
```

Dovecot は後でこのユーザーを使用してメールボックスを管理します。セキュリティー上の理由から、この目的で **dovecot** または **dovenull** システムユーザーを使用しないでください。

2. **/var/mail/** 以外のパスを使用する場合は、それに SELinux コンテキスト **mail_spool_t** を設定します。例:

```
# semanage fcontext -a -t mail_spool_t "<path>(/.*)?"
# restorecon -Rv <path>
```

3. **/var/mail/** への書き込み権限を **vmail** ユーザーにのみ付与します。

```
# chown vmail:vmail /var/mail/
# chmod 700 /var/mail/
```

4. **/etc/dovecot/conf.d/10-mail.conf** ファイルの **mail_location** パラメーターをコメント解除し、メールボックスの形式と場所を設定します。

```
mail_location = sdbox:/var/mail/%n/
```

この設定の場合:

- Dovecot は、**single** モードで高パフォーマンスの **dbox** メールボックス形式を使用します。このモードでは、サービスは、**maildir** 形式と同様に、各メールを個別のファイルに保存します。
- Dovecot はパス内の **%n** 変数をユーザー名に解決します。これは、各ユーザーがメールボックス用に個別のディレクトリーを持つようにするために必要です。

次のステップ

- Dovecot 認証バックエンドとして MariaDB SQL データベースを使用する

関連情報

- **/usr/share/doc/dovecot/wiki/VirtualUsers.txt**
- **/usr/share/doc/dovecot/wiki/MailLocation.txt**
- **/usr/share/doc/dovecot/wiki/MailboxFormat.dbox.txt**
- **/usr/share/doc/dovecot/wiki/Variables.txt**

1.3.4. Dovecot 認証バックエンドとして MariaDB SQL データベースを使用する

Dovecot は、MariaDB データベースからアカウントとパスワードを読み取り、これを使用して、ユーザーが IMAP または POP3 サービスにログインする場合、ユーザーを認証できます。この認証方法の利点は次のとおりです。

- 管理者は、データベースでユーザーを集中管理できます。
- ユーザーはサーバー上でローカルにアクセスできません。

前提条件

- Dovecot がインストールされています。
- 仮想ユーザー機能が設定されています。
- MariaDB サーバーへの接続では、TLS 暗号化がサポートされます。

- **dovecotDB** データベースは MariaDB に存在し、**users** テーブルには、少なくとも **username** および **password** 列が含まれています。
- **password** 列には、Dovecot がサポートするスキームで暗号化されたパスワードが含まれています。
- パスワードは、同じスキームを使用するか、**{pw-storage-scheme}** 接頭辞を使用します。
- MariaDB ユーザー **dovecot** は、**dovecotDB** データベースの **users** テーブルに対する読み取り権限を持っています。
- MariaDB サーバーの TLS 証明書を発行した認証局 (CA) の証明書は、Dovecot サーバーの **/etc/pki/tls/certs/ca.crt** ファイルに保存されます。
- MariaDB サーバーが RHEL 9.2 以降を実行し、FIPS モードが有効になっている場合、この Dovecot サーバーは Extended Master Secret (EMS) 拡張機能をサポートするか、TLS 1.3 を使用します。EMS を使用しない TLS 1.2 接続は失敗します。詳細は、ナレッジベースの記事 TLS extension "Extended Master Secret" enforced を参照してください。

手順

1. **dovecot-mysql** パッケージをインストールします。

```
# dnf install dovecot-mysql
```

2. **/etc/dovecot/conf.d/10-auth.conf** ファイルで認証バックエンドを設定します。
   a. 不要な **auth-*.conf.ext** 認証バックエンド設定ファイルの **include** ステートメントをコメントアウトします。次に例を示します。

```
#!include auth-system.conf.ext
```

   b. 次の行をコメント解除して、SQL 認証を有効にします。

```
!include auth-sql.conf.ext
```

3. **/etc/dovecot/conf.d/auth-sql.conf.ext** ファイルを編集し、**override_fields** パラメーターを **userdb** セクションに次のように追加します。

```
userdb {
  driver = sql
  args = /etc/dovecot/dovecot-sql.conf.ext
  override_fields = uid=vmail gid=vmail home=/var/mail/%n/
}
```

   固定値のため、Dovecot はこれらの設定を SQL サーバーからクエリーしません。

4. 次の設定で **/etc/dovecot/dovecot-sql.conf.ext** ファイルを作成します。

```
driver = mysql
connect = host=mariadb_srv.example.com dbname=dovecotDB user=dovecot
password=dovecotPW ssl_ca=/etc/pki/tls/certs/ca.crt
default_pass_scheme = SHA512-CRYPT
user_query = SELECT username FROM users WHERE username='%u';
```

```
password_query = SELECT username AS user, password FROM users WHERE
username='%u';
iterate_query = SELECT username FROM users;
```

データベースサーバーに対して TLS 暗号化を使用するには、**ssl_ca** オプションに MariaDB サーバー証明書を発行した CA の証明書のパスを設定します。証明書の検証を機能させるには、MariaDB サーバーのホスト名が TLS 証明書で使用されているホスト名と一致する必要があります。

データベースのパスワード値に **{pw-storage-scheme}** 接頭辞が含まれている場合は、**default_pass_scheme** 設定を省略できます。

ファイル内のクエリーは、次のように設定する必要があります。

- **user_query** パラメーターの場合、クエリーは Dovecot ユーザーのユーザー名を返す必要があります。また、クエリーは 1 つの結果のみを返す必要があります。
- **password_query** パラメーターの場合、クエリーはユーザー名とパスワードを返す必要があり、Dovecot は **user** および **password** 変数でこれらの値を使用する必要があります。したがって、データベースが異なる列名を使用している場合は、**AS** SQL コマンドを使用して、結果の列の名前を変更してください。
- **iterate_query** パラメーターの場合、クエリーはすべてのユーザーのリストを返す必要があります。

5. **/etc/dovecot/dovecot-sql.conf.ext** ファイルにセキュアな権限を設定します。

```
# chown root:root /etc/dovecot/dovecot-sql.conf.ext
# chmod 600 /etc/dovecot/dovecot-sql.conf.ext
```

次のステップ

- Dovecot 設定を完了します。

関連情報

- **/usr/share/doc/dovecot/example-config/dovecot-sql.conf.ext**
- **/usr/share/doc/dovecot/wiki/Authentication.PasswordSchemes.txt**

1.3.5. Dovecot 設定の完了

Dovecot をインストールして設定したら、**firewalld** サービスで必要なポートを開き、サービスを有効にして開始します。その後、サーバーをテストできます。

前提条件

- 以下は Dovecot で設定されています。
  - TLS 暗号化
  - 認証バックエンド
- クライアントは認証局 (CA) 証明書を信頼します。

手順

手順

1. IMAP または POP3 サービスのみをユーザーに提供する場合は、**/etc/dovecot/dovecot.conf** ファイルの **protocol** パラメーターをコメント解除し、必要なプロトコルに設定します。たとえば、POP3 を必要としない場合は、次のように設定します。

