Red Hat Enterprise Linux 9

在身份管理中使用库（vault）

在 IdM 中存储和管理敏感数据

Last Updated: 2024-09-29

Red Hat Enterprise Linux 9 在身份管理中使用库（vault）

在 IdM 中存储和管理敏感数据

法律通告

摘要

vault 是红帽身份管理(IdM)中的安全位置，用于存储、检索和共享敏感数据，如服务的身份验证凭据。您可以使用命令行或 Ansible Playbook 管理 vault。

目录

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 Jira 提交反馈（需要帐户）

1. 登录到 Jira 网站。
2. 在顶部导航栏中点 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 IDM 中的 VAULTS

本章论述了 Identity Management(IdM)中的库。它介绍了以下主题：

- 库的概念.
- 与 vault 关联的不同角色。
- IdM 中根据安全性和访问控制有不同类型的 vault。
- 基于所有权的 IdM 中可用的不同类型的 vault。
- vault 容器的概念。
- 在 IdM 中管理 vault 的基本命令 。
- 安装密钥恢复机构(KRA)，这是在 IdM 中使用 vaults 的先决条件 。

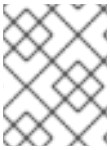
1.1. 库及其优点

对于希望在一个位置保持所有敏感数据存储的 Identity Management(IdM)用户，库是一个有用的功能。 有各种类型的 vault，您应该根据您的要求选择要使用的 vault。

vault 是(IdM)中的安全位置，用于存储、检索、共享和恢复 secret。secret 是安全敏感的数据，通常是身份验证凭据，仅有限的人员或实体可以访问。例如，secret 包括：

- 密码
- PINs
- 私有 SSH 密钥

vault 与密码管理器相当。与密码管理器类似，vault 通常要求用户生成并记住一个主密码，以解锁和访问密码库中存储的任何信息。但是，用户也可以决定使用标准 vault。标准密码库不要求用户输入任何密码来访问密码库中存储的 secret。

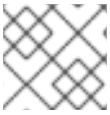


注意

IdM 中的 vaults 的目的是存储身份验证凭证，可让您向外部、非 IdM 相关的服务进行身份验证。

IdM 库的其他重要特性包括：

- vaults 只能供 vault 所有者以及 vault 所有者选择为 vault 成员的用户访问。另外，IdM 管理员也可以访问 vault。
- 如果用户没有足够的权限来创建 vault，IdM 管理员可以创建 vault 并将用户设置为其所有者。
- 用户和服务可以从 IdM 域中注册的任何机器访问存储在 vault 中的 secret。
- 一个 vault 只能包含一个 secret，例如一个文件。但是，文件本身可以包含多个 secret，如密码、keytabs 或证书。



注意

Vault 仅适用于 IdM 命令行(CLI)，而不能从 IdM Web UI 使用。

1.2. VAULT 所有者、成员和管理员

Identity Management(IdM)可区分以下 vault 用户类型：

Vault 所有者

vault 所有者是具有密码库上基本管理特权的用户或服务。例如，vault 所有者可以修改 vault 的属性或添加新的 vault 成员。
每个 vault 必须至少有一个所有者。库也可以有多个所有者。

Vault 成员

vault 成员是用户访问由另一个用户或服务创建的库的用户或服务。

Vault 管理员

Vault 管理员对所有 vaults 具有不受限制的访问权限，并且可以执行所有 vault 操作。

注意

对称和非对称的密码库使用密码或密钥进行保护，并应用特殊的访问控制规则（请参阅 Vault 类型）。管理员必须满足以下条件：

- 访问对称和非对称库中的 secret。
- 更改或重置 vault 密码或密钥。

vault 管理员是具有 **Vault Administrators** 权限。在 IdM 中基于角色的访问控制(RBAC)的上下文中，权限是一个可应用于角色的权限组。

Vault 用户

vault 用户代表密码库所在的用户。**Vault 用户** 信息显示在特定命令的输出中，如 **ipa vault-show** ：

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

有关 vault 容器和用户 vault 的详情，请参阅 Vault 容器。

其他资源

- 如需有关 vault 类型的详情，请参阅 标准的、对称的和非对称的vault 。

1.3. 标准、对称和非对称库

根据安全性和访问控制级别，IdM 将 vaults 统一为以下类型：

标准库

Vault 所有者和 vault 成员可以存档和检索机密，而无需使用密码或密钥。

对称库

密码库中的 secret 使用对称密钥进行保护。Vault 所有者和成员可以存档并检索机密，但它们必须提供 vault 密码。

非对称库

密码库中的 secret 使用非对称密钥进行保护。用户使用公钥归档密码，并使用私钥检索该密码。Vault 成员只能存档机密，而 vault 所有者可以执行、存档和检索机密。

