Red Hat Enterprise Linux 9

Travailler avec des coffres-forts dans la gestion de l'identité

Stockage et gestion des données sensibles dans l'IdM

Last Updated: 2024-01-18

Red Hat Enterprise Linux 9 Travailler avec des coffres-forts dans la gestion de l'identité

Stockage et gestion des données sensibles dans l'IdM

Notice légale

Résumé

Un coffre-fort est un emplacement sécurisé dans Red Hat Identity Management (IdM) pour stocker, récupérer et partager des données sensibles, telles que les informations d'authentification pour les services. Vous pouvez gérer les espaces de stockage à l'aide de la ligne de commande ou des Playbooks Ansible.

Table des matières

RENDRE L'OPEN SOURCE PLUS INCLUSIF

Red Hat s'engage à remplacer les termes problématiques dans son code, sa documentation et ses propriétés Web. Nous commençons par ces quatre termes : master, slave, blacklist et whitelist. En raison de l'ampleur de cette entreprise, ces changements seront mis en œuvre progressivement au cours de plusieurs versions à venir. Pour plus de détails, voir le message de notre directeur technique Chris Wright.

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT

Nous apprécions vos commentaires sur notre documentation. Faites-nous savoir comment nous pouvons l'améliorer.

Soumettre des commentaires sur des passages spécifiques

1. Consultez la documentation au format **Multi-page HTML** et assurez-vous que le bouton **Feedback** apparaît dans le coin supérieur droit après le chargement complet de la page.
2. Utilisez votre curseur pour mettre en évidence la partie du texte que vous souhaitez commenter.
3. Cliquez sur le bouton **Add Feedback** qui apparaît près du texte en surbrillance.
4. Ajoutez vos commentaires et cliquez sur **Submit**.

Soumettre des commentaires via Bugzilla (compte requis)

1. Connectez-vous au site Web de Bugzilla.
2. Sélectionnez la version correcte dans le menu **Version**.
3. Saisissez un titre descriptif dans le champ **Summary**.
4. Saisissez votre suggestion d'amélioration dans le champ **Description**. Incluez des liens vers les parties pertinentes de la documentation.
5. Cliquez sur **Submit Bug**.

CHAPITRE 1. COFFRES-FORTS DANS L'IDM

Ce chapitre décrit les chambres fortes dans la gestion des identités (IdM). Il présente les sujets suivants :

- Le concept de la chambre forte .
- Les différents rôles associés à un coffre-fort .
- Les différents types de chambres fortes disponibles dans l'IdM en fonction du niveau de sécurité et de contrôle d'accès.
- Les différents types de coffres disponibles dans l'IdM en fonction de la propriété .
- Le concept des conteneurs de la chambre forte .
- Commandes de base pour la gestion des coffres-forts dans IdM .
- Installation de l'autorité de récupération des clés (KRA), qui est une condition préalable à l'utilisation des chambres fortes dans IdM.

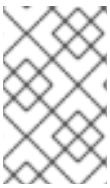
1.1. LES CHAMBRES FORTES ET LEURS AVANTAGES

Un coffre-fort est une fonction utile pour les utilisateurs de la gestion des identités (IdM) qui souhaitent conserver toutes leurs données sensibles en un seul endroit, de manière sûre et pratique. Cette section explique les différents types d'espaces de stockage et leurs utilisations, ainsi que le choix de l'espace de stockage en fonction de vos besoins.

Un coffre-fort est un emplacement sécurisé dans (IdM) pour le stockage, l'extraction, le partage et la récupération d'un secret. Un secret est une donnée sensible sur le plan de la sécurité, généralement des identifiants d'authentification, à laquelle seul un groupe limité de personnes ou d'entités peut avoir accès. Par exemple, les secrets comprennent

- mots de passe
- NIP
- clés SSH privées

Un coffre-fort est comparable à un gestionnaire de mots de passe. Tout comme un gestionnaire de mots de passe, un coffre-fort exige généralement que l'utilisateur génère et mémorise un mot de passe principal pour déverrouiller et accéder à toutes les informations stockées dans le coffre-fort. Toutefois, un utilisateur peut également décider d'opter pour un coffre-fort standard. Dans ce cas, l'utilisateur n'a pas besoin de saisir de mot de passe pour accéder aux secrets stockés dans le coffre-fort.



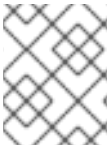
NOTE

L'objectif des chambres fortes dans l'IdM est de stocker les informations d'authentification qui vous permettent de vous authentifier auprès de services externes non liés à l'IdM.

Les autres caractéristiques importantes des chambres fortes IdM sont les suivantes :

- Les chambres fortes ne sont accessibles qu'au propriétaire de la chambre forte et aux utilisateurs de l'IdM que le propriétaire de la chambre forte sélectionne comme membres de la chambre forte. En outre, l'administrateur IdM a accès à l'espace de stockage.

- Si un utilisateur ne dispose pas de privilèges suffisants pour créer un coffre-fort, un administrateur IdM peut créer le coffre-fort et désigner l'utilisateur comme son propriétaire.
- Les utilisateurs et les services peuvent accéder aux secrets stockés dans un coffre-fort à partir de n'importe quelle machine inscrite dans le domaine IdM.
- Un coffre-fort ne peut contenir qu'un seul secret, par exemple un fichier. Toutefois, le fichier lui-même peut contenir plusieurs secrets tels que des mots de passe, des tableaux de clés ou des certificats.



NOTE

Vault n'est disponible qu'à partir de la ligne de commande IdM (CLI), et non à partir de l'interface Web IdM.

1.2. PROPRIÉTAIRES, MEMBRES ET ADMINISTRATEURS DE CHAMBRES FORTES

La gestion de l'identité (IdM) distingue les types d'utilisateurs de la chambre forte suivants :

Propriétaire de la chambre forte

Le propriétaire d'un coffre-fort est un utilisateur ou un service qui dispose de privilèges de gestion de base sur le coffre-fort. Par exemple, un propriétaire de coffre-fort peut modifier les propriétés du coffre-fort ou ajouter de nouveaux membres au coffre-fort.
Chaque chambre forte doit avoir au moins un propriétaire. Un coffre-fort peut également avoir plusieurs propriétaires.

