Red Hat Enterprise Linux 9

Identity Management での Vault の操作

IdM での機密データの保存と管理

Last Updated: 2023-12-01

Red Hat Enterprise Linux 9 Identity Management での Vault の操作

IdM での機密データの保存と管理

法律上の通知

概要

vault は、サービスの認証情報などの機密データを保存、取得、共有するための Red Hat Identity Management (IdM) の安全な場所です。コマンドラインまたは Ansible Playbook を使用して vault を管理できます。

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 IDM の VAULT

本章では、Identity Management(IdM) の vault について説明します。本章では、以下のトピックを紹介します。

- vault の概念。
- vault に関連付けられる各種ロール。
- IdM で利用可能な各種 vault - セキュリティーおよびアクセス制御のレベル別 。
- IdM で利用可能な各種 vault - 所有権別 。
- vault コンテナーの概念。
- IdM での vault 管理向けの基本的なコマンド 。
- IdM で vault を使用するのに必要な KPA (Key Recovery Authority) のインストール 。

1.1. VAULT およびその利点

vault は、機密データをすべてセキュアに保存しつつも、1 箇所で都合よく Identity Management (IdM) を使用するのに便利な機能です。 vault にはさまざまなタイプがあり、要件に応じて使用する vault を選択する必要があります。

vault とは、シークレットの保存、取得、共有、および復旧を行うための (IdM の) セキュアな場所を指し、シークレットは、通常は一部のユーザーまたはエンティティーグループのみがアクセスできる、認証情報などの機密データを指します。たとえば、シークレットには以下が含まれます。

- パスワード
- 暗証番号
- SSH 秘密鍵

vault はパスワードマネージャーと類似しています。valut を使用する場合、通常、パスワードマネージャーと同様に、ロックを解除するためのプライマリーのパスワードを 1 つ生成し、記憶して、vault に保存されている情報にアクセスする必要があります。ただし、標準の vault を指定することも可能です。標準の vault では、vault に保存されているシークレットにアクセスするためにパスワードを入力する必要はありません。

注記

IdM の vault は、認証情報を保存して、IdM 関連以外の外部サービスに対して認証を可能にすることを目的としています。

IdM vault には他にも、次のような重要な特徴があります。

- vault にアクセスできるのは、vault の所有者と、vault メンバーとして vault の所有者が選択した IdM ユーザーだけです。また、IdM 管理者も vault にアクセスできます。
- ユーザーに vault を作成する権限がない場合には、IdM 管理者が vault を作成し、そのユーザーを所有者として設定できます。

- ユーザーおよびサービスは、IdM ドメインに登録されているマシンからであれば、vault に保存されているシークレットにアクセスできます。
- vault 1 つに追加できるシークレットは 1 つのみです (例: ファイル 1 つ)。ただし、ファイル自体には、パスワード、キータブ、証明書など複数のシークレットを含めることができます。

注記

Vault は、IdM Web UI ではなく、IdM コマンドライン (CLI) からしか利用できません。

1.2. VAULT の所有者、メンバー、および管理者

Identity Management (IdM) で識別される vault ユーザータイプは以下のとおりです。

Vault 所有者

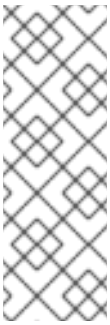
vault 所有者は、vault の基本的な管理権限のあるユーザーまたはサービスです。たとえば、vault の所有者は vault のプロパティーを変更したり、新しい vault メンバーを追加したりできます。
各 vault には最低でも所有者が 1 人必要です。vault には複数の所有者を指定することもできます。

Vault メンバー

vault メンバーは、別のユーザーまたはサービスが作成した vault にアクセスできるユーザーまたはサービスです。

Vault 管理者

vault 管理者は全 vault に制限なくアクセスでき、vault の操作をすべて実行できます。



注記

対称と非対称 vault は、パスワードまたは鍵で保護されており、特別なアクセス制御ルールが適用されます (Vault タイプ を参照)。管理者は、以下を行うためにこの特別なルールを満たす必要があります。

- 対称および非対称 vault のシークレットにアクセスする。
- vault パスワードまたはキーを変更またはリセットする。

vault 管理者は、**vault administrators** 特権を持つユーザーです。IdM のロールベースアクセス制御 (RBAC) のコンテキストでの特権とは、ロールに適用できるパーミッションのグループのことです。

Vault ユーザー

vault ユーザーは、vault のあるコンテナー内のユーザーです。**Vault ユーザー** 情報は、**ipa vault-show** などの特定のコマンドの出力に表示されます。

```
$ ipa vault-show my_vault
  Vault name: my_vault
  Type: standard
  Owner users: user
  Vault user: user
```

vault コンテナーおよびユーザー vault の詳細は、Vault コンテナー を参照してください。

関連情報

- vault のタイプの詳細は、標準、対称および非対称 vault を参照してください。

1.3. 標準、対称および非対称 VAULT

IdM では、セキュリティーおよびアクセス制御のレベルをもとに vault を以下のタイプに分類します。

標準 vault

Vault の所有者と vault メンバーは、パスワードやキーを使用せずにシークレットをアーカイブして取得できます。

対称 vault

vault のシークレットは対称キーを使用して保護されます。Vault の所有者とメンバーは、シークレットをアーカイブして取得できますが、vault パスワードを指定する必要があります。

非対称 vault

vault のシークレットは非対称キーを使用して保護されます。ユーザーは公開鍵でシークレットをアーカイブし、秘密鍵でシークレットを取得します。vault メンバーはシークレットのアーカイブのみが可能ですが、vault 所有者はシークレットのアーカイブと取得の両方が可能です。