1.4. 用户、服务和共享库

根据所有权，IdM 将 vaults 分为几种类型。下表包含有关每种类型、其所有者和使用的信息。

表 1.1. 基于所有权的 IdM 库

类型	描述	所有者	备注
User vault	用户的专用库	单个用户	如果 IdM 管理员允许，任何用户都可以拥有一个或多个用户 vault
Service vault	服务的专用库	单个服务	如果 IdM 管理员允许，任何服务都可以拥有一个或多个用户 vault
共享库	由多个用户和服务共享的库	创建 vault 的 vault 管理员	如果 IdM 管理员允许，用户和服务可以拥有一个或多个用户 vault。除创建密码库以外的 vault 管理员也具有对密码库的完全访问权限。

1.5. VAULT 容器

vault 容器是密码库的集合。下表列出了 Identity Management (IdM) 提供的默认 vault 容器。

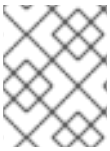
表 1.2. IdM 中的默认 vault 容器

类型	描述	目的
用户容器	用户的私有容器	为特定用户存储用户密码库
服务容器	服务的私有容器	为特定服务存储服务库
共享容器	用于多个用户和服务的容器	存储可由多个用户或服务共享的 vault

当为用户或服务创建第一个私有密码库时，IdM 会自动为每个用户或服务创建用户和服务容器。删除用户或服务后，IdM 会删除容器及其内容。

1.6. 基本 IDM VAULT 命令

您可以使用以下介绍的基本命令管理身份管理(IdM) vault。下表 包含 **ipa vault-*** 命令的列表，并解释了它们的用途。



注意

在运行任何 **ipa vault-*** 命令前，请将密钥恢复授权 (KRA) 证书系统组件安装到 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

表 1.3. 基本 IdM vault 命令解释

命令	目的
ipa help vault	显示有关 IdM 库和示例密码库命令的概念信息。
ipa vault-add --help,**ipa vault-find --help**	在特定的 **ipa vault-*** 命令中添加 **--help** 选项会显示该命令可用的选项和详细帮助。

ipa vault-show user_vault --user idm_user	在将密码库作为 vault 成员访问时，您必须指定 vault 所有者。如果您没有指定 vault 所有者，IdM 会通知您没有找到密码库：
	`[admin@server ~]$ ipa vault-show user_vault`
	`ipa: ERROR: user_vault: vault not found`

ipa vault-show shared_vault --shared	在访问共享密码库时，您必须指定您要访问的 vault 是共享密码库。否则，IdM 会通知您没有找到密码库：
	`[admin@server ~]$ ipa vault-show shared_vault`
	`ipa: ERROR: shared_vault: vault not found`

1.7. 在 IDM 中安装密钥恢复授权

按照以下流程，通过在特定的 IdM 服务器上安装密钥恢复授权(KRA)证书系统(CS)组件来在身份管理(IdM)中启用 vault。

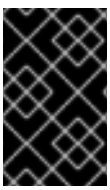
先决条件

- 您已以 **root** 身份登录到 IdM 服务器。
- IdM 证书颁发机构已安装在 IdM 服务器上。
- 您有 **目录管理器** 凭证。

步骤

- 安装 KRA：

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



重要

您可以在隐藏的副本上安装 IdM 集群的第一个 KRA。但是，在在非隐藏的副本上安装 KRA 克隆前，安装额外的 KRA 克隆需要临时激活隐藏的副本。然后您可以再次隐藏原始隐藏的副本。

注意

要使密码库服务高可用且具有弹性，请在两个或多个 IdM 服务器上安装 KRA。维护多个 KRA 服务器可防止数据丢失。

其他资源

- 演示或提升隐藏副本
- 隐藏的副本模式

第 2 章 使用 IDM 用户 VAULT：存储和检索 SECRET

本章论述了如何在身份管理中使用用户库。具体来说，它描述了用户如何将 secret 存储在 IdM vault 中，以及用户如何检索 secret。用户可以通过两个不同的 IdM 客户端进行存储和检索。

先决条件

- 密钥恢复授权中心 (KRA) 证书系统组件已安装在您的 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

2.1. 将 SECRET 存储在用户 VAULT 中

按照以下流程创建带有一个或多个私有 vault 的 vault 容器，以安全地存储具有敏感信息的文件。在以下步骤中使用的示例中，**idm_user** 用户创建标准类型库。标准密码库类型确保无需 **idm_user** 在访问该文件时进行身份验证。**idm_user** 能够从用户登录的任何 IdM 客户端检索 文件。