Membre de la chambre forte

Un membre d'un coffre-fort est un utilisateur ou un service qui peut accéder à un coffre-fort créé par un autre utilisateur ou service.

Administrateur de la chambre forte

Les administrateurs de coffre-fort ont un accès illimité à tous les coffres-forts et sont autorisés à effectuer toutes les opérations sur les coffres-forts.

NOTE

Les chambres fortes symétriques et asymétriques sont protégées par un mot de passe ou une clé et appliquent des règles spéciales de contrôle d'accès (voir Types de chambres fortes). L'administrateur doit respecter ces règles pour :

- Secrets d'accès dans les chambres fortes symétriques et asymétriques.
- Modifier ou réinitialiser le mot de passe ou la clé du coffre-fort.

Un administrateur de coffre-fort est un utilisateur disposant du privilège **Vault Administrators**. Dans le contexte du contrôle d'accès basé sur les rôles (RBAC) dans IdM, un privilège est un groupe de permissions que vous pouvez appliquer à un rôle.

Utilisateur de la chambre forte

L'utilisateur de l'espace de stockage représente l'utilisateur dans le conteneur duquel se trouve l'espace de stockage. L'information **Vault user** est affichée dans la sortie de commandes spécifiques, telles que **ipa vault-show**:

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

Pour plus d'informations sur les conteneurs de coffre-fort et les coffres-forts utilisateur, voir Conteneurs de coffre-fort.

Ressources supplémentaires

- Voir voûtes standard, symétriques et asymétriques pour plus de détails sur les types de voûtes.

1.3. VOÛTES STANDARD, SYMÉTRIQUES ET ASYMÉTRIQUES

En fonction du niveau de sécurité et de contrôle d'accès, l'IdM classe les chambres fortes dans les types suivants :

Voûtes standard

Les propriétaires et les membres des chambres fortes peuvent archiver et récupérer les secrets sans avoir à utiliser de mot de passe ou de clé.

Voûtes symétriques

Les secrets contenus dans le coffre-fort sont protégés par une clé symétrique. Les propriétaires et les membres du coffre-fort peuvent archiver et récupérer les secrets, mais ils doivent fournir le mot de passe du coffre-fort.

Voûtes asymétriques

Les secrets de la chambre forte sont protégés par une clé asymétrique. Les utilisateurs archivent le secret à l'aide d'une clé publique et le récupèrent à l'aide d'une clé privée. Les membres du coffre-fort ne peuvent qu'archiver les secrets, tandis que les propriétaires du coffre-fort peuvent faire les deux, archiver et récupérer les secrets.

1.4. COFFRES-FORTS D'UTILISATEURS, DE SERVICES ET PARTAGÉS

En fonction de la propriété, l'IdM classe les chambres fortes en plusieurs types. Le tableau ci-dessous contient des informations sur chaque type, son propriétaire et son utilisation.

Tableau 1.1. Coffres-forts de l'IdM basés sur la propriété

Type	Description	Propriétaire	Note
User vault	Un coffre-fort privé pour un utilisateur	Un seul utilisateur	Tout utilisateur peut posséder un ou plusieurs coffres-forts d'utilisateur si l'administrateur IdM l'autorise
Service vault	Un caveau privé pour un service	Un seul service	Tout service peut posséder un ou plusieurs coffres-forts d'utilisateur si l'administrateur IdM l'autorise

Type	Description	Propriétaire	Note
Shared vault	Un coffre-fort partagé par plusieurs utilisateurs et services	L'administrateur du coffre-fort qui a créé le coffre-fort	Les utilisateurs et les services peuvent posséder un ou plusieurs coffres-forts d'utilisateur si l'administrateur IdM l'autorise. Les administrateurs de coffre-fort autres que celui qui a créé le coffre-fort ont également un accès complet au coffre-fort.

1.5. CONTENEURS À CLAIRE-VOIE

Un conteneur d'espace de stockage est un ensemble d'espaces de stockage. Le tableau ci-dessous répertorie les conteneurs de coffre-fort par défaut fournis par Identity Management (IdM).

Tableau 1.2. Conteneurs de coffre-fort par défaut dans l'IdM

Type	Description	Objectif
Conteneur utilisateur	Un conteneur privé pour un utilisateur	Stocke les coffres-forts d'un utilisateur particulier
Conteneur de services	Un conteneur privé pour un service	Stocke les coffres-forts d'un service particulier
Conteneur partagé	Un conteneur pour plusieurs utilisateurs et services	Stocke des coffres-forts qui peuvent être partagés par plusieurs utilisateurs ou services

L'IdM crée automatiquement des conteneurs d'utilisateurs et de services pour chaque utilisateur ou service lorsque le premier coffre-fort privé de l'utilisateur ou du service est créé. Après la suppression de l'utilisateur ou du service, l'IdM supprime le conteneur et son contenu.

1.6. COMMANDES DE BASE DU COFFRE-FORT IDM

Cette section décrit les commandes de base que vous pouvez utiliser pour gérer les coffres-forts de la gestion des identités (IdM). Le tableau ci-dessous contient une liste des commandes **ipa vault-*** avec l'explication de leur fonction.

NOTE

Avant d'exécuter une commande **ipa vault-***, installez le composant du système de certificats Key Recovery Authority (KRA) sur un ou plusieurs serveurs de votre domaine IdM. Pour plus de détails, voir Installation de l'autorité de recouvrement des clés dans IdM .