1.4. ユーザー、サービスおよび共有 VAULT

IdM では、所有権をもとに vault を複数のタイプに分類します。以下の表 には、各タイプ、所有者、および使用方法に関する情報が含まれます。

表1.1 所有権に基づく IdM vault

タイプ	説明	所有者	備考
ユーザー vault	ユーザーのプライベート vault	ユーザー x 1	IdM 管理者が許可すれば、ユーザーは 1 つまたは複数のユーザー vault を所有できます。
サービス vault	サービスのプライベート vault	サービス x 1	IdM 管理者が許可すれば、ユーザーは 1 つまたは複数のサービス vault を所有できます。
共有 vault	複数のユーザーおよびグループで共有される vault	vault を作成した vault の管理者	IdM 管理者が許可すれば、ユーザーおよびサービスは 1 つまたは複数のユーザー vault を所有できます。vault の作成者以外に、vault 管理者が vault に対して完全なアクセス権があります。

1.5. VAULT コンテナー

vault コンテナーは vault のコレクションです。以下の表 は、Identity Management (IdM) が提供するデフォルトの vault コンテナーのリストです。

表1.2 IdM のデフォルトの vault コンテナー

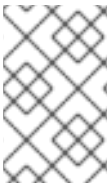
タイプ	説明	目的

タイプ	説明	目的
ユーザーコンテナー	ユーザーのプライベートコンテナー	特定ユーザーのユーザー vault を格納します。
サービスコンテナー	サービスのプライベートコンテナー	特定のサービスのサービス vault を格納します。
共有コンテナー	複数のユーザーおよびサービスのコンテナー	複数のユーザーまたはサービスで共有可能な vault を格納します。

IdM では、ユーザーまたはサービスのプライベート vault が初めて作成されると、ユーザーまたはサービスごとにユーザーコンテナーおよびサービスコンテナーを自動的に作成します。ユーザーまたはサービスが削除されると、IdM はコンテナーとそのコンテンツを削除します。

1.6. 基本的な IDM VAULT コマンド

以下に概説する基本的なコマンドを使用して、Identity Management (IdM) vault を管理できます。以下の表 には、**ipa vault-*** コマンドとその目的が記載されています。



注記

ipa vault-* コマンドを実行する前に、IdM ドメインのサーバー 1 台以上に Key Recovery Authority (KRA) 証明書システムコンポーネントをインストールします。詳細は IdM での Key Recovery Authority (KRA) のインストール を参照してください。

表1.3 基本的な IdM vault コマンドおよび説明

コマンド	目的
ipa help vault	IdM vault コマンドおよびサンプル vault コマンドの概念などの情報を表示します。
ipa vault-add --help、ipa vault-find --help	特定の **ipa vault-*** コマンドに **--help** オプションを追加すると、このコマンドで利用可能なオプションと詳細なヘルプが表示されます。

ipa vault-show user_vault --user idm_user	vault メンバーとして vault にアクセスする場合は、vault 所有者を指定する必要があります。vault 所有者を指定しない場合には、IdM により vault が見つからない旨が通知されます。
	`[admin@server ~]$ ipa vault-show user_vault`
	`ipa: ERROR: user_vault: vault not found`

コマンド	目的
ipa vault-show shared_vault --shared	共有 vault にアクセスする場合には、アクセスする vault が共有 vault であることを指定する必要があります。それ以外の場合は、IdM により vault が見つからない旨が通知されます。 [admin@server ~]$ ipa vault-show shared_vault ipa: ERROR: shared_vault: vault not found

1.7. IDM での KEY RECOVERY AUTHORITY (KRA) のインストール

以下の手順に従って、特定の IdM サーバーに Key Recovery Authority (KRA) Certificate System (CS) コンポーネントをインストールして、Identity Management (IdM) の vault を有効にします。

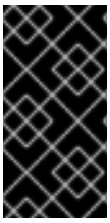
前提条件

- IdM サーバーに **root** としてログインしている。
- IdM 認証局が IdM サーバーにインストールされている。
- **Directory Manager** 認証情報がある。

手順

- KRA をインストールします。

```
# ipa-kra-install
```



重要

非表示のレプリカに、IdM クラスターの最初の KRA をインストールできます。ただし、追加の KRA をインストールするには、非表示レプリカを一時的にアクティベートしてから、表示されているレプリカに KRA のクローンをインストールする必要があります。その後に、最初に非表示レプリカを再度非表示にできます。

注記

vault サービスの可用性と耐障害性を高めるには、2 台以上の IdM サーバーに KRA をインストールします。複数の KRA サーバーを保持することで、データの損失を防ぎます。

関連情報

- Demoting or promoting hidden replicas を参照してください。
- The hidden replica mode を参照してください。

第2章 IDM ユーザー VAULT の使用: シークレットの保存および取得

本章では、Identity Management でユーザー vault を使用する方法を説明します。具体的には、IdM vault にシークレットを保存する方法と、シークレットを取得する方法を説明します。異なる IdM クライアント 2 台から保存と取得が可能です。

前提条件

- Key Recovery Authority (KRA) Certificate System コンポーネントが IdM ドメインの 1 つ以上のサーバーにインストールされている。詳細は IdM での Key Recovery Authority (KRA) のインストール を参照してください。

2.1. ユーザー VAULT でのシークレットの保存

この手順に従って、機密情報を含むファイルを安全に保存するための 1 つ以上のプライベート vault を含む vault コンテナーを作成します。以下の手順で使用する例では、**idm_user** ユーザーが標準タイプの vault を作成します。標準タイプの vault では、ファイルへのアクセス時に **idm_user** を認証する必要がありません。**IdM_user** は、ユーザーがログインしている IdM クライアントからファイルを取得できます。

この手順では、以下を想定しています。

- **IdM_user** は vault を作成するユーザーである。
- **my_vault** はユーザーの証明書保存に使用する vault である。
- アーカイブした証明書にアクセスするのに vault のパスワードを指定しなくてもいいように vault タイプが **standard** に設定されている。
- **secret.txt** は vault に保存する証明書が含まれるファイルです。