```
protocols = imap lmtp
```

   デフォルトでは、**imap**、**pop3**、および **lmtp** プロトコルが有効になっています。

2. ローカルファイアウォールでポートを開きます。たとえば、IMAPS、IMAP、POP3S、および POP3 プロトコルのポートを開くには、次のように入力します。

```
# firewall-cmd --permanent --add-service=imaps --add-service=imap --add-service=pop3s --add-service=pop3
# firewall-cmd --reload
```

3. **dovecot** サービスを有効にして開始します。

```
# systemctl enable --now dovecot
```

検証

1. Dovecot に接続して電子メールを読むには、Mozilla Thunderbird などのメールクライアントを使用します。メールクライアントの設定は、使用するプロトコルによって異なります。

   表1.3 Dovecot サーバーへの接続設定

プロトコル	ポート	接続セキュリティー	認証方法
IMAP	143	STARTTLS	PLAIN[a]
IMAPS	993	SSL/TLS	PLAIN[a]
POP3	110	STARTTLS	PLAIN[a]
POP3S	995	SSL/TLS	PLAIN[a]

[a] クライアントは、TLS 接続を介して暗号化されたデータを送信します。したがって、認証情報は開示されません。

   デフォルトでは、Dovecot は TLS を使用しない接続ではプレーンテキスト認証を受け入れないため、この表には暗号化されていない接続の設定がリストされていないことに注意してください。

2. デフォルト以外の値を含む設定を表示します。

```
# doveconf -n
```

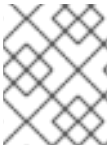
関連情報

- **firewall-cmd(1)** man ページ

1.4. 2 つの DOVECOT サーバー間のレプリケーションの設定

双方向のレプリケーションを使用すると、Dovecot サーバーを高可用性にすることができ、IMAP および POP3 クライアントは両方のサーバーのメールボックスにアクセスできます。Dovecot は、各メールボックスのインデックスログの変更を追跡し、競合を安全な方法で解決します。

両方の複製パートナーでこの手順を実行します。



注記

レプリケーションは、サーバーペア間でのみ機能します。したがって、大規模なクラスターでは、複数の独立したバックエンドペアが必要になります。

前提条件

- 両方のサーバーが同じ認証バックエンドを使用します。できれば、LDAP または SQL を使用して、アカウントを集中管理してください。
- Dovecot ユーザーデータベース設定は、ユーザーリストをサポートします。これを確認するには、**doveadm user '*'** コマンドを使用します。
- Dovecot は、ユーザー ID (UID) ではなく、**vmail** ユーザーとしてファイルシステム上のメールボックスにアクセスします。

手順

1. **/etc/dovecot/conf.d/10-replication.conf** ファイルを作成し、その中で次の手順を実行します。
   a. **notify** および **replication** プラグインを有効にします。

```
mail_plugins = $mail_plugins notify replication
```

   b. **service replicator** セクションを追加します。

```
service replicator {
  process_min_avail = 1

  unix_listener replicator-doveadm {
    mode = 0600
    user = vmail
  }
}
```

   これらの設定により、**dovecot** サービスの開始時に、Dovecot は 1 つ以上のレプリケータープロセスを開始します。さらに、このセクションは **replicator-doveadm** ソケットの設定を定義します。

   c. **service aggregator** セクションを追加して、**replication-notify-fifo** パイプと **replication-notify** ソケットを設定します。

```
service aggregator {
  fifo_listener replication-notify-fifo {
```

```
    user = vmail
  }
  unix_listener replication-notify {
    user = vmail
  }
}
```

d. **service doveadm** セクションを追加して、レプリケーションサービスのポートを定義します。

```
service doveadm {
  inet_listener {
    port = 12345
  }
}
```

e. **doveadm** レプリケーションサービスのパスワードを設定します。

```
doveadm_password = replication_password
```

パスワードは、両方のサーバーで同じにする必要があります。

f. レプリケーションパートナーを設定します。

```
plugin {
  mail_replica = tcp:server2.example.com:12345
}
```

g. オプション: 並列 **dsync** プロセスの最大数を定義します。

```
replication_max_conns = 20
```

replication_max_conns のデフォルト値は **10** です。

2. **/etc/dovecot/conf.d/10-replication.conf** ファイルにセキュアな権限を設定します。

```
# chown root:root /etc/dovecot/conf.d/10-replication.conf
# chmod 600 /etc/dovecot/conf.d/10-replication.conf
```

3. Dovecot が **doveadm** レプリケーションポートを開くことができるように、SELinux ブール値 **nis_enabled** を有効にします。

```
setsebool -P nis_enabled on
```

4. レプリケーションパートナーのみがレプリケーションポートにアクセスできるように、**firewalld** ルールを設定します。次に例を示します。

```
# firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv4" source
address="192.0.2.1/32" port protocol="tcp" port="12345" accept"
# firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv6" source
address="2001:db8:2::1/128" port protocol="tcp" port="12345" accept"
# firewall-cmd --reload
```

IPv4 アドレスのサブネットマスク **/32** と IPv6 アドレスのサブネットマスク **/128** は、指定されたアドレスへのアクセスを制限します。

5. この手順は、他のレプリケーションパートナーでも実行します。

6. Dovecot をリロードします。

```
# systemctl reload dovecot
```

検証

1. 1 つのサーバーのメールボックスでアクションを実行し、Dovecot が変更を他のサーバーにレプリケートしたかどうかを確認します。

2. レプリケーターステータスを表示します。

```
# doveadm replicator status
Queued 'sync' requests       0
Queued 'high' requests       0
Queued 'low' requests        0
Queued 'failed' requests     0
Queued 'full resync' requests 30
Waiting 'failed' requests    0
Total number of known users  75
```

3. 特定のユーザーのレプリケーターステータスを表示します。