在此过程中：

- **idm_user** 是要创建 vault 的用户。
- **my_vault** 是用于存储用户证书的 vault。
- vault 类型是 **standard**，因此访问存档证书不需要用户提供 vault 密码。
- **secret.txt** 是包含用户要在密码库中存储的证书的文件。

先决条件

- 您知道 **idm_user** 的密码。
- 您登录到一个 IdM 客户端的主机。

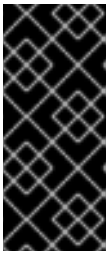
步骤

1. 获取 **idm_user** 的 Kerberos ticket granting ticket (TGT) ：

```
$ kinit idm_user
```

2. 使用 **ipa vault-add** 命令和 **--type standard** 选项来创建标准 vault：

```
$ 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

确保一个用户的第一个 vault 由相同的用户创建。为用户创建第一个 vault 也会创建用户的 vault 容器。创建的代理变成 vault 容器的所有者。

例如，如果另一个用户（如 **admin**）为 **user1** 创建了第一个用户 vault，该用户的 vault 容器的所有者会是 **admin**，**user1** 无法访问用户 vault 或创建新用户 vault。

3. 使用带有 **--in** 选项的 **ipa vault-archive** 命令，将 **secret.txt** 文件归档到 vault 中：

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

2.2. 从用户 VAULT 检索 SECRET

作为 Identity Management(IdM)，您可以从登陆到的任何 IdM 客户端中的用户私有 vault 中检索 secret。

按照以下流程，以名为 **idm_user** 的 IdM 用户身份，将名为 **my_vault** 的用户私有 vault 中的 secret 检索到 **idm_client.idm.example.com**。

先决条件

- **idm_user** 是 **my_vault** 的所有者。
- **idm_user** 在 vault 中存档了一个 secret。
- **my_vault** 是标准 vault，这意味着 **idm_user** 不必输入任何密码就可以访问 vault 的内容。

步骤

1. 以 **idm_user** 用户身份通过 SSH 连接到 **idm_client** ：

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

2. 以 **idm_user** 身份登录：

```
$ kinit user
```

3. 使用带有 **--out** 选项的 **ipa vault-retrieve --out** 命令来检索密码库的内容并将其保存到 **secret_exported.txt** 文件中。

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

2.3. 其他资源

- 请参阅 使用 Ansible 来管理 IdM 服务库：存储和检索 secret 。

第 3 章 使用 ANSIBLE 管理 IDM 用户库：存储和检索 SECRET

本章论述了如何使用 Ansible **vault** 模块在身份管理中管理用户密码库。具体来说，它描述了用户可以如何使用 Ansible playbook 执行以下三个连续操作：

- 在 IdM 中创建用户 vault。
- 在密码库中存储机密。
- 从密码库检索机密。

用户可以通过两个不同的 IdM 客户端进行存储和检索。

先决条件

- 密钥恢复授权中心 (KRA) 证书系统组件已安装在您的 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

3.1. 使用 ANSIBLE 在 IDM 中存在标准用户库

按照以下流程，使用 Ansible playbook 创建包含一个或多个私有 vault 的 vault 容器，以安全地存储敏感信息。在以下步骤中使用的示例中，**idm_user** 用户创建名为 **my_vault** 的标准类型库。标准密码库类型确保无需 **idm_user** 在访问该文件时进行身份验证。**idm_user** 能够从用户登录的任何 IdM 客户端检索 文件。

先决条件

- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包，这是您在该流程中执行步骤的主机。
- 您知道 **idm_user** 的密码。

步骤

1. 导航到 **/usr/share/doc/ansible-freeipa/playbooks/vault** 目录：

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

2. 创建一个清单文件，如 **inventory.file** ：