前提条件

- **idm_user** のパスワードを知っている。
- IdM クライアントであるホストにログインしている。

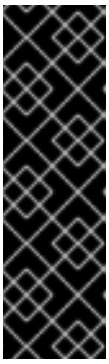
手順

1. **idm_user** の Kerberos Ticket Granting Ticket (TGT) を取得します。

```
$ kinit idm_user
```

2. **ipa vault-add** コマンドに **--type standard** オプションを指定して、標準 vault を作成します。

```
$ ipa vault-add my_vault --type standard
----------------------
Added vault "my_vault"
----------------------
  Vault name: my_vault
  Type: standard
  Owner users: idm_user
  Vault user: idm_user
```



重要

最初のユーザー vault の作成には、同じユーザーが使用されているようにしてください。ユーザーの最初の vault を作成すると、ユーザーの vault コンテナーも作成されます。作成エージェントは vault コンテナーの所有者になります。

たとえば、**admin** などの別のユーザーが **user1** の最初のユーザー vault を作成する場合には、ユーザーの vault コンテナーの所有者も **admin** になり、**user1** はユーザー vault にアクセスしたり、新しいユーザー vault を作成したりできません。

3. **ipa vault-archive** コマンドに **--in** オプションを指定して、**secret.txt** ファイルを vault にアーカイブします。

```
$ ipa vault-archive my_vault --in secret.txt
----------------------------------
Archived data into vault "my_vault"
----------------------------------
```

2.2. ユーザー VAULT からのシークレットの取得

Identity Management (IdM) では、ユーザープライベート vault からシークレットを取得して、ログインしている IdM クライアントに配置できます。

この手順に従って、**idm_user** という名前の IdM ユーザーが **my_vault** という名前のユーザープライベート vault からシークレットを取得して **idm_client.idm.example.com** に配置します。

前提条件

- **IdM_user** が **my_vault** の所有者である。
- **IdM_user** が vault にシークレットをアーカイブしてある。
- **my_vault** は標準の vault であるため、**idm_user** は vault のコンテンツへのアクセスにパスワードを入力する必要はありません。

手順

1. **idm_client** に **idm_user** として SSH 接続します。

```
$ ssh idm_user@idm_client.idm.example.com
```

2. **idm_user** としてログインします。

```
$ kinit user
```

3. **--out** オプションを指定して **ipa vault-retrieve --out** コマンドを使用し、vault のコンテンツを取得して、**secret_exported.txt** ファイルに保存します。

```
$ ipa vault-retrieve my_vault --out secret_exported.txt
-------------------------------------
Retrieved data from vault "my_vault"
-------------------------------------
```

2.3. 関連情報

- Ansible を使用した IdM サービス vault の管理: シークレットの保存および取得 を参照してください。

第3章 ANSIBLE を使用した IDM ユーザー VAULT の管理: シークレットの保存および取得

本章では、Ansible **vault** モジュールを使用して Identity Management でユーザー vault を管理する方法を説明します。具体的には、ユーザーが Ansible Playbook を使用して以下の 3 つのアクションを実行する方法を説明しています。

- IdM でユーザーコンテナーを作成する。
- シークレットを vault に保存する。
- vault からシークレットを取得する。

異なる IdM クライアント 2 台から保存と取得が可能です。

前提条件

- Key Recovery Authority (KRA) Certificate System コンポーネントが IdM ドメインの 1 つ以上のサーバーにインストールされている。詳細は IdM での Key Recovery Authority (KRA) のインストール を参照してください。

3.1. ANSIBLE を使用して IDM に標準ユーザー VAULT を存在させる手順

以下の手順に従って、Ansible Playbook を使用して 1 つ以上のプライベート vault を持つ vault コンテナーを作成し、機密情報を安全に保存します。以下の手順で使用する例では、**idm_user ユーザー** は **my_vault** という名前の標準タイプの vault を作成します。標準タイプの vault では、ファイルへのアクセス時に **idm_user** を認証する必要がありません。**IdM_user** は、ユーザーがログインしている IdM クライアントからファイルを取得できます。

前提条件

- Ansible コントローラー (手順の内容を実行するホスト) に ansible-freeipa パッケージがインストールされている。
- **idm_user** のパスワードを知っている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. **inventory.file** などのインベントリーファイルを作成します。

```
$ touch inventory.file
```

3. **inventory.file** を開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

```
[ipaserver]
server.idm.example.com
```

4. Ansible Playbook の **ensure-standard-vault-is-present.yml** ファイルのコピーを作成します。以下に例を示します。

```
$ cp ensure-standard-vault-is-present.yml ensure-standard-vault-is-present-copy.yml
```

5. **ensure-standard-vault-is-present-copy.yml** ファイルを開いて編集します。

6. **ipavault** タスクセクションに以下の変数を設定して、ファイルを調整します。

   - **ipaadmin_principal** 変数は idm_user に設定します。
   - **ipaadmin_password** 変数は idm_user のパスワードに設定します。
   - **user** 変数は idm_user に設定します。
   - **name** 変数は my_vault に設定します。
   - **vault_type** 変数は standard に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      vault_type: standard
```

7. ファイルを保存します。

8. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-standard-vault-is-present-copy.yml
```

3.2. ANSIBLE を使用して IDM の標準ユーザー VAULT でシークレットをアーカイブする手順

以下の手順に従って、Ansible Playbook を使用してパーソナル vault に機密情報を保存します。この例では、**idm_user** ユーザーは **my_vault** という名前の vault に **password.txt** という名前で機密情報が含まれるファイルをアーカイブします。

前提条件

- Ansible コントローラー (手順の内容を実行するホスト) に ansible-freeipa パッケージがインストールされている。

- idm_user のパスワードを知っている。
- IdM_user が所有者であるか、my_vault のメンバーユーザーである。
- password.txt (my_vault にアーカイブするシークレット) にアクセスできる。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. インベントリーファイルを開き、設定する IdM サーバーが **[ipaserver]** セクションに記載されていることを確認します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

```
[ipaserver]
server.idm.example.com
```