```
# doveadm replicator status example_user
username        priority  fast sync  full sync  success sync  failed
example_user    none      02:05:28   04:19:07   02:05:28      -
```

関連情報

- **dsync(1)** man ページ
- **/usr/share/doc/dovecot/wiki/Replication.txt**

1.5. ユーザーを IMAP メールボックスに自動的に登録する

通常、IMAP サーバー管理者は、Dovecot が **Sent** や **Trash** などの特定のメールボックスを自動的に作成し、ユーザーをそれらに登録することを望んでいます。これは設定ファイルに設定できます。

さらに、**特殊用途のメールボックス** を定義できます。多くの場合、IMAP クライアントは、メールの送信など、特別な目的のためにメールボックスを定義することをサポートしています。ユーザーが正しいメールボックスを手動で選択して設定する必要がないようにするために、IMAP サーバーは IMAP **LIST** コマンドで **special-use** 属性を送信できます。その後、クライアントはこの属性を使用して、送信済みメールのメールボックスなどを識別および設定できます。

前提条件

- Dovecot が設定されている。

手順

1. **/etc/dovecot/conf.d/15-mailboxes.conf** ファイルの **inbox** namespace セクションを更新します。

   a. **auto = subscribe** 設定を、ユーザーが利用できるようにする必要がある各特殊用途のメールボックスに追加します。次に例を示します。

```
namespace inbox {
  ...
  mailbox Drafts {
    special_use = \Drafts
    auto = subscribe
  }

  mailbox Junk {
    special_use = \Junk
    auto = subscribe
  }

  mailbox Trash {
    special_use = \Trash
    auto = subscribe
  }

  mailbox Sent {
    special_use = \Sent
    auto = subscribe
  }
  ...
}
```

   メールクライアントがより特殊用途のメールボックスをサポートしている場合は、同様のエントリーを追加できます。**special_use** パラメーターは、Dovecot が **special-use** 属性でクライアントに送信する値を定義します。

   b. オプション: 特別な目的のない他のメールボックスを定義する場合は、ユーザーの受信トレイにそれらの **mailbox** セクションを追加します。次に例を示します。

```
namespace inbox {
  ...
  mailbox "Important Emails" {
    auto = <value>
  }
  ...
}
```

   auto パラメーターは、次のいずれかの値に設定できます。

   - **subscribe**: メールボックスを自動的に作成し、ユーザーを登録します。
   - **create**: ユーザーを登録せずに、メールボックスを自動的に作成します。
   - **no** (デフォルト): Dovecot はメールボックスを作成することも、ユーザーを登録することもしません。

2. Dovecot をリロードします。

```
# systemctl reload dovecot
```

検証

- IMAP クライアントを使用してメールボックスにアクセスします。
  auto = subscribe が設定されたメールボックスは、自動的に表示されます。クライアントが特殊用途のメールボックスと定義された目的をサポートしている場合、クライアントはそれらを自動的に使用します。

関連情報

- RFC 6154: 特殊用途メールボックスの IMAP LIST 拡張
- **/usr/share/doc/dovecot/wiki/MailboxSettings.txt**

1.6. LMTP ソケットと LMTPS リスナーの設定

Postfix などの SMTP サーバーは、Local Mail Transfer Protocol (LMTP) を使用して電子メールを Dovecot に配信します。SMTP サーバーが実行されている場合:

- Dovecot と同じホストで、LMTP ソケットを使用します。
- 別のホストで、LMTP サービスを使用する
  デフォルトでは、LMTP プロトコルは暗号化されていません。ただし、TLS 暗号化を設定した場合、Dovecot は LMTP サービスに同じ設定を自動的に使用します。その後、SMTP サーバーは、LMTPS プロトコルまたは LMTP 上の **STARTTLS** コマンドを使用して接続できます。

前提条件

- Dovecot がインストールされています。
- LMTP サービスを設定する場合、Dovecot で TLS 暗号化が設定されます。

手順

1. LMTP プロトコルが有効になっていることを確認します。

   ```
   # doveconf -a | egrep "^protocols"
   protocols = imap pop3 lmtp
   ```

   出力に **lmtp** が含まれている場合、プロトコルは有効になっています。

2. **lmtp** プロトコルが無効になっている場合は、**/etc/dovecot/dovecot.conf** ファイルを編集し、**protocols** パラメーターの値に **lmtp** を追加します。

   ```
   protocols = ... lmtp
   ```

3. LMTP ソケットまたはサービスが必要かどうかに応じて、**/etc/dovecot/conf.d/10-master.conf** ファイルの **service lmtp** セクションで次の変更を行います。

   - LMTP ソケット: デフォルトでは、Dovecot は自動的に **/var/run/dovecot/lmtp** ソケットを作成します。
     オプション: 所有権と権限をカスタマイズします。

     ■

```
service lmtp {
  ...
  unix_listener lmtp {
    mode = 0600
    user = postfix
    group = postfix
  }
  ...
}
```

- LMTP サービス: **inet_listener** サブセクションを追加します。

```
service lmtp {
  ...
  inet_listener lmtp {
    port = 24
  }
  ...
}
```

4. SMTP サーバーのみが LMTP ポートにアクセスできるように、**firewalld** ルールを設定します。次に例を示します。

```
# firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv4" source
address="192.0.2.1/32" port protocol="tcp" port="24" accept"
# firewall-cmd --permanent --zone=public --add-rich-rule="rule family="ipv6" source
address="2001:db8:2::1/128" port protocol="tcp" port="24" accept"
# firewall-cmd --reload
```

IPv4 アドレスのサブネットマスク **/32** と IPv6 アドレスのサブネットマスク **/128** は、指定されたアドレスへのアクセスを制限します。

5. Dovecot をリロードします。

```
# systemctl reload dovecot
```

検証

1. LMTP ソケットを設定した場合は、Dovecot がソケットを作成したことと、権限が正しいことを確認します。

```
# ls -l /var/run/dovecot/lmtp
srw-------. 1 postfix postfix 0 Nov 22 17:17 /var/run/dovecot/lmtp
```

2. LMTP ソケットまたはサービスを使用して、Dovecot に電子メールを送信するように、SMTP サーバーを設定します。
LMTP サービスを使用する場合は、SMTP サーバーが LMTPS プロトコルを使用するか、**STARTTLS** コマンドを送信して暗号化された接続を使用するようにしてください。

関連情報

- **/usr/share/doc/dovecot/wiki/LMTP.txt**

1.7. DOVECOT で IMAP または POP3 サービスを無効にする

デフォルトでは、Dovecot は IMAP および POP3 サービスを提供します。そのうちの 1 つだけが必要な場合は、もう 1 つを無効にして、攻撃サーフェスを減らすことができます。

前提条件

- Dovecot がインストールされています。

手順

1. **/etc/dovecot/dovecot.conf** ファイルの **protocols** パラメーターをコメント解除し、必要なプロトコルを使用するように設定します。たとえば、POP3 を必要としない場合は、次のように設定します。

```
protocols = imap lmtp
```

   デフォルトでは、**imap**、**pop3**、および **lmtp** プロトコルが有効になっています。

2. Dovecot をリロードします。