Tableau 1.3. Commandes de base du coffre-fort de l'IdM avec explications

Commandement	Objectif
ipa help vault	Affiche des informations conceptuelles sur les espaces de stockage IdM et des exemples de commandes d'espaces de stockage.
ipa vault-add --help, **ipa vault-find --help**	L'ajout de l'option **--help** à une commande **ipa vault-*** spécifique permet d'afficher les options et l'aide détaillée disponibles pour cette commande.

ipa vault-show user_vault --user idm_user	Lorsque vous accédez à un coffre-fort en tant que membre du coffre-fort, vous devez spécifier le propriétaire du coffre-fort. Si vous n'indiquez pas le propriétaire de l'espace de stockage, l'IdM vous informe qu'il n'a pas trouvé l'espace de stockage :
	`[admin@server ~]$ ipa vault-show user_vault`
	`ipa: ERROR: user_vault: vault not found`

ipa vault-show shared_vault --shared	Lorsque vous accédez à un coffre-fort partagé, vous devez spécifier que le coffre-fort auquel vous voulez accéder est un coffre-fort partagé. Sinon, l'IdM vous informe qu'il n'a pas trouvé l'espace de stockage :
	`[admin@server ~]$ ipa vault-show shared_vault`
	`ipa: ERROR: shared_vault: vault not found`

1.7. INSTALLATION DE L'AUTORITÉ DE RECOUVREMENT DES CLÉS DANS IDM

Cette section explique comment activer les chambres fortes dans la gestion des identités (IdM) en installant le composant Système de certificats (CS) de l'Autorité de recouvrement des clés (KRA) sur un serveur IdM spécifique.

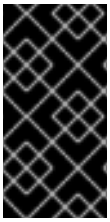
Conditions préalables

- Vous êtes connecté en tant que **root** sur le serveur IdM.
- Une autorité de certification IdM est installée sur le serveur IdM.
- Vous avez les références **Directory Manager**.

Procédure

- Installer l'ARK :

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



IMPORTANT

Vous pouvez installer le premier ARK d'un cluster IdM sur un réplica caché. Toutefois, l'installation d'ARK supplémentaires nécessite l'activation temporaire du réplica caché avant d'installer le clone de l'ARK sur un réplica non caché. Vous pouvez ensuite masquer à nouveau le réplica initialement masqué.

NOTE

Pour que le service de coffre-fort soit hautement disponible et résilient, installez l'ARK sur deux serveurs IdM ou plus. La maintenance de plusieurs serveurs KRA permet d'éviter les pertes de données.

Ressources supplémentaires

- Voir Rétrograder ou promouvoir des répliques cachées.
- Voir Le mode réplique caché.

CHAPITRE 2. UTILISATION DES COFFRES-FORTS DES UTILISATEURS IDM : STOCKAGE ET RÉCUPÉRATION DES SECRETS

Ce chapitre décrit l'utilisation des coffres-forts d'utilisateurs dans la gestion de l'identité. Il décrit en particulier comment un utilisateur peut stocker un secret dans un coffre-fort IdM et comment il peut le récupérer. L'utilisateur peut effectuer le stockage et l'extraction à partir de deux clients IdM différents.

Conditions préalables

- Le composant du système de certificats de l'autorité de récupération des clés (KRA) a été installé sur un ou plusieurs serveurs de votre domaine IdM. Pour plus de détails, voir Installation de l'autorité de recouvrement des clés dans IdM.

2.1. STOCKER UN SECRET DANS UN COFFRE-FORT D'UTILISATEUR

Cette section montre comment un utilisateur peut créer un conteneur de coffre-fort avec un ou plusieurs coffres-forts privés pour stocker en toute sécurité des fichiers contenant des informations sensibles. Dans l'exemple utilisé dans la procédure ci-dessous, l'utilisateur **idm_user** crée un coffre-fort de type standard. Le type de coffre-fort standard garantit que **idm_user** n'aura pas à s'authentifier pour accéder au fichier. **idm_user** pourra récupérer le fichier à partir de n'importe quel client IdM auquel l'utilisateur est connecté.

Dans la procédure :

- **idm_user** est l'utilisateur qui souhaite créer l'espace de stockage.
- **my_vault** est le coffre-fort utilisé pour stocker le certificat de l'utilisateur.
- Le type de coffre-fort est **standard**, de sorte que l'accès au certificat archivé ne nécessite pas que l'utilisateur fournisse un mot de passe pour le coffre-fort.
- **secret.txt** est le fichier contenant le certificat que l'utilisateur souhaite stocker dans le coffre-fort.

Conditions préalables

- Vous connaissez le mot de passe de **idm_user**.
- Vous êtes connecté à un hôte qui est un client IdM.

Procédure

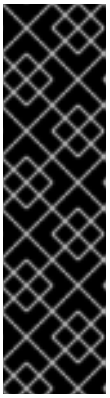
1. Obtenir le ticket d'octroi de ticket Kerberos (TGT) pour **idm_user**:

```
$ kinit idm_user
```

2. Utilisez la commande **ipa vault-add** avec l'option **--type standard** pour créer un coffre-fort standard :

```
$ 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
```



IMPORTANT

Assurez-vous que le premier coffre-fort d'un utilisateur est créé par le même utilisateur. La création du premier coffre-fort d'un utilisateur crée également le conteneur de coffre-fort de l'utilisateur. L'agent de la création devient le propriétaire du conteneur de l'espace de stockage.

Par exemple, si un autre utilisateur, tel que **admin**, crée le premier coffre-fort d'utilisateur pour **user1**, le propriétaire du conteneur de coffre-fort de l'utilisateur sera également **admin**, et **user1** ne pourra pas accéder au coffre-fort d'utilisateur ou créer de nouveaux coffres-forts d'utilisateur.

3. Utilisez la commande **ipa vault-archive** avec l'option **--in** pour archiver le fichier **secret.txt** dans l'espace de stockage :

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

2.2. RÉCUPÉRATION D'UN SECRET DANS LE COFFRE-FORT D'UN UTILISATEUR

En tant que gestionnaire d'identité (IdM), vous pouvez récupérer un secret de votre coffre-fort privé d'utilisateur sur n'importe quel client IdM auquel vous êtes connecté.

Cette section montre comment récupérer, en tant qu'utilisateur IdM nommé **idm_user**, un secret du coffre-fort privé de l'utilisateur nommé **my_vault** sur **idm_client.idm.example.com**.

Conditions préalables

- **idm_user** est le propriétaire de **my_vault**.
- **idm_user** a archivé un secret dans la chambre forte .
- **my_vault** est un coffre-fort standard, ce qui signifie que **idm_user** ne doit pas entrer de mot de passe pour accéder au contenu du coffre-fort.