3. Ansible Playbook の **data-archive-in-symmetric-vault.yml** ファイルのコピーを作成して symmetric を standard に置き換えます。以下に例を示します。

```
$ cp data-archive-in-symmetric-vault.yml data-archive-in-standard-vault-copy.yml
```

4. **data-archive-in-standard-vault-copy.yml** ファイルを開いて編集します。
5. **ipavault** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_principal** 変数は idm_user に設定します。
   - **ipaadmin_password** 変数は idm_user のパスワードに設定します。
   - **user** 変数は idm_user に設定します。
   - **name** 変数は my_vault に設定します。
   - **in** 変数は機密情報が含まれるファイルへのパスに設定します。
   - **action** 変数は member に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
```

```
    name: my_vault
    in: /usr/share/doc/ansible-freeipa/playbooks/vault/password.txt
    action: member
```

6. ファイルを保存します。
7. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-archive-in-standard-vault-copy.yml
```

3.3. ANSIBLE を使用して IDM の標準ユーザー VAULT からシークレットを取得する手順

以下の手順に従って、Ansible Playbook を使用してユーザーのパーソナル vault からシークレットを取得します。以下の手順で使用する例では、**idm_user** ユーザーは、**my_vault** という名前の標準タイプの vault から機密データを含むファイルを取得して、**host01** という名前の IdM クライアントに配置します。ファイルへのアクセス時に **IdM_user** を認証する必要はありません。**IdM_user** は Ansible を使用して、Ansible がインストールされている IdM クライアントからファイルを取得できます。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible ボールトに **ipaadmin_password** が保存されていることを前提としている。
- **idm_user** のパスワードを知っている。
- **IdM_user** が **my_vault** の所有者である。
- **IdM_user** が **my_vault** にシークレットを保存している。
- Ansible が、シークレットを取得して配置する先の IdM ホストのディレクトリーに書き込みができる。
- **IdM_user** はシークレットを取得して配置する先の IdM ホストのディレクトリーから読み取りができる。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. インベントリーファイルを開き、明確に定義されたセクションで、シークレットを取得する IdM クライアントを記載します。たとえば、Ansible に対して **host01.idm.example.com** にシークレットを取得して配置するよう指示するには、次のコマンドを実行します。

```
[ipahost]
host01.idm.example.com
```

3. Ansible Playbook ファイル (**retrive-data-symmetric-vault.yml**) のコピーを作成します。symbolicmetric を Standard に置き換えます。以下に例を示します。

```
$ cp retrive-data-symmetric-vault.yml retrieve-data-standard-vault.yml-copy.yml
```

4. **retrieve-data-standard-vault.yml-copy.yml** ファイルを開いて編集します。
5. **hosts** 変数は **ipahost** に設定して、ファイルを調整します。
6. **ipavault** タスクセクションに以下の変数を設定して、ファイルを調整します。
   - **ipaadmin_principal** 変数は **idm_user** に設定します。
   - **ipaadmin_password** 変数は **idm_user** のパスワードに設定します。
   - **user** 変数は **idm_user** に設定します。
   - **name** 変数は **my_vault** に設定します。
   - **out** 変数は、シークレットをエクスポートするファイルの完全パスに設定します。
   - **state** 変数は **retrieved** に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- name: Tests
  hosts: ipahost
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_principal: idm_user
      ipaadmin_password: idm_user_password
      user: idm_user
      name: my_vault
      out: /tmp/password_exported.txt
      state: retrieved
```

7. ファイルを保存します。
8. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-standard-vault.yml-copy.yml
```

検証手順

1. **host01** に **user01** として **SSH** 接続します。

```
$ ssh user01@host01.idm.example.com
```

2. Ansible Playbook ファイルに **out** 変数で指定したファイルを表示します。

```
$ vim /tmp/password_exported.txt
```

これで、エクスポートされたシークレットが表示されます。

- Ansible を使用して IdM vault およびユーザーシークレットを管理する方法および、Playbook 変数の情報は、**/usr/share/doc/ansible-freeipa/** ディレクトリーで利用可能な README-vault.md Markdown ファイルおよび **/usr/share/doc/ansible-freeipa/playbooks/vault/** で利用可能なサンプルの Playbook を参照してください。

第4章 IDM サービスシークレットの管理: シークレットの保存と取得

本セクションでは、管理者が **ansible-freeipa vault** モジュールを使用してサービスシークレットを一元的にセキュアに保存する方法を説明します。この例で使用される vault は非対称であるため、これを使用する場合は、管理者は以下の手順を実行する必要があります。

1. **openssl** ユーティリティーなどを使用して秘密鍵を生成する。
2. 秘密鍵をもとに公開鍵を生成する。

サービスシークレットは、管理者が vault にアーカイブする時に公開鍵を使用して暗号化されます。その後、ドメイン内の特定のマシンでホストされるサービスインスタンスが、秘密鍵を使用してシークレットを取得します。シークレットにアクセスできるのは、サービスと管理者のみです。

シークレットが漏洩した場合には、管理者はサービス Vault でシークレットを置き換えて、漏洩されていないサービスインスタンスに配布しなおすことができます。

前提条件

- Key Recovery Authority (KRA) Certificate System コンポーネントが IdM ドメインの 1 つ以上のサーバーにインストールされている。詳細は IdM での Key Recovery Authority (KRA) のインストール を参照してください。

このセクションでは、以下の手順について説明します。

1. 非対称 vault での IdM サービスシークレットの保存
2. IdM サービスインスタンスのサービスシークレットの取得
3. シークレットが漏洩した場合の IdM サービス vault シークレットの変更