```
# systemctl reload dovecot
```

3. ローカルファイアウォールで不要になったポートを閉じます。たとえば、POP3S および POP3 プロトコルのポートを閉じるには、次のように入力します。

```
# firewall-cmd --remove-service=pop3s --remove-service=pop3
# firewall-cmd --reload
```

検証

- **dovecot** プロセスによって開かれた **LISTEN** モードのすべてのポートを表示します。

```
# ss -tulp | grep dovecot
tcp  LISTEN 0  100  0.0.0.0:993  0.0.0.0:*  users:(("dovecot",pid=1405,fd=44))
tcp  LISTEN 0  100  0.0.0.0:143  0.0.0.0:*  users:(("dovecot",pid=1405,fd=42))
tcp  LISTEN 0  100     [::]:993     [::]:*  users:(("dovecot",pid=1405,fd=45))
tcp  LISTEN 0  100     [::]:143     [::]:*  users:(("dovecot",pid=1405,fd=43))
```

   この例では、Dovecot は TCP ポート **993** (IMAPS) と **143** (IMAP) のみをリッスンします。

   ソケットを使用する代わりにポートをリッスンするようにサービスを設定した場合、Dovecot は LMTP プロトコルのポートのみを開くことに注意してください。

関連情報

- **firewall-cmd(1)** man ページ

1.8. DOVECOT IMAP サーバーで SIEVE を使用してサーバーサイドメールフィルタリングを有効にする

ManageSieve プロトコルを使用して、Sieve スクリプトをサーバーにアップロードできます。Sieve スクリプトは、受信メールに対してサーバーが検証して実行するルールとアクションを定義します。たと

えば、ユーザーは Sieve を使用して特定の送信者からの電子メールを転送でき、管理者はグローバルフィルターを作成して、スパムフィルターによってフラグが付けられたメールを別の IMAP フォルダーに移動できます。

ManageSieve プラグインは、Sieve スクリプトと ManageSieve プロトコルのサポートを Dovecot IMAP サーバーに追加します。

警告

TLS 接続を介した ManageSieve プロトコルの使用をサポートするクライアントのみを使用してください。このプロトコルの TLS を無効にすると、クライアントはネットワーク経由で認証情報をプレーンテキストで送信します。

前提条件

- Dovecot が設定され、IMAP メールボックスを提供します。
- TLS 暗号化は Dovecot で設定されます。
- メールクライアントは、TLS 接続を介して ManageSieve プロトコルをサポートします。

手順

1. **dovecot-pigeonhole** パッケージをインストールします。

```
# dnf install dovecot-pigeonhole
```

2. **/etc/dovecot/conf.d/20-managesieve.conf** の次の行をコメント解除して、**sieve** プロトコルを有効にします。

```
protocols = $protocols sieve
```

この設定により、すでに有効になっている他のプロトコルに加えて、Sieve が有効になります。

3. **firewalld** で ManageSieve ポートを開きます。

```
# firewall-cmd --permanent --add-service=managesieve
# firewall-cmd --reload
```

4. Dovecot をリロードします。

```
# systemctl reload dovecot
```

検証

1. クライアントを使用し、Sieve スクリプトをアップロードします。次の接続設定を使用します。

- ポート: 4190
- 接続セキュリティー: SSL/TLS
- 認証方法: PLAIN

2. Sieve スクリプトをアップロードしたユーザーに電子メールを送信します。電子メールがスクリプトのルールと一致する場合は、サーバーが定義されたアクションを実行することを確認します。

関連情報

- **/usr/share/doc/dovecot/wiki/Pigeonhole.Sieve.Plugins.IMAPSieve.txt**
- **/usr/share/doc/dovecot/wiki/Pigeonhole.Sieve.Troubleshooting.txt**
- **firewall-cmd(1)** man ページ

1.9. DOVECOT が設定ファイルを処理する方法

dovecot パッケージは、メインの設定ファイル **/etc/dovecot/dovecot.conf**、および **/etc/dovecot/conf.d/** ディレクトリー内の複数の設定ファイルを提供します。Dovecot は、サービスの開始時にファイルを組み合わせて設定を構築します。

複数の設定ファイルの主な利点は、設定をグループ化し、読みやすくすることです。単一の設定ファイルを使用する場合は、代わりに **/etc/dovecot/dovecot.conf** ですべての設定を維持し、そのファイルからすべての **include** および **include_try** ステートメントを削除できます。

関連情報

- **/usr/share/doc/dovecot/wiki/ConfigFile.txt**
- **/usr/share/doc/dovecot/wiki/Variables.txt**

第2章 POSTFIX SMTP サーバーのデプロイと設定

システム管理者は、Postfix などのメールトランスポートエージェント (MTA) を使用して、電子メールインフラストラクチャーを設定し、SMTP プロトコルを使用してホスト間で電子メールメッセージを転送できます。Postfix は、メールのルーティングと配信を行うサーバー側アプリケーションです。Postfix を使用して、ローカルメールサーバーの設定、null クライアントメールリレーの作成、複数のドメインの宛先としての Postfix サーバーの使用、検索用ファイルに代わる LDAP ディレクトリーの選択を行うことができます。

Postfix の主な機能:

- 一般的なメール関連の脅威から保護するためのセキュリティー機能
- 仮想ドメインおよびエイリアスのサポートを含むカスタマイズオプション

2.1. 主な POSTFIX 設定ファイルの概要

postfix パッケージは、**/etc/postfix/** ディレクトリーに複数の設定ファイルを提供します。

電子メールインフラストラクチャーを設定するには、次の設定ファイルを使用します。

- **main.cf** – Postfix のグローバル設定が含まれています。
- **master.cf** – メール配信を実現するために、さまざまなプロセスとの Postfix の対話を指定します。
- **access** – Postfix に接続できるホストなどのアクセスルールを指定します。
- **transport** – 電子メールアドレスをリレーホストにマッピングします。
- **aliases** – ユーザー ID エイリアスを説明するメールプロトコルで必要な設定可能な一覧が含まれます。このファイルは、**/etc/** ディレクトリーにあることに留意してください。

2.2. POSTFIX SMTP サーバーのインストールおよび設定

電子メールメッセージを受信、保存、配信するように Postfix SMTP サーバーを設定できます。システムのインストール時にメールサーバーパッケージが選択されていない場合、Postfix はデフォルトで利用できません。Postfix をインストールするには、以下の手順を実行します。

前提条件

- root アクセスがある。
- システムを登録する。
- Sendmail を無効にして削除するには、以下を実行します。