Procédure

1. SSH à **idm_client** en tant que **idm_user**:

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

2. Connectez-vous en tant que **idm_user**:

```
$ kinit user
```

3. Utilisez la commande **ipa vault-retrieve --out** avec l'option **--out** pour récupérer le contenu de la chambre forte et l'enregistrer dans le fichier **secret_exported.txt**.

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

2.3. RESSOURCES SUPPLÉMENTAIRES

- Voir Utiliser Ansible pour gérer les coffres-forts des services IdM : stocker et récupérer les secrets.

CHAPITRE 3. UTILISER ANSIBLE POUR GÉRER LES COFFRES-FORTS DES UTILISATEURS IDM : STOCKER ET RÉCUPÉRER LES SECRETS

Ce chapitre décrit comment gérer les coffres-forts des utilisateurs dans Identity Management à l'aide du module Ansible **vault**. Plus précisément, il décrit comment un utilisateur peut utiliser les playbooks Ansible pour effectuer les trois actions consécutives suivantes :

- Créer un coffre-fort d'utilisateur dans IdM .
- Conserver un secret dans la chambre forte .
- Récupérer un secret dans le coffre .

L'utilisateur peut effectuer le stockage et l'extraction à partir de deux clients IdM différents.

Conditions préalables

- Le composant du système de certificats de l'autorité de récupération des clés (KRA) a été installé sur un ou plusieurs serveurs de votre domaine IdM. Pour plus de détails, voir Installation de l'autorité de recouvrement des clés dans IdM.

3.1. ASSURER LA PRÉSENCE D'UN COFFRE-FORT UTILISATEUR STANDARD DANS IDM À L'AIDE D'ANSIBLE

Cette section montre comment un utilisateur de la gestion des identités (IdM) peut utiliser un playbook Ansible pour créer un conteneur d'espace de stockage avec un ou plusieurs espaces de stockage privés pour stocker en toute sécurité des informations sensibles. Dans l'exemple utilisé dans la procédure ci-dessous, l'utilisateur **idm_user** crée un espace de stockage de type standard nommé **my_vault**. Le type de coffre-fort standard garantit que **idm_user** n'aura pas à s'authentifier pour accéder au fichier. **idm_user** pourra récupérer le fichier à partir de n'importe quel client IdM auquel l'utilisateur est connecté.

Conditions préalables

- Vous avez installé le paquet ansible-freeipa sur le contrôleur Ansible, c'est-à-dire l'hôte sur lequel vous exécutez les étapes de la procédure.
- Vous connaissez le mot de passe de **idm_user**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