使用される用語

本手順での以下の用語について説明します。

- **admin** は、サービスパスワードを管理する管理者です。
- **private-key-to-an-externally-signed-certificate.pem** は、サービスシークレットを含むファイルです (ここでは外部署名証明書への秘密鍵)。この秘密鍵と、vault からのシークレットの取得に使用する秘密鍵と混同しないようにしてください。
- **secret_vault** は、サービス向けに作成された vault です。
- **HTTP/webserver.idm.example.com** は、シークレットがアーカイブされるサービスです。
- **service-public.pem** は、**password_vault** に保存されているパスワードの暗号化に使用するサービスの公開鍵です。
- **service-private.pem** は、**secret_vault** に保存されているパスワードの復号化に使用するサービスの秘密鍵です。

4.1. 非対称 VAULT での IDM サービスシークレットの保存

この手順に従って非対称 vault を作成し、それを使用してサービスシークレットをアーカイブします。

前提条件

- IdM 管理者パスワードを把握している。

手順

1. 管理者としてログインします。

```
$ kinit admin
```

2. サービスインスタンスの公開鍵を取得します。たとえば、**openssl** ユーティリティーを使用する場合は以下を行います。

   a. **service-private.pem** 秘密鍵を生成します。

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....................................+++
e is 65537 (0x10001)
```

   b. 秘密鍵をもとに **service-public.pem** 公開鍵を生成します。

```
$ openssl rsa -in service-private.pem -out service-public.pem -pubout
writing RSA key
```

3. サービスインスタンス vault として非対称 vault を作成し、公開鍵を指定します。

```
$ ipa vault-add secret_vault --service HTTP/webserver.idm.example.com --type
asymmetric --public-key-file service-public.pem
----------------------------
Added vault "secret_vault"
----------------------------
Vault name: secret_vault
Type: asymmetric
Public key: LS0tLS1C...S0tLS0tCg==
Owner users: admin
Vault service: HTTP/webserver.idm.example.com@IDM.EXAMPLE.COM
```

   vault にアーカイブされたパスワードはこの鍵で保護されます。

4. サービスシークレットをサービス vault にアーカイブします。

```
$ ipa vault-archive secret_vault --service HTTP/webserver.idm.example.com --in
private-key-to-an-externally-signed-certificate.pem
----------------------------------
Archived data into vault "secret_vault"
----------------------------------
```

   これにより、サービスインスタンスの公開鍵でシークレットが暗号化されます。

上記の手順を、シークレットを必要とする全サービスインスタンスで繰り返します。サービスインスタンスごとに新規の非対称 vault を作成します。

4.2. IDM サービスインスタンスのサービスシークレットの取得

サービスインスタンスを使用して、ローカルに保存されたサービス秘密キーを使用してサービスコンテナーのシークレットを取得するには、次の手順に従います。

前提条件

- サービス vault を所有するサービスプリンシパルのキータブにアクセスできる (例: HTTP/webserver.idm.example.com)。
- 非対称 vault を作成して vault にシークレットをアーカイブしている。
- サービス vault のシークレットの取得に使用する秘密鍵を使用できる。

手順

1. 管理者としてログインします。

```
$ kinit admin
```

2. サービスの Kerberos チケットを取得します。

```
# kinit HTTP/webserver.idm.example.com -k -t /etc/httpd/conf/ipa.keytab
```

3. サービス vault パスワードを取得します。

```
$ ipa vault-retrieve secret_vault --service HTTP/webserver.idm.example.com --private-key-file service-private.pem --out secret.txt
----------------------------------
Retrieved data from vault "secret_vault"
----------------------------------
```

4.3. シークレットが漏洩した場合の IDM サービス VAULT シークレットの変更

サービスコンテナーのシークレットを変更して、侵害されたサービスインスタンスを隔離するには、次の手順に従います。

前提条件

- **IdM 管理者** パスワードが分かっている。
- サービスシークレットの保存先の 非対称 vault を作成している。
- 新しいシークレットを生成し、そのシークレットにアクセスできる (例: **new-private-key-to-an-externally-signed-certificate.pem** ファイル)。

手順

1. 新規シークレットをサービスインスタンス vault にアーカイブします。

```
$ ipa vault-archive secret_vault --service HTTP/webserver.idm.example.com --in new-private-key-to-an-externally-signed-certificate.pem
```