```
# dnf remove sendmail
```

手順

1. Postfix をインストールします。

```
# dnf install postfix
```

2. Postfix を設定するには、**/etc/postfix/main.cf** ファイルを編集し、以下の変更を加えます。

   a. デフォルトでは、Postfix は **loopback** インターフェイスでのみメールを受信します。特定のインターフェイスをリッスンするように Postfix を設定するには、**inet_interfaces** パラメーターをこれらのインターフェイスの IP アドレスに更新します。

   ```
   inet_interfaces = 127.0.0.1/32, [::1]/128, 192.0.2.1, [2001:db8:1::1]
   ```

   すべてのインターフェイスをリッスンするように Postfix を設定するには、以下を設定します。

   ```
   inet_interfaces = all
   ```

   b. **gethostname()** 関数によって返される完全修飾ドメイン名 (FQDN) とは異なるホスト名を Postfix が使用するようにしたい場合は、**myhostname** パラメーターを追加します。

   ```
   myhostname = <smtp.example.com>
   ```

   たとえば、Postfix はこのホスト名を、処理するメールのヘッダーに追加します。

   c. ドメイン名が **myhostname** パラメーターのものと異なる場合は、**mydomain** パラメーターを追加します。

   ```
   mydomain = <example.com>
   ```

   d. **myorigin** パラメーターを追加し、**mydomain** の値に設定します。

   ```
   myorigin = $mydomain
   ```

   この設定では、Postfix はホスト名ではなく、ローカルで投稿されたメールの発信元としてドメイン名を使用します。

   e. **mynetworks** パラメーターを追加し、メールの送信が許可される信頼できるネットワークの IP 範囲を定義します。

   ```
   mynetworks = 127.0.0.1/32, [::1]/128, 192.0.2.1/24, [2001:db8:1::1]/64
   ```

   インターネットなどの信頼できないネットワークからのクライアントがこのサーバー経由でメールを送信できるようにする必要がある場合は、後の手順でリレー制限を設定する必要があります。

3. **main.cf** ファイルの Postfix 設定が正しいか確認します。

   ```
   $ postfix check
   ```

4. **postfix** サービスが起動時に開始できるように有効化し、開始します。

   ```
   # systemctl enable --now postfix
   ```

5. smtp トラフィックがファイアウォールを通過することを許可し、ファイアウォールルールをリロードします。

```
# firewall-cmd --permanent --add-service smtp

# firewall-cmd --reload
```

検証

1. **postfix** サービスが実行していることを確認します。

```
# systemctl status postfix
```

- オプション: 出力が停止し、待機中、またはサービスが実行されていない場合は、**postfix** サービスを再起動します。

```
# systemctl restart postfix
```

- オプション: **/etc/postfix/** ディレクトリーの設定ファイル内のオプションを変更した後、**postfix** サービスをリロードして、これらの変更を適用します。

```
# systemctl reload postfix
```

2. システム上のローカルユーザー間の電子メール通信を確認します。

```
# echo "This is a test message" | mail -s <SUBJECT> <user@mydomain.com>
```

3. クライアント (**server1**) からメールサーバー (**server2**) へ、ドメイン外のメールアドレスにメールを送信して、メールサーバーがオープンリレーではないことを確認します。

a. 次のように、server1 の **/etc/postfix/main.cf** ファイルを編集します。

```
relayhost = <ip_address_of_server2>
```

b. 次のように、server2 の **/etc/postfix/main.cf** ファイルを編集します。

```
mynetworks = <ip_address_of_server2>
```

c. **server1** で、以下のメールを送信します。

```
# echo "This is an open relay test message" | mail -s <SUBJECT>
<user@example.com>
```

d. **/var/log/maillog** ファイルを確認します。

```
554 Relay access denied - the server is not going to relay.
250 OK or similar - the server is going to relay.
```

トラブルシューティング

- エラーが発生した場合は、**/var/log/maillog** を確認してください。

関連情報

- **/etc/postfix/main.cf** 設定ファイル

- **/usr/share/doc/postfix/README_FILES** ディレクトリー
- firewalld の使用および設定

2.3. POSTFIX サーバーの TLS 設定のカスタマイズ

電子メールトラフィックを暗号化してよりセキュアにするために、自己署名証明書の代わりに、信頼できる認証局 (CA) からの証明書を使用し、Transport Layer Security (TLS) セキュリティー設定をカスタマイズするように Postfix を設定できます。RHEL 9 では、TLS 暗号化プロトコルが Postfix サーバーでデフォルトで有効になっています。基本的な Postfix TLS 設定には、受信 SMTP 用の自己署名証明書と、発信 SMTP の日和見 TLS が含まれています。

前提条件

- root アクセスがある。
- サーバーに **postfix** パッケージがインストールされている。
- 信頼できる認証局 (CA) によって署名された証明書と秘密鍵を持っている。
- 以下のファイルを Postfix サーバーにコピーしている。
  - サーバー証明書: **/etc/pki/tls/certs/postfix.pem**
  - 秘密鍵y: **/etc/pki/tls/private/postfix.key**
- サーバーが RHEL 9.2 以降を実行し、FIPS モードが有効になっている場合、クライアントが Extended Master Secret (EMS) 拡張機能をサポートしているか、TLS 1.3 を使用している必要があります。EMS を使用しない TLS 1.2 接続は失敗します。詳細は、ナレッジベースの記事 TLS extension "Extended Master Secret" enforced を参照してください。

手順

1. 以下の行を **/etc/postfix/main.cf** ファイルに追加して、Postfix が実行されているサーバー上の証明書と秘密鍵ファイルへのパスを設定します。

```
smtpd_tls_cert_file = /etc/pki/tls/certs/postfix.pem
smtpd_tls_key_file = /etc/pki/tls/private/postfix.key
```

2. **/etc/postfix/main.cf** ファイルを編集して、受信した SMTP 接続を認証されたユーザーのみに制限します。

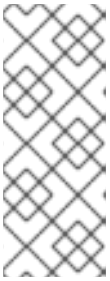
```
smtpd_tls_auth_only = yes
```

3. **postfix** サービスをリロードして変更を適用します。

```
# systemctl reload postfix
```

検証

- TLS 暗号化を使用してメールを送信するようにクライアントを設定します。



注記

Postfix クライアント TLS アクティビティーに関する追加情報を取得するには、**/etc/postfix/main.cf** の次の行を変更して、ログレベルを **0** から **1** に増やします。