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

2. Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ touch inventory.file
```

3. Ouvrez **inventory.file** et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

4. Faites une copie du fichier **ensure-standard-vault-is-present.yml** Ansible playbook. Par exemple :

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

5. Ouvrez le fichier **ensure-standard-vault-is-present-copy.yml** pour le modifier.

6. Adaptez le fichier en définissant les variables suivantes dans la section **ipavault** task :

   - Fixer la variable **ipaadmin_principal** à **idm_user**.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de **idm_user**.
   - Fixer la variable **user** à **idm_user**.
   - Fixer la variable **name** à **my_vault**.
   - Fixer la variable **vault_type** à **standard**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- 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. Enregistrer le fichier.

8. Exécutez le manuel de jeu :

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

3.2. ARCHIVAGE D'UN SECRET DANS UN COFFRE-FORT UTILISATEUR STANDARD DANS IDM À L'AIDE D'ANSIBLE

Cette section montre comment un utilisateur de la gestion des identités (IdM) peut utiliser un playbook Ansible pour stocker des informations sensibles dans un coffre-fort personnel. Dans l'exemple utilisé, l'utilisateur **idm_user** archive un fichier contenant des informations sensibles nommé **password.txt** dans un coffre-fort nommé **my_vault**.

Conditions préalables

- Vous avez installé le paquet ansible-freeipa sur le contrôleur Ansible, c'est-à-dire l'hôte sur lequel vous exécutez les étapes de la procédure.
- Vous connaissez le mot de passe de **idm_user**.
- **idm_user** est le propriétaire, ou au moins un membre utilisateur de **my_vault**.
- Vous avez accès à **password.txt**, le secret que vous voulez archiver dans **my_vault**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

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

2. Ouvrez votre fichier d'inventaire et assurez-vous que le serveur IdM que vous souhaitez configurer est répertorié dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

3. Faites une copie du fichier **data-archive-in-symmetric-vault.yml** Ansible playbook, mais remplacez "symmetric" par "standard". Par exemple :

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

4. Ouvrez le fichier **data-archive-in-standard-vault-copy.yml** pour le modifier.
5. Adaptez le fichier en définissant les variables suivantes dans la section **ipavault** task :
   - Fixer la variable **ipaadmin_principal** à **idm_user**.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de **idm_user**.
   - Fixer la variable **user** à **idm_user**.
   - Fixer la variable **name** à **my_vault**.
   - Définissez la variable **in** avec le chemin d'accès complet au fichier contenant des informations sensibles.
   - Fixer la variable **action** à **member**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- 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. Enregistrer le fichier.
7. Exécutez le manuel de jeu :

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

3.3. RÉCUPÉRER UN SECRET À PARTIR D'UN COFFRE-FORT D'UTILISATEUR STANDARD DANS IDM EN UTILISANT ANSIBLE

Cette section montre comment un utilisateur de la gestion des identités (IdM) peut utiliser un playbook Ansible pour récupérer un secret dans le coffre-fort personnel de l'utilisateur. Dans l'exemple utilisé dans la procédure ci-dessous, l'utilisateur **idm_user** récupère un fichier contenant des données sensibles à partir d'un coffre-fort de type standard nommé **my_vault** sur un client IdM nommé **host01**. **idm_user** n'a pas besoin de s'authentifier pour accéder au fichier. **idm_user** peut utiliser Ansible pour récupérer le fichier à partir de n'importe quel client IdM sur lequel Ansible est installé.

Conditions préalables

- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.
  - L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.
- Vous connaissez le mot de passe de **idm_user**.
- **idm_user** est le propriétaire de **my_vault**.
- **idm_user** a stocké un secret dans **my_vault**.
- Ansible peut écrire dans le répertoire de l'hôte IdM dans lequel vous souhaitez récupérer le secret.
- **idm_user** peut lire le répertoire de l'hôte IdM dans lequel vous souhaitez récupérer le secret.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

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

2. Ouvrez votre fichier d'inventaire et mentionnez, dans une section clairement définie, le client IdM sur lequel vous souhaitez récupérer le secret. Par exemple, pour demander à Ansible de récupérer le secret sur **host01.idm.example.com**, entrez :

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

3. Effectuez une copie du fichier **retrive-data-symmetric-vault.yml** Ansible playbook. Remplacez "symétrique" par "standard". Par exemple :

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

4. Ouvrez le fichier **retrieve-data-standard-vault.yml-copy.yml** pour le modifier.

5. Adapter le fichier en fixant la variable **hosts** à **ipahost**.

6. Adaptez le fichier en définissant les variables suivantes dans la section **ipavault** task :

   - Fixer la variable **ipaadmin_principal** à **idm_user**.
   - Définissez la variable **ipaadmin_password** avec le mot de passe de **idm_user**.
   - Fixer la variable **user** à **idm_user**.
   - Fixer la variable **name** à **my_vault**.
   - Définissez la variable **out** avec le chemin complet du fichier dans lequel vous souhaitez exporter le secret.
   - Fixer la variable **state** à **retrieved**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- 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. Enregistrer le fichier.

8. Exécutez le manuel de jeu :

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

Verification steps

1. **SSH** à **host01** comme **user01**:

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

2. Afficher le fichier spécifié par la variable **out** dans le fichier playbook Ansible :

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

Vous pouvez maintenant voir le secret exporté.

- Pour plus d'informations sur l'utilisation d'Ansible pour gérer les coffres-forts IdM et les secrets d'utilisateur, ainsi que sur les variables des playbooks, consultez le fichier README-vault.md Markdown disponible dans le répertoire **/usr/share/doc/ansible-freeipa/** et les exemples de playbooks disponibles dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault/**.

CHAPITRE 4. GESTION DES SECRETS DU SERVICE IDM : STOCKAGE ET RÉCUPÉRATION DES SECRETS

Cette section montre comment un administrateur peut utiliser le module **ansible-freeipa vault** pour stocker en toute sécurité un secret de service dans un emplacement centralisé. Le coffre-fort utilisé dans l'exemple est asymétrique, ce qui signifie que pour l'utiliser, l'administrateur doit effectuer les étapes suivantes :

1. Générez une clé privée en utilisant, par exemple, l'utilitaire **openssl**.
2. Générer une clé publique à partir de la clé privée.

Le secret de service est crypté avec la clé publique lorsqu'un administrateur l'archive dans le coffre-fort. Ensuite, une instance de service hébergée sur une machine spécifique du domaine récupère le secret à l'aide de la clé privée. Seuls le service et l'administrateur sont autorisés à accéder au secret.

Si le secret est compromis, l'administrateur peut le remplacer dans le coffre-fort du service, puis le redistribuer aux instances de service individuelles qui n'ont pas été compromises.

Conditions préalables

- Le composant du système de certificats de l'autorité de récupération des clés (KRA) a été installé sur un ou plusieurs serveurs de votre domaine IdM. Pour plus de détails, voir Installation de l'autorité de recouvrement des clés dans IdM.

Cette section comprend les procédures suivantes

1. Stockage d'un secret de service IdM dans un coffre-fort asymétrique
2. Récupération d'un secret de service pour une instance de service IdM
3. Modification du secret de la chambre forte d'un service IdM en cas de compromission

Terminologie utilisée

Dans les procédures :

- **admin** est l'administrateur qui gère le mot de passe du service.
- **private-key-to-an-externally-signed-certificate.pem** est le fichier contenant le secret du service, dans ce cas une clé privée d'un certificat signé en externe. Ne confondez pas cette clé privée avec la clé privée utilisée pour récupérer le secret dans le coffre-fort.
- **secret_vault** est le coffre-fort créé pour le service.
- **HTTP/webserver.idm.example.com** est le service dont le secret est archivé.
- **service-public.pem** est la clé publique du service utilisée pour crypter le mot de passe stocké dans **password_vault**.
- **service-private.pem** est la clé privée du service utilisée pour décrypter le mot de passe stocké dans **secret_vault**.

4.1. STOCKAGE D'UN SECRET DE SERVICE IDM DANS UN COFFRE-FORT ASYMÉTRIQUE

Cette section explique comment créer un coffre-fort asymétrique et l'utiliser pour archiver un secret de service.

Conditions préalables

- Vous connaissez le mot de passe de l'administrateur IdM.

Procédure

1. Connectez-vous en tant qu'administrateur :

```
$ kinit admin
```

2. Obtenir la clé publique de l'instance de service. Par exemple, en utilisant l'utilitaire **openssl**:

   a. Générer la clé privée **service-private.pem**.

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

   b. Générer la clé publique **service-public.pem** à partir de la clé privée.

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

3. Créez un coffre-fort asymétrique en tant que coffre-fort de l'instance de service et fournissez la clé publique :

```
$ 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
```

   Le mot de passe archivé dans la chambre forte sera protégé par la clé.

4. Archiver le secret de service dans le coffre-fort de service :

```
$ 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"
----------------------------------
```

   Cela permet de chiffrer le secret avec la clé publique de l'instance de service.

Répétez ces étapes pour chaque instance de service nécessitant le secret. Créez un nouveau coffre-fort asymétrique pour chaque instance de service.

4.2. RÉCUPÉRATION D'UN SECRET DE SERVICE POUR UNE INSTANCE DE SERVICE IDM

Cette section décrit comment une instance de service peut récupérer le secret du coffre-fort de service en utilisant une clé privée de service stockée localement.

Conditions préalables

- Vous avez accès au keytab du principal de service propriétaire de l'espace de stockage, par exemple HTTP/webserver.idm.example.com.
- Vous avez créé un coffre-fort asymétrique et archivé un secret dans le coffre-fort .
- Vous avez accès à la clé privée utilisée pour récupérer le secret du coffre-fort de service.

Procédure

1. Connectez-vous en tant qu'administrateur :

```
$ kinit admin
```

2. Obtenir un ticket Kerberos pour le service :

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

3. Récupérer le mot de passe du coffre-fort du service :

```
$ 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. MODIFICATION DU SECRET DE LA CHAMBRE FORTE D'UN SERVICE IDM EN CAS DE COMPROMISSION

Cette section décrit comment isoler une instance de service compromise en modifiant le secret de l'espace de stockage du service.