```
---------------------------------
Archived data into vault "secret_vault"
---------------------------------
```

これにより、vault に保存されている現在のシークレットが上書きされます。

2. 不正アクセスがされていないサービスインスタンスのみで新規シークレットを取得します。詳細は IdM サービスインスタンスのサービスシークレットの取得 を参照し てください。

4.4. 関連情報

- Ansible を使用した IdM サービス vault の管理: シークレットの保存および取得 を参照してください。

第5章 ANSIBLE を使用した IDM サービス VAULT の管理: シークレットの保存および取得

本セクションでは、管理者が **ansible-freeipa vault** モジュールを使用してサービスシークレットを一元的にセキュアに保存する方法を説明します。この例で使用される vault は非対称であるため、これを使用する場合は、管理者は以下の手順を実行する必要があります。

1. **openssl** ユーティリティーなどを使用して秘密鍵を生成する。
2. 秘密鍵をもとに公開鍵を生成する。

サービスシークレットは、管理者が vault にアーカイブする時に公開鍵を使用して暗号化されます。その後、ドメイン内の特定のマシンでホストされるサービスインスタンスが、秘密鍵を使用してシークレットを取得します。シークレットにアクセスできるのは、サービスと管理者のみです。

シークレットが漏洩した場合には、管理者はサービス Vault でシークレットを置き換えて、漏洩されていないサービスインスタンスに配布しなおすことができます。

前提条件

- Key Recovery Authority (KRA) Certificate System コンポーネントが IdM ドメインの 1 つ以上のサーバーにインストールされている。詳細は IdM での Key Recovery Authority (KRA) のインストール を参照してください。

このセクションでは、以下の手順について説明します。

- Ansible を使用して IdM に非対称サービス vault を存在させる手順
- Ansible を使用した非対称 vault への IdM サービスシークレットの保存
- Ansible を使用した IdM サービスのサービスシークレットの取得
- シークレットが漏洩した場合の Ansible での IdM サービス vault シークレットの変更

本手順での以下の用語について説明します。

- **admin** は、サービスパスワードを管理する管理者です。
- **private-key-to-an-externally-signed-certificate.pem** は、サービスシークレットを含むファイルです (ここでは外部署名証明書への秘密鍵)。この秘密鍵と、vault からのシークレットの取得に使用する秘密鍵と混同しないようにしてください。
- **secret_vault** は、サービスシークレット保存向けに作成された vault です。
- **HTTP/webserver1.idm.example.com** は vault の所有者となるサービスです。
- **HTTP/webserver2.idm.example.com** および **HTTP/webserver3.idm.example.com** は vault メンバーサービスです。
- **service-public.pem** は、**password_vault** に保存されているパスワードの暗号化に使用するサービスの公開鍵です。
- **service-private.pem** は、**secret_vault** に保存されているパスワードの復号化に使用するサービスの秘密鍵です。

5.1. ANSIBLE を使用して IDM に非対称サービス VAULT を存在させる手順

以下の手順に従って、Ansible Playbook を使用して 1 つ以上のプライベート Vault を持つサービス vault コンテナーを作成し、機密情報を安全に保存します。以下の手順で使用する例では、管理者は **secret_vault** という名前の非対称 vault を作成します。こうすることで、vault メンバーは、vault のシークレットを取得する際に、秘密鍵を使用して必ず認証することになります。vault メンバーは、IdM クライアントからファイルを取得できます。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible ボールトに **ipaadmin_password** が保存されていることを前提としている。
- **IdM 管理者** パスワードが分かっている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. サービスインスタンスの公開鍵を取得します。たとえば、**openssl** ユーティリティーを使用する場合は以下を行います。
   a. **service-private.pem** 秘密鍵を生成します。

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....................................+++
e is 65537 (0x10001)
```

   b. 秘密鍵をもとに **service-public.pem** 公開鍵を生成します。

```
$ openssl rsa -in service-private.pem -out service-public.pem -pubout
writing RSA key
```

3. オプション: **inventory.file** など、存在しない場合はインベントリーファイルを作成します。

```
$ touch inventory.file
```

4. インベントリーファイルを開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

```
[ipaserver]
server.idm.example.com
```

5. Ansible Playbook ファイル **ensure-asymmetric-vault-is-present.yml** のコピーを作成します。以下に例を示します。

```
$ cp ensure-asymmetric-vault-is-present.yml ensure-asymmetric-service-vault-is-present-copy.yml
```

6. **ensure-asymmetric-vault-is-present-copy.yml** ファイルを開いて編集します。

7. Ansible コントローラーから **server.idm.example.com** サーバーに **service-public.pem** の公開鍵をコピーするタスクを追加します。

8. **ipavault** タスクセクションに以下の変数を設定して、残りのファイルを変更します。

   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **secret_vault** などの **name** 変数を使用して vault の名前を定義します。
   - **vault_type** 変数は **asymmetric** に設定します。
   - **service** 変数は、vault を所有するサービスのプリンシパル (例: **HTTP/webserver1.idm.example.com**) に設定します。
   - **public_key_file** は、公開鍵の場所に設定します。
     以下は、今回の例で使用するように変更した Ansible Playbook ファイルです。

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Copy public key to ipaserver.
    copy:
      src: /path/to/service-public.pem
      dest: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
      mode: 0600
  - name: Add data to vault, from a LOCAL file.
    ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      vault_type: asymmetric
      service: HTTP/webserver1.idm.example.com
      public_key_file: /usr/share/doc/ansible-freeipa/playbooks/vault/service-public.pem
```

9. ファイルを保存します。

10. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file ensure-asymmetric-service-vault-is-present-copy.yml
```

5.2. ANSIBLE を使用した非対称 VAULT へのメンバーサービスの追加

以下の手順に従って、Ansible Playbook を使用してメンバーサービスをサービス vault に追加し、これらすべてが vault に保存されているシークレットを取得できるようにします。以下の手順で使用する例では、IdM の管理者は **HTTP/webserver2.idm.example.com** と **HTTP/webserver3.idm.example.com** のサービスプリンシパルを **HTTP/webserver1.idm.example.com** が所有する **secret_vault** vault に追加します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible ボールトに **ipaadmin_password** が保存されていることを前提としている。
- **IdM 管理者** パスワードが分かっている。
- サービスシークレットの保存先の 非対称 vault を作成している。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. オプション: **inventory.file** など、存在しない場合はインベントリーファイルを作成します。

```
$ touch inventory.file
```

3. インベントリーファイルを開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

```
[ipaserver]
server.idm.example.com
```

4. Ansible Playbook ファイル (**data-archive-in-asymmetric-vault.yml**) を作成します。以下に例を示します。

```
$ cp data-archive-in-asymmetric-vault.yml add-services-to-an-asymmetric-vault.yml
```

5. **data-archive-in-asymmetric-vault-copy.yml** ファイルを開いて編集します。
6. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は vault の名前 (例: **secret_vault**) に設定します。

- **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver1.idm.example.com**)。
- services 変数を使用して、vault シークレットにアクセスできる **サービス** を定義します。
- **action** 変数は **member** に設定します。
  今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      services:
      - HTTP/webserver2.idm.example.com
      - HTTP/webserver3.idm.example.com
      action: member
```

7. ファイルを保存します。
8. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file add-services-to-an-asymmetric-vault.yml
```