```
$ touch inventory.file
```

3. 打开 **inventory.file**，并在 **[ipaserver]** 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

4. 生成 **ensure-standard-vault-is-present.yml** Ansible playbook 文件的副本。例如：

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

5. 打开 **ensure-standard-vault-is-present-copy.yml** 文件进行编辑。

6. 通过在 **ipavault** 任务部分设置以下变量来调整文件：
   - 将 **ipaadmin_principal** 变量设置为 **idm_user**。
   - 将 **ipaadmin_password** 变量设置为 **idm_user** 密码。
   - 将 **user** 变量设置为 **idm_user**。
   - 将 **name** 变量设置为 **my_vault**。
   - 将 **vault_type** 变量设置为 **standard**。
     对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
8. 运行 playbook：

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

3.2. 使用 ANSIBLE 将 SECRET 归档到 IDM 中的标准用户库中

按照以下流程，使用 Ansible playbook 将敏感信息存储在个人 vault 中。在使用的示例中，**idm_user** 用户在名为 **my_vault** 的库中归档含有名为 **password.txt** 的敏感信息的文件。

先决条件

- 您已在 Ansible 控制器上安装了 ansible-freeipa 软件包，这是您在该流程中执行步骤的主机。
- 您知道 **idm_user** 的密码。
- **idm_user** 是所有者，或者至少是 **my_vault** 的成员用户。
- 您可以访问 **password.txt**，这是要在 **my_vault** 中存档的机密。

步骤

1. 导航到 **/usr/share/doc/ansible-freeipa/playbooks/vault** 目录：

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

2. 打开清单文件，并确保 **[ipaserver]** 部分中列出了您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

3. 制作 **data-archive-in-symmetric-vault.yml** Ansible playbook 文件的副本，但将"symmetric"替换为 "standard"。例如：

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

4. 打开 **data-archive-in-standard-vault-copy.yml** 文件进行编辑。

5. 通过在 **ipavault** 任务部分设置以下变量来调整文件：

- 将 **ipaadmin_principal** 变量设置为 **idm_user**。
- 将 **ipaadmin_password** 变量设置为 **idm_user** 密码。
- 将 **user** 变量设置为 **idm_user**。
- 将 **name** 变量设置为 **my_vault**。
- 将 **in** 变量设置为包含敏感信息的文件的完整路径。
- 将 **action** 变量设置为 **member**。
  对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。

7. 运行 playbook：

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

3.3. 使用 ANSIBLE 从 IDM 中的标准用户库检索 SECRET

按照以下流程，使用 Ansible playbook 从用户个人 vault 检索 secret。在以下步骤中使用的示例中，**idm_user** 用户从名为 **my_vault** 的标准类型库检索包含敏感数据的文件，并检索名为 **host01** 的 IdM 客户端。**idm_user** 在访问该文件时不必进行身份验证。**idm_user** 可以使用 Ansible 从安装 Ansible 的任何 IdM 客户端检索 文件。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible vault 存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 **idm_user** 的密码。
- **idm_user** 是 **my_vault** 的所有者。
- **idm_user** 已将 secret 存储在 **my_vault** 中。
- Ansible 可以写入要检索该 secret 的 IdM 主机上的 目录。
- **idm_user** 可以从要检索 secret 的 IdM 主机上的 目录读取。

步骤

1. 导航到 **/usr/share/doc/ansible-freeipa/playbooks/vault** 目录：

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

2. 打开清单文件，并在一个明确定义的部分中提到您要检索该 secret 的 IdM 客户端。例如，要指示 Ansible 在 **host01.idm.example.com** 上检索 secret，请输入：

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

3. 生成 **retrive-data-symmetric-vault.yml** Ansible playbook 文件的副本。将 "symmetric" 替换为 "standard"。例如：

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

4. 打开 **retrieve-data-standard-vault.yml-copy.yml** 文件进行编辑。
5. 通过将 **hosts** 变量设置为 **ipahost** 来调整 文件。
6. 通过在 **ipavault** 任务部分设置以下变量来调整文件：
   - 将 **ipaadmin_principal** 变量设置为 **idm_user**。

- 将 **ipaadmin_password** 变量设置为 **idm_user** 密码。
- 将 **user** 变量设置为 **idm_user**。
- 将 **name** 变量设置为 **my_vault**。
- 将 **out** 变量设置为您要将 secret 导出到的文件的完整路径。
- 将 **state** 变量设置为 **retrieve**。
  对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
8. 运行 playbook：

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

验证

1. 以 **user01** 身份通过 **SSH** 连接到 **host01**：

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

2. 查看 Ansible playbook 文件中 **out** 变量指定的文件：

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

现在，您可以看到导出的 secret。

- 有关使用 Ansible 管理 IdM vaults 和用户 secret 以及 playbook 变量的更多信息，请参阅 **/usr/share/doc/ansible-freeipa/** 目录中的 README-vault.md Markdown 文件，和 **/usr/share/doc/ansible-freeipa/playbooks/vault/** 目录中的示例 playbook。

第 4 章 管理 IDM 服务 SECRET：存储和检索 SECRET

本节演示了管理员如何在身份管理(IdM)中使用服务 vault，来在集中的地方安全地存储服务 secret 。示例中使用的 vault 是非对称的，这意味着要使用它，管理员需要执行以下步骤：

1. 使用 **openssl** 实用程序生成私钥。
2. 根据私钥生成公钥。

当管理员将服务 secret 归档到密码库时，会用公钥对其进行加密。之后，托管在域中特定计算机上的服务实例使用私钥检索该 secret。只有服务和管理员可以访问该 secret。

如果该机密泄露，管理员可以在服务 vault 中替换它，然后将它重新分发到尚未遭入侵的服务实例。

先决条件

- 密钥恢复授权中心 (KRA) 证书系统组件已安装在您的 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

本节包括这些步骤

1. 将 IdM 服务 secret 存储在非对称库中
2. 检索 IdM 服务实例的服务 secret
3. 在被破坏时更改 IdM 服务 vault secret

使用的术语

在流程中：

- **admin** 是管理服务密码的管理员。
- **private-key-to-an-externally-certificate.pem** 是包含服务 secret 的文件，本例中为外部签名证书的私钥。请勿将此私钥与用于从密码库检索机密的私钥混淆。
- **secret_vault** 是为服务创建的 vault。
- **HTTP/webserver.idm.example.com** 是存档其 secret 的服务。
- **service-public.pem** 是用于加密 **password_vault** 中存储的密码的服务公钥。
- **service-private.pem** 是用于解密 **secret_vault** 中存储的密码的服务私钥。

4.1. 将 IDM 服务 SECRET 存储在非对称库中

按照以下流程创建非对称 vault，并使用它来归档服务 secret。

先决条件

- 您知道 IdM 管理员密码。

步骤

1. 以管理员身份登录：