Conditions préalables

- Vous connaissez le mot de passe de **IdM administrator**.
- Vous avez créé un coffre-fort asymétrique pour stocker le secret de service.
- Vous avez généré le nouveau secret et y avez accès, par exemple dans le fichier **new-private-key-to-an-externally-signed-certificate.pem**.

Procédure

1. Archiver le nouveau secret dans le coffre-fort de l'instance de service :

```
$ 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"
----------------------------------
```

   Cette opération écrase le secret actuel stocké dans le coffre-fort.

2. Récupérer le nouveau secret sur les instances de service non compromises uniquement. Pour plus de détails, voir Récupération d'un secret de service pour une instance de service IdM .

4.4. RESSOURCES SUPPLÉMENTAIRES

- Voir Utiliser Ansible pour gérer les coffres-forts des services IdM : stocker et récupérer les secrets.

CHAPITRE 5. UTILISER ANSIBLE POUR GÉRER LES COFFRES-FORTS DES SERVICES IDM : STOCKER ET RÉCUPÉRER LES SECRETS

Cette section montre comment un administrateur peut utiliser le module **ansible-freeipa vault** pour stocker en toute sécurité un secret de service dans un emplacement centralisé. Le coffre-fort utilisé dans l'exemple est asymétrique, ce qui signifie que pour l'utiliser, l'administrateur doit effectuer les étapes suivantes :

1. Générez une clé privée en utilisant, par exemple, l'utilitaire **openssl**.
2. Générer une clé publique à partir de la clé privée.

Le secret de service est crypté avec la clé publique lorsqu'un administrateur l'archive dans le coffre-fort. Ensuite, une instance de service hébergée sur une machine spécifique du domaine récupère le secret à l'aide de la clé privée. Seuls le service et l'administrateur sont autorisés à accéder au secret.

Si le secret est compromis, l'administrateur peut le remplacer dans le coffre-fort du service, puis le redistribuer aux instances de service individuelles qui n'ont pas été compromises.

Conditions préalables

- Le composant du système de certificats de l'autorité de récupération des clés (KRA) a été installé sur un ou plusieurs serveurs de votre domaine IdM. Pour plus de détails, voir Installation de l'autorité de recouvrement des clés dans IdM.

Cette section comprend ces procédures :

- Assurer la présence d'un coffre-fort de service asymétrique dans IdM à l'aide d'Ansible
- Stocker un secret de service IdM dans un coffre-fort asymétrique à l'aide d'Ansible
- Récupérer un secret de service pour un service IdM en utilisant Ansible
- Changer le secret du coffre d'un service IdM en cas de compromission en utilisant Ansible

Dans les procédures :

- **admin** est l'administrateur qui gère le mot de passe du service.
- **private-key-to-an-externally-signed-certificate.pem** est le fichier contenant le secret du service, dans ce cas une clé privée d'un certificat signé en externe. Ne confondez pas cette clé privée avec la clé privée utilisée pour récupérer le secret dans le coffre-fort.
- **secret_vault** est le coffre-fort créé pour stocker le secret de service.
- **HTTP/webserver1.idm.example.com** est le service propriétaire de la chambre forte.
- **HTTP/webserver2.idm.example.com** et **HTTP/webserver3.idm.example.com** sont les services aux membres de la voûte.
- **service-public.pem** est la clé publique du service utilisée pour crypter le mot de passe stocké dans **password_vault**.
- **service-private.pem** est la clé privée du service utilisée pour décrypter le mot de passe stocké dans **secret_vault**.

5.1. ASSURER LA PRÉSENCE D'UN COFFRE-FORT DE SERVICE ASYMÉTRIQUE DANS IDM À L'AIDE D'ANSIBLE

Cette section montre comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour créer un conteneur de coffre-fort de service avec un ou plusieurs coffres-forts privés pour stocker en toute sécurité des informations sensibles. Dans l'exemple utilisé dans la procédure ci-dessous, l'administrateur crée un coffre-fort asymétrique nommé **secret_vault**. Cela garantit que les membres de l'espace de stockage doivent s'authentifier à l'aide d'une clé privée pour récupérer le secret dans l'espace de stockage. Les membres du coffre-fort pourront récupérer le fichier à partir de n'importe quel client IdM.

Conditions préalables

- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.
  - L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.
- Vous connaissez le mot de passe de **IdM administrator**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

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

2. Obtenir la clé publique de l'instance de service. Par exemple, en utilisant l'utilitaire **openssl**:
   a. Générer la clé privée **service-private.pem**.

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

   b. Générer la clé publique **service-public.pem** à partir de la clé privée.

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

3. Facultatif : Créez un fichier d'inventaire s'il n'existe pas, par exemple **inventory.file**:

```
$ touch inventory.file
```

4. Ouvrez votre fichier d'inventaire et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

5. Faites une copie du fichier **ensure-asymmetric-vault-is-present.yml** Ansible playbook. Par exemple :

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

6. Ouvrez le fichier **ensure-asymmetric-vault-is-present-copy.yml** pour le modifier.

7. Ajoutez une tâche qui copie la clé publique **service-public.pem** du contrôleur Ansible vers le serveur **server.idm.example.com**.