```
smtp_tls_loglevel = 1
```

2.4. すべての電子メールをメールリレーに転送するように POSTFIX を設定する

すべての電子メールをメールリレーに転送する場合は、Postfix サーバーを Null クライアントとして設定できます。この設定では、Postfix はメールを別のメールサーバーに転送するだけで、メールの受信はできません。

前提条件

- root アクセスがある。
- サーバーに **postfix** パッケージがインストールされている。
- メールを転送するリレーホストの IP アドレスまたはホスト名がある。

手順

1. Postfix がローカルの電子メール配信を受け入れ、それが Null クライアントになるのを防ぐには、**/etc/postfix/main.cf** ファイルを編集し、以下の変更を加えます。

   a. **mydestination** パラメーターを空の値に等しくなるように設定して、すべてのメールを転送するように Postfix を設定します。

   ```
   mydestination =
   ```

   この設定では、Postfix サーバーはメールの宛先ではなく、null クライアントとして機能します。

   b. Null クライアントからメールを受信するメールリレーサーバーを指定します。

   ```
   relayhost = <[ip_address_or_hostname]>
   ```

   リレーホストはメール配信を行います。**<ip_address_or_hostname>** を角括弧で囲みます。

   c. メールを配信するために、ループバックインターフェイスでのみリッスンするように Postfix メールサーバーを設定します。

   ```
   inet_interfaces = loopback-only
   ```

   d. Postfix がすべての送信メールの送信者ドメインをリレーメールサーバーの企業ドメインに書き換えるには、以下を設定します。

   ```
   myorigin = <relay.example.com>
   ```

e. ローカルメール配信を無効にするには、設定ファイルの最後に次のディレクティブを追加します。

```
local_transport = error: local delivery disabled
```

f. **mynetworks** パラメーターを追加して、Postfix が 127.0.0.0/8 IPv4 ネットワークと [::1]/128 IPv6 ネットワークから送信されたローカルシステムからの電子メールをメールリレーサーバーに転送するようにします。

```
mynetworks = 127.0.0.0/8, [::1]/128
```

2. **main.cf** ファイルの Postfix 設定が正しいか確認します。

```
$ postfix check
```

3. **postfix** サービスを再起動して変更を適用します。

```
# systemctl restart postfix
```

検証

- 電子メール通信がメールリレーに転送されていることを確認します。

```
# echo "This is a test message" | mail -s <SUBJECT> <user@example.com>
```

トラブルシューティング

- エラーが発生した場合は、**/var/log/maillog** を確認してください。

関連情報

- **/etc/postfix/main.cf** 設定ファイル

2.5. POSTFIX を複数のドメインの宛先として設定する

Postfix を、複数のドメインのメールを受信できるメールサーバーとして設定できます。この設定では、Postfix は、指定されたドメイン内のアドレスに送信された電子メールの最終宛先として機能します。以下を設定できます。

- 同じ電子メール宛先を指す複数の電子メールアドレスを設定する。
- 複数のドメインの受信メールを同じ Postfix サーバーにルーティングする。

前提条件

- root アクセスがある。
- Postfix サーバーを設定している。

手順

1. **/etc/postfix/virtual** 仮想エイリアスファイルで、各ドメインのメールアドレスを指定します。各電子メールアドレスを新しい行に追加します。

```
<info@example.com>  <user22@example.net>
<sales@example.com>  <user11@example.org>
```

この例では、Postfix は info@example.com に送信されたすべての電子メールを user22@example.net にリダイレクトし、sales@example.com に送信された電子メールを user11@example.org にリダイレクトします。

2. 仮想エイリアスマップのハッシュファイルを作成します。

```
# postmap /etc/postfix/virtual
```

このコマンドは、**/etc/postfix/virtual.db** ファイルを作成します。**/etc/postfix/virtual** ファイルを更新した後に、このコマンドを常に再実行する必要があります。

3. Postfix **/etc/postfix/main.cf** 設定ファイルで、**virtual_alias_maps** パラメーターを追加して、ハッシュファイルを指すようにします。

```
virtual_alias_maps = hash:/etc/postfix/virtual
```

4. **postfix** サービスをリロードして変更を適用します。

```
# systemctl reload postfix
```

検証

- 仮想メールアドレスの 1 つに電子メールを送信して、設定をテストします。

トラブルシューティング

- エラーが発生した場合は、**/var/log/maillog** を確認してください。

2.6. LDAP ディレクトリーの検索テーブルとしての使用

Lightweight Directory Access Protocol (LDAP) サーバーを使用してアカウント、ドメイン、またはエイリアスを保存する場合は、LDAP サーバーを検索テーブルとして使用するように Postfix を設定できます。検索用ファイルの代わりに LDAP を使用すると、中央データベースを使用できます。

前提条件

- root アクセスがある。
- サーバーに **postfix** パッケージがインストールされている。
- 必要なスキーマおよびユーザークレデンシャルを持つ LDAP サーバーがある。
- Postfix を実行しているサーバーに **postfix-ldap** プラグインがインストールされている。

手順

1. 以下の内容で **/etc/postfix/ldap-aliases.cf** ファイルを作成して、LDAP 検索パラメーターを設定します。

a. LDAP サーバーのホスト名を指定します。

```
server_host = <ldap.example.com>
```

b. LDAP 検索のベースドメイン名を指定します。

```
search_base = dc=<example>,dc=<com>
```

c. オプション: 要件に応じて LDAP 検索フィルターと属性をカスタマイズします。ディレクトリーを検索するフィルターのデフォルトは **query_filter = mailacceptinggeneralid=%s** です。

2. 以下の内容を追加して、LDAP ソースを **/etc/postfix/main.cf** 設定ファイルの検索テーブルとして有効にします。

```
virtual_alias_maps = ldap:/etc/postfix/ldap-aliases.cf
```

3. **postmap** コマンドを実行して LDAP 設定を確認します。これは、構文エラーまたは接続の問題をチェックします。

```
# postmap -q @<example.com> ldap:/etc/postfix/ldap-aliases.cf
```

4. **postfix** サービスをリロードして変更を適用します。