```
$ kinit admin
```

2. 获取服务实例的公钥。例如，使用 **openssl** 工具：

   a. 生成 **service-private.pem** 私钥。

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

   b. 根据私钥生成 **service-public.pem** 公钥。

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

3. 创建一个非对称库作为服务实例 vault，并提供公钥：

   ```
   $ 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
   ```

   归档到密码库的密码将使用密钥进行保护。

4. 将服务 secret 归档到 service vault 中：

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

   这会使用服务实例公钥加密该 secret。

对需要 secret 的每个服务实例重复这些步骤。为每个服务实例创建一个新的非对称库。

4.2. 检索 IDM 服务实例的服务 SECRET

按照以下流程，使用本地存储的服务私钥，使用服务实例检索服务 vault secret。

先决条件

- 您可以访问拥有服务 vault 的服务主体的 keytab，如 HTTP/webserver.idm.example.com。
- 您已创建了非对称 vault，并在 vault 中归档一个 secret 。

- 您可以访问用于检索服务 vault secret 的私钥。

步骤

1. 以管理员身份登录：

```
$ kinit admin
```

2. 为该服务获取 Kerberos ticket：

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

3. 检索服务 vault 密码：

```
$ 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. 在被破坏时更改 IDM 服务 VAULT SECRET

按照以下流程，通过更改服务 vault secret 来隔离被入侵的服务实例。

先决条件

- 您知道 IdM 管理员密码。
- 您已创建了非对称密码库用于存储服务机密。
- 您已生成新的 secret 并有权访问它，如在 **new-private-key-to-an-externally-signed-certificate.pem** 文件中。

步骤

1. 将新 secret 归档到 service instance vault 中：

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

这会覆盖存储在 vault 中的当前 secret。

2. 仅在非正式服务实例上检索新机密。详情请参阅 为 IdM 服务实例查找服务 secret。

4.4. 其他资源

- 请参阅 使用 Ansible 来管理 IdM 服务库：存储和检索 secret 。

第 5 章 使用 ANSIBLE 管理 IDM 服务库：存储和检索 SECRET

本节介绍管理员可以如何使用 **ansible-freeipa vault** 模块安全地将服务 secret 存储在集中式位置。示例中使用的 vault 是非对称的，这意味着要使用它，管理员需要执行以下步骤：

1. 使用 **openssl** 实用程序生成私钥。
2. 根据私钥生成公钥。

当管理员将服务 secret 归档到密码库时，会用公钥对其进行加密。之后，托管在域中特定计算机上的服务实例使用私钥检索该 secret。只有服务和管理员可以访问该 secret。

如果该机密泄露，管理员可以在服务 vault 中替换它，然后将它重新分发到尚未遭入侵的服务实例。

先决条件

- 密钥恢复授权中心 (KRA) 证书系统组件已安装在您的 IdM 域中的一个或多个服务器上。详情请参阅在 IdM 中安装密钥恢复授权。

本节包括以下步骤：

- 使用 Ansible 在 IdM 中存在非对称服务库
- 使用 Ansible 将 IdM 服务 secret 存储在非对称库中
- 使用 Ansible 为 IdM 服务检索服务 secret
- 在使用 Ansible 泄露时更改 IdM 服务 vault secret