8. Modifiez le reste du fichier en définissant les variables suivantes dans la section **ipavault** task :

   - Fixer la variable **ipaadmin_password** au mot de passe de l'administrateur de l'IdM.
   - Définissez le nom de la chambre forte à l'aide de la variable **name**, par exemple **secret_vault**.
   - Fixer la variable **vault_type** à **asymmetric**.
   - Définissez la variable **service** comme étant le principal du service propriétaire de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.
   - Réglez l'adresse **public_key_file** sur l'emplacement de votre clé publique.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- 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. Enregistrer le fichier.

10. Exécutez le manuel de jeu :

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

5.2. AJOUTER DES SERVICES MEMBRES À UN COFFRE-FORT ASYMÉTRIQUE EN UTILISANT ANSIBLE

Cette section montre comment un administrateur Identity Management (IdM) peut utiliser un playbook Ansible pour ajouter des services membres à un coffre-fort de service afin qu'ils puissent tous récupérer le secret stocké dans le coffre-fort. Dans l'exemple utilisé dans la procédure ci-dessous, l'administrateur IdM ajoute les principaux services **HTTP/webserver2.idm.example.com** et **HTTP/webserver3.idm.example.com** au coffre-fort **secret_vault** qui appartient à **HTTP/webserver1.idm.example.com**.

Conditions préalables

- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.
  - L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.
- Vous connaissez le mot de passe de **IdM administrator**.
- Vous avez créé un coffre-fort asymétrique pour stocker le secret de service.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

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

2. Facultatif : Créez un fichier d'inventaire s'il n'existe pas, par exemple **inventory.file**:

```
$ touch inventory.file
```

3. Ouvrez votre fichier d'inventaire et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

4. Faites une copie du fichier **data-archive-in-asymmetric-vault.yml** Ansible playbook. Par exemple :

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

5. Ouvrez le fichier **data-archive-in-asymmetric-vault-copy.yml** pour le modifier.
6. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'administrateur de l'IdM.
   - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
   - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.
   - Définissez les services qui doivent avoir accès au secret de l'espace de stockage à l'aide de la variable **services**.
   - Fixer la variable **action** à **member**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- 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. Enregistrer le fichier.
8. Exécutez le manuel de jeu :

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

5.3. STOCKER UN SECRET DE SERVICE IDM DANS UN COFFRE-FORT ASYMÉTRIQUE À L'AIDE D'ANSIBLE

Cette section montre comment un administrateur de gestion des identités (IdM) peut utiliser un playbook Ansible pour stocker un secret dans un coffre-fort de service afin qu'il puisse être récupéré ultérieurement par le service. Dans l'exemple utilisé dans la procédure ci-dessous, l'administrateur stocke un fichier **PEM** avec le secret dans un coffre-fort asymétrique nommé **secret_vault**. Cela garantit que le service devra s'authentifier à l'aide d'une clé privée pour récupérer le secret dans le coffre-fort. Les membres du coffre-fort pourront récupérer le fichier à partir de n'importe quel client IdM.

Conditions préalables

- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.
  - L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.
- Vous connaissez le mot de passe de **IdM administrator**.
- Vous avez créé un coffre-fort asymétrique pour stocker le secret de service.
- Le secret est stocké localement sur le contrôleur Ansible, par exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

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

2. Facultatif : Créez un fichier d'inventaire s'il n'existe pas, par exemple **inventory.file**:

```
$ touch inventory.file
```

3. Ouvrez votre fichier d'inventaire et définissez le serveur IdM que vous souhaitez configurer dans la section **[ipaserver]**. Par exemple, pour demander à Ansible de configurer **server.idm.example.com**, entrez :

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

4. Faites une copie du fichier **data-archive-in-asymmetric-vault.yml** Ansible playbook. Par exemple :

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

5. Ouvrez le fichier **data-archive-in-asymmetric-vault-copy.yml** pour le modifier.
6. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :
   - Fixer la variable **ipaadmin_password** au mot de passe de l'administrateur de l'IdM.
   - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
   - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.

- Définissez la variable **in** à **"{{ lookup('file', 'private-key-to-an-externally-signed-certificate.pem') | b64encode }}"**, ce qui garantit qu'Ansible récupère le fichier contenant la clé privée dans le répertoire de travail du contrôleur Ansible plutôt que sur le serveur IdM.

- Fixer la variable **action** à **member**.
  Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- 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. Enregistrer le fichier.

8. Exécutez le manuel de jeu :

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

5.4. RÉCUPÉRER UN SECRET DE SERVICE POUR UN SERVICE IDM EN UTILISANT ANSIBLE

Cette section montre comment un utilisateur de la gestion des identités (IdM) peut utiliser un livre de jeu Ansible pour récupérer un secret dans un coffre-fort de service au nom du service. Dans l'exemple utilisé dans la procédure ci-dessous, l'exécution du livre de jeu récupère un fichier **PEM** avec le secret d'un coffre-fort asymétrique nommé **secret_vault**, et le stocke à l'emplacement spécifié sur tous les hôtes répertoriés dans le fichier d'inventaire Ansible sous le nom **ipaservers**.

Les services s'authentifient auprès de l'IdM à l'aide de keytabs, et ils s'authentifient auprès du coffre-fort à l'aide d'une clé privée. Vous pouvez récupérer le fichier au nom du service à partir de n'importe quel client IdM sur lequel **ansible-freeipa** est installé.

Conditions préalables

- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.

- L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.

- Vous connaissez le mot de passe de l'administrateur IdM.
- Vous avez créé un coffre-fort asymétrique pour stocker le secret de service.
- Vous avez archivé le secret dans la chambre forte .
- Vous avez stocké la clé privée utilisée pour récupérer le secret du coffre-fort du service dans l'emplacement spécifié par la variable **private_key_file** sur le contrôleur Ansible.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

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

2. Facultatif : Créez un fichier d'inventaire s'il n'existe pas, par exemple **inventory.file**:

```
$ touch inventory.file
```

3. Ouvrez votre fichier d'inventaire et définissez les hôtes suivants :
   - Définissez votre serveur IdM dans la section **[ipaserver]**.
   - Définissez les hôtes sur lesquels vous souhaitez récupérer le secret dans la section **[webservers]**. Par exemple, pour demander à Ansible de récupérer le secret sur **webserver1.idm.example.com**, **webserver2.idm.example.com**, et **webserver3.idm.example.com**, entrez :

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

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

4. Faites une copie du fichier **retrieve-data-asymmetric-vault.yml** Ansible playbook. Par exemple :

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

5. Ouvrez le fichier **retrieve-data-asymmetric-vault-copy.yml** pour le modifier.
6. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :
   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
   - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.