```
# systemctl reload postfix
```

検証

- テストメールを送信して、LDAP 検索が正しく機能していることを確認します。**/var/log/maillog** のメールログでエラーがないか確認します。

関連情報

- **/usr/share/doc/postfix/README_FILES/LDAP_README** ファイル
- **/usr/share/doc/postfix/README_FILES/DATABASE_README** ファイル

2.7. 認証されたユーザーのリレーを行う送信メールサーバーとしての POSTFIX の設定

認証されたユーザーのメールをリレーするように Postfix を設定できます。このシナリオでは、SMTP 認証、TLS 暗号化、および送信者アドレス制限を備えた送信メールサーバーとして Postfix を設定することで、ユーザーが自分自身を認証し、自分の電子メールアドレスを使用して SMTP サーバー経由でメールを送信できるようにします。

前提条件

- root アクセスがある。
- Postfix サーバーを設定している。

手順

1. Postfix を送信メールサーバーとして設定するには、**/etc/postfix/main.cf** ファイルを編集し、以下を追加します。

   a. SMTP 認証を有効にします。

   ```
   smtpd_sasl_auth_enable = yes
   broken_sasl_auth_clients = yes
   ```

   b. TLS を使用しないアクセスを無効にします。

   ```
   smtpd_tls_auth_only = yes
   ```

   c. 認証されたユーザーに対してのみメールリレーを許可します。

   ```
   smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
   defer_unauth_destination
   ```

   d. オプション: ユーザーが自分の電子メールアドレスを送信者としてのみ使用するように制限します。

   ```
   smtpd_sender_restrictions = reject_sender_login_mismatch
   ```

2. **postfix** サービスをリロードして変更を適用します。

   ```
   # systemctl reload postfix
   ```

検証

- TLS および SASL をサポートする SMTP クライアントで認証します。テストメールを送信して、SMTP 認証が正しく機能していることを確認します。

2.8. 同じホストで実行している POSTFIX から DOVECOT への電子メールの配信

UNIX ソケット経由で LMTP を使用して、受信メールを同じホスト上の Dovecot に配信するように Postfix を設定できます。このソケットは、ローカルマシンの Postfix と Dovecot との間の直接通信を有効にします。

前提条件

- root アクセスがある。
- Postfix サーバーを設定している。
- Dovecot サーバーを設定している。Dovecot IMAP および POP3 サーバーの設定と管理 を参照してください。
- Dovecot サーバーに LMTP ソケットを設定している。LMTP ソケットと LMTPS リスナーの設定 を参照してください。

手順

1. **/etc/postfix/main.cf** ファイルの Dovecot にメールを配信するために LMTP プロトコルと UNIX ドメインソケットを使用するように Postfix を設定します。
   - 仮想メールボックスを使用する場合は、次のコンテンツを追加します。

     ```
     virtual_transport = lmtp:unix:/var/run/dovecot/lmtp
     ```

   - 仮想以外のメールボックスを使用する場合は、次のコンテンツを追加します。

     ```
     mailbox_transport = lmtp:unix:/var/run/dovecot/lmtp
     ```

2. **postfix** をリロードして変更を適用します。

   ```
   # systemctl reload postfix
   ```

検証

- テストメールを送信して、LMTP ソケットが正常に動作することを確認します。**/var/log/maillog** のメールログでエラーがないか確認します。

2.9. POSTFIX から別のホストで実行されている DOVECOT への電子メールの配信

ネットワーク経由で Postfix メールサーバーと Dovecot 配信エージェントの間にセキュアな接続を確立できます。これを行うには、メールサーバー間でのメール配信にネットワークソケットを使用するように LMTP サービスを設定します。デフォルトでは、LMTP プロトコルは暗号化されていません。ただし、TLS 暗号化を設定した場合、Dovecot は LMTP サービスに同じ設定を自動的に使用します。続いて、SMTP サーバーは、LMTP 経由で **STARTTLS** コマンドを使用してそれに接続できます。

前提条件

- root アクセスがある。
- Postfix サーバーを設定している。
- Dovecot サーバーを設定している。Dovecot IMAP および POP3 サーバーの設定と管理 を参照してください。
- Dovecot サーバーで LMTP サービスを設定している。LMTP ソケットと LMTPS リスナーの設定 を参照してください。

手順

1. 以下の内容を追加して、**/etc/postfix/main.cf** ファイルで Dovecot にメールを配信するために LMTP プロトコルと INET ドメインソケットを使用するように Postfix を設定します。

   ```
   mailbox_transport = lmtp:inet:<dovecot_host>:<port>
   ```

   <dovecot_host> を Dovecot サーバーの IP アドレスまたはホスト名に置き換え、 **<port>** を LMTP サービスのポート番号に置き換えます。

2. **postfix** サービスをリロードして変更を適用します。