在流程中：

- **admin** 是管理服务密码的管理员。
- **private-key-to-an-externally-certificate.pem** 是包含服务 secret 的文件，本例中为外部签名证书的私钥。请勿将此私钥与用于从密码库检索机密的私钥混淆。
- **secret_vault** 是为存储服务 secret 而创建的库。
- **HTTP/webserver1.idm.example.com** 是密码库的所有者服务。
- **HTTP/webserver2.idm.example.com** 和 **HTTP/webserver3.idm.example.com** 是 vault 成员服务。
- **service-public.pem** 是用于加密 **password_vault** 中存储的密码的服务公钥。
- **service-private.pem** 是用于解密 **secret_vault** 中存储的密码的服务私钥。

5.1. 使用 ANSIBLE 在 IDM 中存在非对称服务库

按照以下流程，使用 Ansible playbook 创建包含一个或多个私有 vault 容器的服务 vault 容器，以安全地存储敏感信息。在以下流程中使用的示例中，管理员创建名为 **secret_vault** 的非对称库。这样可确保 vault 成员必须使用私钥进行身份验证，以检索 vault 中的 secret。vault 成员能够从任何 IdM 客户端检索文件。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible vault 存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 **IdM 管理**员密码。

步骤

1. 导航到 **/usr/share/doc/ansible-freeipa/playbooks/vault** 目录：

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

2. 获取服务实例的公钥。例如，使用 **openssl** 工具：
   a. 生成 **service-private.pem** 私钥。

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

   b. 根据私钥生成 **service-public.pem** 公钥。

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

3. 可选：如果不存在，创建一个清单文件（如 **inventory.file**）。

```
$ touch inventory.file
```

4. 打开清单文件，并在 **[ipaserver]** 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

5. 生成 **ensure-asymmetric-vault-is-present.yml** Ansible playbook 文件的副本。例如：

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

6. 打开 **ensure-asymmetric-vault-is-present-copy.yml** 文件进行编辑。

7. 添加一个任务，该任务将 **service-public.pem** 公钥从 Ansible 控制器复制到 **server.idm.example.com** 服务器。
8. 通过在 **ipavault** 任务部分设置以下变量来修改文件的其余部分：
   - 将 **ipaadmin_password** 变量设置为 IdM 管理员密码。
   - 使用 **name** 变量定义 vault 的名称，如 **secret_vault**。
   - 将 **vault_type** 变量设置为非 **对称**。
   - 将 **service** 变量设置为拥有密码库的服务主体，如 **HTTP/webserver1.idm.example.com**。
   - 将 **public_key_file** 设置为您的公钥的位置。
     这是当前示例修改的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。
10. 运行 playbook：

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

5.2. 使用 ANSIBLE 将成员服务添加到非对称库

按照以下流程，使用 Ansible playbook 将成员服务添加到服务 vault 中，以便它们都可以检索 vault 中存储的 secret。在以下流程中使用的示例中，IdM 管理员将 **HTTP/webserver2.idm.example.com** 和 **HTTP/webserver3.idm.example.com** 服务主体添加到由 **HTTP/webserver1.idm.example.com** 所有的 **secret_vault** vault 中。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。

- 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
- 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
- 示例假定 **secret.yml** Ansible vault 存储了 **ipaadmin_password**。

- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 **IdM 管理员**密码。
- 您已创建了非对称密码库用于存储服务机密。

步骤

1. 导航到 **/usr/share/doc/ansible-freeipa/playbooks/vault** 目录：

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

2. 可选：如果不存在，创建一个清单文件（如 **inventory.file**）。

```
$ touch inventory.file
```

3. 打开清单文件，并在 **[ipaserver]** 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

4. 生成 **data-archive-in-asymmetric-vault.yml** Ansible playbook 文件的副本。例如：

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

5. 打开 **data-archive-in-asymmetric-vault-copy.yml** 文件进行编辑。
6. 通过在 **ipavault** 任务部分设置以下变量来修改该文件：
   - 将 **ipaadmin_password** 变量设置为 IdM 管理员密码。
   - 将 **name** 变量设置为 vault 的名称，如 **secret_vault**。
   - 将 **service** 变量设置为密码库的服务所有者，如 **HTTP/webserver1.idm.example.com**。
   - 定义您要使用 **services** 变量访问 vault 机密的服务。
   - 将 **action** 变量设置为 **member**。
     对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。

8. 运行 playbook：

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

5.3. 使用 ANSIBLE 将 IDM 服务 SECRET 存储在非对称库中

按照以下流程，使用 Ansible playbook 将 secret 存储在服务 vault 中，以便稍后可被服务检索。在以下流程中使用的示例中，管理员将带有 secret 的 **PEM** 文件存储在名为 **secret_vault** 的非对称库中。这样可确保服务必须使用私钥进行身份验证，以便从 vault 检索 secret。vault 成员能够从任何 IdM 客户端检索 文件。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 *~/MyPlaybooks/* 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible vault 存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 **IdM 管理**员密码。
- 您已创建了非对称密码库用于存储服务机密。
- secret 存储在 Ansible 控制器上，例如 **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-certificate.pem** 文件中。