5.3. ANSIBLE を使用した非対称 VAULT への IDM サービスシークレットの保存

以下の手順に従って、Ansible Playbook を使用して、サービス vault にシークレットを保存し、サービスが後で取得できるようにします。以下の手順で使用する例では、管理者は **secret_vault** という名前の非対称 vault にシークレットが含まれる **PEM** ファイルを保存します。こうすることで、サービスは、vault からシークレットを取得する際に、秘密鍵を使用して必ず認証することになります。vault メンバーは、IdM クライアントからファイルを取得できます。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible ボールトに **ipaadmin_password** が保存されていることを前提としている。

- **IdM 管理者** パスワードが分かっている。
- サービスシークレットの保存先の 非対称 vault を作成している。
- シークレットが **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem** ファイルなど、Ansible コントローラーのローカルに保存されている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. オプション: **inventory.file** など、存在しない場合はインベントリーファイルを作成します。

```
$ touch inventory.file
```

3. インベントリーファイルを開き、**[ipaserver]** セクションに、設定する IdM サーバーを定義します。たとえば、Ansible に対して **server.idm.example.com** を設定するように指示するには、次のコマンドを実行します。

```
[ipaserver]
server.idm.example.com
```

4. Ansible Playbook ファイル (**data-archive-in-asymmetric-vault.yml**) を作成します。以下に例を示します。

```
$ cp data-archive-in-asymmetric-vault.yml data-archive-in-asymmetric-vault-copy.yml
```

5. **data-archive-in-asymmetric-vault-copy.yml** ファイルを開いて編集します。
6. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は vault の名前 (例: **secret_vault**) に設定します。
   - **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver1.idm.example.com**)。
   - **in** 変数は "**{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem')| b64encode }}**" に設定します。この設定では、Ansible が IdM サーバーではなく、Ansible コントローラーの作業ディレクトリーから秘密鍵のあるファイルを取得するようになります。
   - **action** 変数は **member** に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
```

```
    - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      in: "{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem') | b64encode }}"
      action: member
```

7. ファイルを保存します。

8. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-archive-in-asymmetric-vault-copy.yml
```

5.4. ANSIBLE を使用した IDM サービスのサービスシークレットの取得

以下の手順に従って、Ansible Playbook を使用してサービスの代わりにサービス vault からシークレットを取得します。以下の手順で使用する例では、Playbook を実行して **secret_vault** という名前の **PEM** ファイルを取得し、Ansible インベントリーファイルに **ipaservers** として記載されている全ホストの指定の場所に保存します。

サービスは、キータブを使用して IdM に対して、さらに秘密鍵を使用して vault に対して認証を実行します。**ansible-freeipa** がインストールされている IdM クライアントから、サービスの代わりにファイルを取得できます。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible ボールトに **ipaadmin_password** が保存されていることを前提としている。
- IdM 管理者パスワードを把握している。
- サービスシークレットの保存先の 非対称 vault を作成している。
- vault にシークレットをアーカイブ している。
- Ansible コントローラーで **private_key_file** 変数が指定する場所にサービス vault のシークレットの取得に使用する秘密鍵が保存されている。

手順

1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. オプション: **inventory.file** など、存在しない場合はインベントリーファイルを作成します。

```
$ touch inventory.file
```

3. インベントリーファイルを開き、以下のホストを定義します。

- **[ipaserver]** セクションで、使用する IdM サーバーを定義します。
- **[webservers]** セクションで、シークレットを取得するホストを定義します。たとえば Ansible に対して **webserver1.idm.example.com**、**webserver2.idm.example.com** および **webserver3.idm.example.com** にシークレットを取得するように指示するには以下を実行します。

```
[ipaserver]
server.idm.example.com

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
webserver3.idm.example.com
```

4. Ansible Playbook ファイル (**retrieve-data-asymmetric-vault.yml**) のコピーを作成します。以下に例を示します。

```
$ cp retrieve-data-asymmetric-vault.yml retrieve-data-asymmetric-vault-copy.yml
```

5. **retrieve-data-asymmetric-vault-copy.yml** ファイルを開いて編集します。

6. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。

- **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
- **name** 変数は vault の名前 (例: **secret_vault**) に設定します。
- **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver1.idm.example.com**)。
- **private_key_file** 変数は、サービス vault のシークレットの取得に使用する秘密鍵の場所に設定します。
- **out** 変数は、現在の作業ディレクトリーなど、**private-key-to-an-externally-signed-certificate.pem** シークレットの取得先となる IdM サーバーの場所に設定します。
- **action** 変数は **member** に設定します。
  今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
```

```
  - name: Retrieve data from the service vault
    ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      vault_type: asymmetric
      private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
      out: private-key-to-an-externally-signed-certificate.pem
      state: retrieved
```

7. Playbook に、IdM サーバーから Ansible コントローラーにデータファイルを取得するセクションを追加します。

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false
  tasks:
[...]
  - name: Retrieve data file
    fetch:
      src: private-key-to-an-externally-signed-certificate.pem
      dest: ./
      flat: yes
      mode: 0600
```

8. インベントリーファイルの **webservers** セクションに記載されている Web サーバーに、Ansible コントローラーから取得した **private-key-to-an-externally-signed-certificate.pem** ファイルを転送するセクションを Playbook に追加します。

```
---
- name: Send data file to webservers
  become: no
  gather_facts: no
  hosts: webservers
  tasks:
  - name: Send data to webservers
    copy:
      src: private-key-to-an-externally-signed-certificate.pem
      dest: /etc/pki/tls/private/httpd.key
      mode: 0444
```