```
# systemctl reload postfix
```

検証

- リモート Dovecot サーバーがホストするアドレスにテストメールを送信し、Dovecot ログをチェックして、メールが正常に配信されたことを確認します。

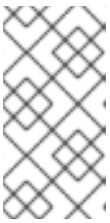
2.10. POSTFIX サービスを保護する

Postfix は、SMTP (Simple Mail Transfer Protocol) を使用して他の MTA 間で電子メッセージを配信したり、クライアントや配信エージェントに電子メールを送信したりするメール転送エージェント (MTA) です。MTA は相互間のトラフィックを暗号化できますが、デフォルトではそうしない場合があります。設定をより安全な値に変更することで、さまざまな攻撃に対するリスクを軽減することもできます。

2.10.1. Postfix ネットワーク関連のセキュリティーリスクの軽減

攻撃者がネットワーク経由でシステムに侵入するリスクを軽減するには、次のタスクをできるだけ多く実行してください。

- ネットワークファイルシステム (NFS) 共有ボリュームで **/var/spool/postfix/** メールスプールディレクトリーを共有しないでください。NFSv2 と NFSv3 は、ユーザー ID とグループ ID に対する制御を維持しません。したがって、2 人以上のユーザーが同じ UID を持っていると、互いのメールを受信して読むことができ、セキュリティー上のリスクが生じます。



注記

SECRPC_GSS カーネルモジュールは UID ベースの認証を使用しないため、この規則は Kerberos を使用する NFSv4 には適用されません。ただし、セキュリティーリスクを軽減するために、メールスプールディレクトリーを NFS 共有ボリュームに配置しないでください。

- Postfix サーバーの悪用の可能性を減らすために、メールユーザーは電子メールプログラムを使用して Postfix サーバーにアクセスする必要があります。メールサーバーでシェルアカウントを許可せず、**/etc/passwd** ファイル内のすべてのユーザーシェルを **/sbin/nologin** に設定します (**root** ユーザーは例外の可能性があります)。

- Postfix をネットワーク攻撃から保護するために、デフォルトではローカルループバックアドレスのみをリッスンするように設定されています。これは、**/etc/postfix/main.cf** ファイルの **inet_interfaces = localhost** 行を表示することで確認できます。これにより、Postfix はネットワークからではなく、ローカルシステムからのメールメッセージ (**cron** ジョブのレポートなど) のみを受け入れるようになります。これはデフォルトの設定で、Postfix をネットワーク攻撃から保護します。localhost の制限を取り除き、Postfix がすべてのインターフェイスでリッスンできるようにするには、**/etc/postfix/main.cf** で **inet_interfaces** パラメーターを **all** に設定します。

2.10.2. DoS 攻撃を制限するための Postfix 設定オプション

攻撃者は、トラフィックでサーバーをあふれさせたり、クラッシュを引き起こす情報を送信したりして、サービス拒否 (DoS) 攻撃を引き起こす可能性があります。**/etc/postfix/main.cf** ファイルで制限を設定することにより、このような攻撃のリスクを軽減するようにシステムを設定できます。既存のディレクティブの値を変更するか、**<directive> = <value>**形式のカスタム値で新しいディレクティブを追加できます。

DoS 攻撃を制限するには、次のディレクティブリストを使用します。

smtpd_client_connection_rate_limit

このディレクティブは、時間単位ごとにクライアントがこのサービスに対して行うことができる接続試行の最大数を制限します。デフォルト値は **0** です。これは、クライアントが時間単位で Postfix が受け入れることができる数と同じ数の接続を行うことができることを意味します。デフォルトでは、ディレクティブは信頼できるネットワークのクライアントを除外します。

anvil_rate_time_unit

このディレクティブは、レート制限を計算する時間単位です。デフォルト値は **60** 秒です。

smtpd_client_event_limit_exceptions

このディレクティブは、接続およびレート制限コマンドからクライアントを除外します。デフォルトでは、ディレクティブは信頼できるネットワークのクライアントを除外します。

smtpd_client_message_rate_limit

このディレクティブは、単位時間当たりのクライアントからリクエストへのメッセージ配信の最大数を定義します (Postfix が実際にそれらのメッセージを受け入れるかどうかに関係なく)。

default_process_limit

このディレクティブは、特定のサービスを提供する Postfix 子プロセスのデフォルトの最大数を定義します。**master.cf** ファイル内の特定のサービスについては、このルールを無視できます。デフォルトでは、値は **100** です。

queue_minfree

このディレクティブは、キューファイルシステムでメールを受信するために必要な空き容量の最小量を定義します。このディレクティブは現在、Postfix SMTP サーバーがメールを受け入れるかどうかを決定するために使用されています。デフォルトでは、Postfix SMTP サーバーは、空き容量が **message_size_limit** の 1.5 倍未満の場合に、**MAIL FROM** コマンドを拒否します。空き容量の最小値をこれよりも高く指定するには、**message_size_limit** の 1.5 倍以上の **queue_minfree** 値を指定します。デフォルトの **queue_minfree** 値は **0** です。

header_size_limit

このディレクティブは、メッセージヘッダーを格納するためのメモリーの最大量をバイト単位で定義します。ヘッダーが大きい場合、余分なヘッダーは破棄されます。デフォルトでは、値は **102400** バイトです。

message_size_limit

このディレクティブは、エンベロープ情報を含むメッセージの最大サイズをバイト単位で定義します。デフォルトでは、値は **10240000** バイトです。

2.10.3. Postfix が SASL を使用する設定

Postfix は Simple Authentication and Security Layer (SASL) ベースの SMTP 認証 (AUTH) をサポートしています。SMTP AUTH は Simple Mail Transfer Protocol の拡張です。現在、Postfix SMTP サーバーは次の方法で SASL 実装をサポートしています:

Dovecot SASL

Postfix SMTP サーバーは、UNIX ドメインソケットまたは TCP ソケットのいずれかを使用して、Dovecot SASL 実装と通信できます。Postfix と Dovecot アプリケーションが別のマシンで実行している場合は、この方法を使用します。

Cyrus SASL

有効にすると、SMTP クライアントは、サーバーとクライアントの両方でサポートおよび受け入れられる認証方法を使用して、SMTP サーバーで認証する必要があります。

前提条件

- **dovecot** パッケージがシステムにインストールされている

手順

1. Dovecot をセットアップします。

   a. **/etc/dovecot/conf.d/10-master.conf** ファイルに次の行を含めます。

   ```
   service auth {
     unix_listener /var/spool/postfix/private/auth {
       mode = 0660
       user = postfix
       group = postfix
     }
   }
   ```

   前の例では、Postfix と Dovecot の間の通信に UNIX ドメインソケットを使用しています。また、**/var/spool/postfix/** ディレクトリーにあるメールキュー、および **postfix** ユーザーとグループの下で実行しているアプリケーションを含む Postfix SMTP サーバーのデフォルト設定を想定しています。

   b. オプション: TCP 経由で Postfix 認証リクエストをリッスンするように Dovecot をセットアップします。

   ```
   service auth {
     inet_listener {
         port = port-number
     }
   }
   ```

   c. **/etc/dovecot/conf.d/10-auth.conf** ファイルの **auth_mechanisms** パラメーターを編集して、電子メールクライアントが Dovecot での認証に使用する方法を指定します。

   ```
   auth_mechanisms = plain login
   ```

   auth_mechanisms パラメーターは、さまざまなプレーンテキストおよび非プレーンテキストの認証方法をサポートしています。

2. **/etc/postfix/main.cf** ファイルを変更して Postfix をセットアップします。

   a. Postfix SMTP サーバーで SMTP 認証を有効にします。

   ```
   smtpd_sasl_auth_enable = yes
   ```

   b. SMTP 認証用の Dovecot SASL 実装の使用を有効にします。

   ```
   smtpd_sasl_type = dovecot
   ```

   c. Postfix キューディレクトリーに相対的な認証パスを指定します。相対パスを使用すると、Postfix サーバーが **chroot** で実行しているかどうかに関係なく、設定が確実に機能することに注意してください。

   ```
   smtpd_sasl_path = private/auth
   ```

この手順では、Postfix と Dovecot の間の通信に UNIX ドメインソケットを使用します。

通信に TCP ソケットを使用する場合に、別のマシンで Dovecot を探すように Postfix を設定するには、次のような設定値を使用します。

```
smtpd_sasl_path = inet: ip-address : port-number
```

前の例で、**ip-address** を Dovecot マシンの IP アドレスに置き換え、 **port-number** を Dovecot の **/etc/dovecot/conf.d/10-master.conf** ファイルで指定されたポート番号に置き換えます。

d. Postfix SMTP サーバーがクライアントに提供する SASL メカニズムを指定します。暗号化されたセッションと暗号化されていないセッションに異なるメカニズムを指定できることに注意してください。

```
smtpd_sasl_security_options = noanonymous, noplaintext
smtpd_sasl_tls_security_options = noanonymous
```

前のディレクティブは、暗号化されていないセッションでは匿名認証が許可されず、暗号化されていないユーザー名またはパスワードを送信するメカニズムが許可されていないことを指定しています。暗号化セッション (TLS を使用) の場合、非匿名認証メカニズムのみが許可されます。

関連情報

- Postfix SMTP server policy - SASL mechanism properties
- Postfix and Dovecot SASL
- Postfix SMTP サーバーで SASL 認証を設定する