步骤

1. 导航到 **/usr/share/doc/ansible-freeipa/playbooks/vault** 目录：

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

2. 可选：如果不存在，创建一个清单文件（如 **inventory.file**）。

```
$ touch inventory.file
```

3. 打开清单文件，并在 **[ipaserver]** 部分中定义您要配置的 IdM 服务器。例如，要指示 Ansible 配置 **server.idm.example.com**，请输入：

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

4. 生成 **data-archive-in-asymmetric-vault.yml** Ansible playbook 文件的副本。例如：

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

5. 打开 **data-archive-in-asymmetric-vault-copy.yml** 文件进行编辑。

6. 通过在 **ipavault** 任务部分设置以下变量来修改该文件：

   - 将 **ipaadmin_password** 变量设置为 IdM 管理员密码。
   - 将 **name** 变量设置为 vault 的名称，如 **secret_vault**。
   - 将 **service** 变量设置为密码库的服务所有者，如 **HTTP/webserver1.idm.example.com**。
   - 将 **in** 变量设置为 **"{{ lookup('file', 'private-key-to-an-externally-certificate.pem')| b64encode }}"**。这可确保 Ansible 使用私钥从 Ansible 控制器上的工作目录检索文件，而不是从 IdM 服务器检索。
   - 将 **action** 变量设置为 **member**。
     对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。

8. 运行 playbook：

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

5.4. 使用 ANSIBLE 为 IDM 服务检索服务 SECRET

按照以下流程，使用 Ansible playbook 代表服务从服务 vault 中检索 secret。在以下流程中使用的示例中，运行 playbook 从名为 **secret_vault** 的非对称库检索带有 secret 的 **PEM** 文件，并将它存储在 Ansible 清单文件中列出的所有主机上的指定位置，存为 **ipaservers**。

服务使用 keytabs 验证 IdM，并使用私钥与密码库进行身份验证。您可以代表服务从安装 **ansible-freeipa** 的任何 IdM 客户端检索 文件。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible vault 存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 IdM 管理员密码。
- 您已创建了非对称密码库用于存储服务机密。
- 您已在密码库中存档了机密。
- 您已将用于检索服务 vault secret 的私钥存储在 Ansible 控制器上的 **private_key_file** 变量指定的位置。

步骤

1. 导航到 **/usr/share/doc/ansible-freeipa/playbooks/vault** 目录：

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

2. 可选：如果不存在，创建一个清单文件（如 **inventory.file**）。

```
$ touch inventory.file
```

3. 打开清单文件并定义以下主机：
   - 在 **[ipaserver]** 部分中定义您的 IdM 服务器。
   - 在 **[webservers]** 部分中定义要检索机密的主机。例如，要指示 Ansible 获取到 **webserver1.idm.example.com**、**webserver2.idm.example.com** 和 **webserver3.idm.example.com** 的 secret，请输入：

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

[webservers]
```

```
webserver1.idm.example.com
webserver2.idm.example.com
webserver3.idm.example.com
```

4. 生成 **retrieve-data-asymmetric-vault.yml** Ansible playbook 文件的副本。例如：

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

5. 打开 **retrieve-data-asymmetric-vault-copy.yml** 文件进行编辑。

6. 通过在 **ipavault** 任务部分设置以下变量来修改该文件：

   - 将 **ipaadmin_password** 变量设置为 IdM 管理员密码。
   - 将 **name** 变量设置为 vault 的名称，如 **secret_vault**。
   - 将 **service** 变量设置为密码库的服务所有者，如 **HTTP/webserver1.idm.example.com**。
   - 将 **private_key_file** 变量设置为用于检索服务 vault secret 的私钥的位置。
   - 将 **out** 变量设置为 IdM 服务器上您要检索 **private-key-to-an-externally-certificate.pem** 机密的位置，如当前工作目录。
   - 将 **action** 变量设置为 **member**。
     对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 在 playbook 中添加一个部分，它将从 IdM 服务器检索数据文件到 Ansible 控制器：

```
---
- 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: true
      mode: 0600
```

8. 在 playbook 中添加一个部分，将检索到的 **private-key-to-an-externally-signed-certificate.pem** 文件从 Ansible 控制器所在的地方传输到清单文件的 **webservers** 部分所列出的 webserver 中：