- Définissez la variable **private_key_file** à l'emplacement de la clé privée utilisée pour récupérer le secret du coffre-fort de service.

- Définissez la variable **out** à l'emplacement du serveur IdM où vous souhaitez récupérer le secret **private-key-to-an-externally-signed-certificate.pem**, par exemple le répertoire de travail actuel.

- Fixer la variable **action** à **member**.
  Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- 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. Ajouter une section au playbook qui récupère le fichier de données du serveur IdM vers le contrôleur Ansible :

```
---
- 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. Ajoutez une section au playbook qui transfère le fichier **private-key-to-an-externally-signed-certificate.pem** récupéré depuis le contrôleur Ansible vers les serveurs web répertoriés dans la section **webservers** du fichier d'inventaire :

```
---
- 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. Enregistrer le fichier.
10. Exécutez le manuel de jeu :

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

5.5. CHANGER LE SECRET DU COFFRE D'UN SERVICE IDM EN CAS DE COMPROMISSION EN UTILISANT ANSIBLE

Cette section montre comment un administrateur de gestion des identités (IdM) peut réutiliser un playbook Ansible pour modifier le secret stocké dans un coffre-fort de service lorsqu'une instance de service a été compromise. Le scénario de l'exemple suivant suppose que sur **webserver3.idm.example.com**, le secret récupéré a été compromis, mais pas la clé du coffre-fort asymétrique stockant le secret. Dans cet exemple, l'administrateur réutilise les playbooks Ansible utilisés pour stocker un secret dans un coffre-fort as ymétrique et pour récupérer un secret du coffre-fort asymétrique sur les hôtes IdM. Au début de la procédure, l'administrateur IdM stocke un nouveau fichier **PEM** avec un nouveau secret dans le coffre-fort asymétrique, adapte le fichier d'inventaire de manière à ne pas récupérer le nouveau secret sur le serveur web compromis, **webserver3.idm.example.com**, puis réexécute les deux procédures.

Conditions préalables

- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.
  - L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.
- Vous connaissez le mot de passe de **IdM administrator**.
- Vous avez créé un coffre-fort asymétrique pour stocker le secret de service.
- Vous avez généré une nouvelle clé **httpd** pour les services web fonctionnant sur les hôtes IdM afin de remplacer l'ancienne clé compromise.
- La nouvelle clé **httpd** est stockée localement sur le contrôleur Ansible, par exemple dans le fichier **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-signed-certificate.pem**.

Procédure

1. Naviguez jusqu'au répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault**:

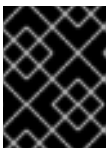
```
$ cd /usr/share/doc/ansible-freeipa/playbooks/vault
```

2. Ouvrez votre fichier d'inventaire et assurez-vous que les hôtes suivants sont définis correctement :

   - Le serveur IdM dans la section **[ipaserver]**.
   - Les hôtes sur lesquels vous souhaitez récupérer le secret dans la section **[webservers]**. Par exemple, pour demander à Ansible de récupérer le secret sur **webserver1.idm.example.com** et **webserver2.idm.example.com**, entrez :

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

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



IMPORTANT

Assurez-vous que la liste ne contient pas le serveur web compromis, dans l'exemple actuel **webserver3.idm.example.com**.

3. Ouvrez le fichier **data-archive-in-asymmetric-vault-copy.yml** pour le modifier.
4. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :

   - Fixer la variable **ipaadmin_password** au mot de passe de l'administrateur de l'IdM.
   - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
   - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver.idm.example.com**.
   - Définissez la variable **in** à **"{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem') | b64encode }}"**, ce qui garantit qu'Ansible récupère le fichier contenant la clé privée dans le répertoire de travail du contrôleur Ansible plutôt que sur le serveur IdM.
   - Fixer la variable **action** à **member**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- 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. Enregistrer le fichier.

6. Exécutez le manuel de jeu :

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

7. Ouvrez le fichier **retrieve-data-asymmetric-vault-copy.yml** pour le modifier.

8. Modifiez le fichier en définissant les variables suivantes dans la section **ipavault** task :

   - Définissez la variable **ipaadmin_password** avec votre mot de passe d'administrateur IdM.
   - Attribuez à la variable **name** le nom de la chambre forte, par exemple **secret_vault**.
   - Définissez la variable **service** comme étant le propriétaire du service de la chambre forte, par exemple **HTTP/webserver1.idm.example.com**.
   - Définissez la variable **private_key_file** à l'emplacement de la clé privée utilisée pour récupérer le secret du coffre-fort de service.
   - Définissez la variable **out** à l'emplacement du serveur IdM où vous souhaitez récupérer le secret **new-private-key-to-an-externally-signed-certificate.pem**, par exemple le répertoire de travail actuel.
   - Fixer la variable **action** à **member**.
     Il s'agit du fichier playbook Ansible modifié pour l'exemple actuel :

```
---
- 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. Ajouter une section au playbook qui récupère le fichier de données du serveur IdM vers le contrôleur Ansible :

```
---
```

```
- name: Retrieve data from vault
  hosts: ipaserver
  become: yes
  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. Ajoutez une section au playbook qui transfère le fichier **new-private-key-to-an-externally-signed-certificate.pem** récupéré depuis le contrôleur Ansible vers les serveurs web répertoriés dans la section **webservers** du fichier d'inventaire :

```
---
- name: Send data file to webservers
  become: yes
  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. Enregistrer le fichier.

12. Exécutez le manuel de jeu :

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

5.6. RESSOURCES SUPPLÉMENTAIRES

- Voir le fichier README-vault.md Markdown dans le répertoire **/usr/share/doc/ansible-freeipa/**.
- Voir les exemples de playbooks dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/vault/**.