9. ファイルを保存します。

10. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-
data-asymmetric-vault-copy.yml
```

5.5. シークレットが漏洩した場合の ANSIBLE での IDM サービス VAULT シークレットの変更

以下の手順に従って、サービスインスタンスが危険にさらされたときに Ansible Playbook を再利用し、

サービス vault に保存されているシークレットを変更します。以下の例のシナリオでは、シークレットを保存する非対称 vault への鍵は漏洩されておらず、**webserver3.idm.example.com** で取得したシークレットの情報が漏洩した場合を前提としています。この例では、管理者は 非対称 vault でシークレットを保存する時 および IdM ホストに非対称 vault からシークレットを取得する時 に Ansible Playbook を再利用します。この手順のはじめに、IdM 管理者は新規シークレットを含む新しい **PEM** ファイルを非対称 vault に保存し、不正アクセスのあった Web サーバー (**webserver3.idm.example.com**) に新しいシークレットを配置しないようにインベントリーファイルを調節し、もう一度この 2 つの手順を実行します。

前提条件

- 次の要件を満たすように Ansible コントロールノードを設定している。
  - Ansible バージョン 2.14 以降を使用している。
  - Ansible コントローラーに **ansible-freeipa** パッケージがインストールされている。
  - **~/MyPlaybooks/** ディレクトリーに、IdM サーバーの完全修飾ドメイン名 (FQDN) を使用して Ansible インベントリーファイル を作成している (この例の場合)。
  - この例では、**secret.yml** Ansible ボールトに **ipaadmin_password** が保存されていることを前提としている。
- **IdM 管理者** パスワードが分かっている。
- サービスシークレットの保存先の 非対称 vault を作成している。
- IdM ホストで実行中の Web サービスの **httpd** 鍵を新たに生成し、不正アクセスのあった以前の鍵を置き換えている。
- 新しい **httpd** キーが **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem** ファイルなど、Ansible コントローラーのローカルに保存されている。

手順

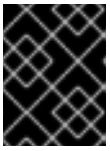
1. **/usr/share/doc/ansible-freeipa/playbooks/vault** ディレクトリーに移動します。

```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. インベントリーファイルを開き、以下のホストが正しく定義されていることを確認します。
   - **[ipaserver]** セクションの IdM サーバー
   - **[webservers]** セクションのシークレット取得先のホスト。たとえば Ansible に対して **webserver1.idm.example.com** と **webserver2.idm.example.com** にシークレットを取得するように指示するには、以下を入力します。

```
[ipaserver]
server.idm.example.com

[webservers]
webserver1.idm.example.com
webserver2.idm.example.com
```



重要

このリストに不正アクセスのあった Web サーバーが含まれないようにしてください (現在の例では **webserver3.idm.example.com**)。

3. **data-archive-in-asymmetric-vault-copy.yml** ファイルを開いて編集します。
4. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は vault の名前 (例: **secret_vault**) に設定します。
   - **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver.idm.example.com**)。
   - **in** 変数は "**{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem')| b64encode }}**" に設定します。この設定では、Ansible が IdM サーバーではなく、Ansible コントローラーの作業ディレクトリーから秘密鍵のあるファイルを取得するようになります。
   - **action** 変数は **member** に設定します。
     今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- name: Tests
  hosts: ipaserver
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver.idm.example.com
      in: "{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem') | b64encode
}}"
      action: member
```

5. ファイルを保存します。
6. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file data-
archive-in-asymmetric-vault-copy.yml
```

7. **retrieve-data-asymmetric-vault-copy.yml** ファイルを開いて編集します。
8. ファイルを変更するには、**ipavault** タスクセクションに以下の変数を設定します。
   - **ipaadmin_password** 変数は IdM 管理者パスワードに設定します。
   - **name** 変数は vault の名前 (例: **secret_vault**) に設定します。

- **service** 変数は vault のサービス所有者に設定します (例: **HTTP/webserver1.idm.example.com**)。
- **private_key_file** 変数は、サービス vault のシークレットの取得に使用する秘密鍵の場所に設定します。
- **out** 変数は、現在の作業ディレクトリーなど、**new-private-key-to-an-externally-signed-certificate.pem** シークレットの取得先となる IdM サーバーの場所に設定します。
- **action** 変数は **member** に設定します。
  今回の例で使用するように変更した Ansible Playbook ファイル:

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: no
  gather_facts: false

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml
  tasks:
  - name: Retrieve data from the service vault
    ipavault:
      ipaadmin_password: "{{ ipaadmin_password }}"
      name: secret_vault
      service: HTTP/webserver1.idm.example.com
      vault_type: asymmetric
      private_key: "{{ lookup('file', 'service-private.pem') | b64encode }}"
      out: new-private-key-to-an-externally-signed-certificate.pem
      state: retrieved
```

9. Playbook に、IdM サーバーから Ansible コントローラーにデータファイルを取得するセクションを追加します。

```
---
- name: Retrieve data from vault
  hosts: ipaserver
  become: true
  gather_facts: false
  tasks:
[...]
  - name: Retrieve data file
    fetch:
      src: new-private-key-to-an-externally-signed-certificate.pem
      dest: ./
      flat: yes
      mode: 0600
```

10. インベントリーファイルの **webservers** セクションに記載されている Web サーバーに、Ansible コントローラーから取得した **new-private-key-to-an-externally-signed-certificate.pem** ファイルを転送するセクションを Playbook に追加します。

```
---
- name: Send data file to webservers
  become: true
```

```
gather_facts: no
hosts: webservers
tasks:
- name: Send data to webservers
  copy:
    src: new-private-key-to-an-externally-signed-certificate.pem
    dest: /etc/pki/tls/private/httpd.key
    mode: 0444
```

11. ファイルを保存します。

12. Playbook を実行します。

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory.file retrieve-data-asymmetric-vault-copy.yml
```

5.6. 関連情報

- **/usr/share/doc/ansible-freeipa/** ディレクトリーの README-vault.md Markdown ファイルを参照してください。

- **/usr/share/doc/ansible-freeipa/playbooks/vault/** ディレクトリーのサンプルの Playbook を参照してください。