```
---
- 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. 保存这个文件。

10. 运行 playbook：

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

5.5. 在使用 ANSIBLE 泄露时更改 IDM 服务 VAULT SECRET

当服务实例被破坏时，请按照此流程重复使用 Ansible playbook 来更改存储在服务 vault 中的 secret。以下示例中，假设获取的机密在 **webserver3.idm.example.com** 上已被破坏，而存储机密的非对称 vault 存储的密钥没有被破坏。在示例中，管理员重复利用在非对称库中存储一个 secret 时，以及从非对称库中获取一个 secret 导入到 IdM 主机 时使用的 Ansible playbook。在流程开始时，IdM 管理员将新的 **PEM** 文件存储在非对称的密码库中，对清单文件进行调整，以便不会从已被侵入的 Web 服务器（**webserver3.idm.example.com**）检索新机密，，然后重新运行这两个过程。

先决条件

- 您已配置了 Ansible 控制节点以满足以下要求：
  - 您使用 Ansible 版本 2.14 或更高版本。
  - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
  - 示例假定在 ***~/MyPlaybooks/*** 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 Ansible 清单文件。
  - 示例假定 **secret.yml** Ansible vault 存储了 **ipaadmin_password**。
- 目标节点（这是执行 **ansible-freeipa** 模块的节点）是 IdM 域的一部分，来作为 IdM 客户端、服务器或副本。
- 您知道 **IdM 管理员**密码。

- 您已创建了非对称密码库用于存储服务机密。
- 您已为 IdM 主机上运行的 web 服务生成了一个新的 **httpd** 密钥，以替换已被破坏的旧密钥。
- 新 **httpd** 密钥存储在本地 Ansible 控制器上，例如 **/usr/share/doc/ansible-freeipa/playbooks/vault/private-key-to-an-externally-certificate.pem** 文件中。

步骤

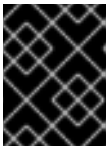
1. 导航到 **/usr/share/doc/ansible-freeipa/playbooks/vault** 目录：

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

2. 打开清单文件，并确保正确定义了以下主机：
   - 在 **[ipaserver]** 部分中的 IdM 服务器。
   - 要在 **[webservers]** 部分中检索 secret 的主机。例如，要指示 Ansible 获取到 **webserver1.idm.example.com** 和 **webserver2.idm.example.com** 的 secret，请输入：

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

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



重要

确保列表不包含被入侵的 webserver，在当前的示例 **webserver3.idm.example.com** 中。

3. 打开 **data-archive-in-asymmetric-vault-copy.yml** 文件进行编辑。
4. 通过在 **ipavault** 任务部分设置以下变量来修改该文件：
   - 将 **ipaadmin_password** 变量设置为 IdM 管理员密码。
   - 将 **name** 变量设置为 vault 的名称，如 **secret_vault**。
   - 将 **service** 变量设置为 vault 的服务所有者，如 **HTTP/webserver.idm.example.com**。
   - 将 **in** 变量设置为 **"{{ lookup('file', 'new-private-key-to-an-externally-signed-certificate.pem') | b64encode }}"**。这可确保 Ansible 使用私钥从 Ansible 控制器上的工作目录检索文件，而不是从 IdM 服务器检索。
   - 将 **action** 变量设置为 **member**。

对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 保存这个文件。

6. 运行 playbook:

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

7. 打开 **retrieve-data-asymmetric-vault-copy.yml** 文件进行编辑。

8. 通过在 **ipavault** 任务部分设置以下变量来修改该文件：

   - 将 **ipaadmin_password** 变量设置为 IdM 管理员密码。
   - 将 **name** 变量设置为 vault 的名称，如 **secret_vault**。
   - 将 **service** 变量设置为密码库的服务所有者，如 **HTTP/webserver1.idm.example.com**。
   - 将 **private_key_file** 变量设置为用于检索服务 vault secret 的私钥的位置。
   - 将 **out** 变量设置为 IdM 服务器上您要检索 **new-private-key-to-an-externally-signed-certificate.pem** 机密的位置，如当前工作目录。
   - 将 **action** 变量设置为 **member**。
     对于当前示例为修改过的 Ansible playbook 文件：

```
---
- 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. 在 playbook 中添加一个部分，它将从 IdM 服务器检索数据文件到 Ansible 控制器：

```
---
- 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: true
      mode: 0600
```

10. 在 playbook 中添加一个部分，将检索到的 **new-private-key-to-an-externally-signed-certificate.pem** 文件从 Ansible 控制器所在的地方传输到清单文件的 **webservers** 部分所列出的 webserver 中：

```
---
- 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. 保存这个文件。

12. 运行 playbook:

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

5.6. 其他资源

- 请参阅 **/usr/share/doc/ansible-freeipa/** 目录中的 README-vault.md Markdown 文件。
- 请参阅 **/usr/share/doc/ansible-freeipa/playbooks/vault/** 目录中的 playbook 示例。