

Использование Vault в качестве источника конфигурационных свойств

Оглавление

- # Введение
- # Настройка Vault
 - # Подготовка хранилища секретов
 - # Настройка аутентификации через AppRole
- # Настройка UIDM
 - # Настройка config.xml
 - # Настройка конфигурационных свойств

Введение

HashiCorp Vault — это инструмент для централизованного и безопасного управления секретами.

RooX UIDM поддерживает использование Vault в качестве одного из источников конфигурационных свойств. Это позволяет хранить чувствительные данные (например, пароли для подключения к СУБД) не в локальных файлах, а во внешнем надёжном хранилище.

Данный документ предназначен для администраторов RooX UIDM и описывает порядок интеграции UIDM с Vault.

Настройка включает в себя следующие шаги, подробно описанные в дальнейших разделах: - подготовка Vault (создание хранилища, политики и AppRole); - настройка UIDM для использования Vault как источника конфигурации.

Настройка Vault

Подготовка хранилища секретов

- В Vault создаётся хранилище секретов (KV v2):

```
Path: "uidm_secrets/"    <-- пример имени
Type: "kv"
Version: 2
```

- В созданном хранилище добавляется путь, например: `"uidm-test"`.
- В `"uidm-test"` создаются пары ключ/значение для всех параметров, которые требуется перенести из файлов с настройками RooX UIDM (`env.properties`). Например:

```
key: com.rooxteam.webapi.database.password
```

value: password

1. Создаётся политика `uidm-policy`, предоставляющая доступ к хранилищу и указанному пути:

```
path "uidm_secrets/data/uidm-test" {  
  capabilities = [ "read", "list" ]  
}
```

ЗАМЕТКА

Сегмент `data` в `path` используется только в KV v2 (в KV v1 его нет).

Настройка аутентификации через AppRole

1. Создаётся AppRole с привязанной политикой `uidm-policy`.
2. В результате будут сгенерированы `role_id` и `secret_id`. Эти значения необходимо указать в конфигурации UIDM. UIDM будет использовать метод аутентификации `AppRole` для доступа к Vault.

Настройка UIDM

Настройка UIDM включает в себя следующие шаги, подробно описанные далее:

1. В файл `config.xml` добавляются параметры подключения к Vault.
2. В конфигурационных файлах (`env.properties`, `custom.properties` и др.) значения секретов заменяются на ссылки вида `${vault.<key>}`.
3. Выполняется перезапуск всех компонентов RooX UIDM.

Настройка config.xml

В `config.xml` необходимо добавить два новых элемента:

1. `<providers>` — включает поддержку Vault. Размещается внутри блока `<header>`.
2. `<roox-vault>` — задаёт параметры подключения к Vault. Размещается сразу после блока `<header>` и содержит атрибуты:
 - a. `base-url` — URL подключения к Vault
 - b. `role-id` и `role-secret-id` — идентификатор и секрет AppRole, сгенерированные при настройке Vault
 - c. `path` — путь до ключей в хранилище Vault (сегмент `data` указывать не требуется)

ВАЖНО

Порядок элементов в `config.xml` определяет приоритет поиска параметров. Чтобы значения из Vault имели максимальный приоритет, элемент `<roox-vault>` должен быть указан первым после `<header>`.

Пример настроек config.xml, добавленные теги выделены комментариями:

```
<configuration>
  <header>
    <result forceReloadCheck="true">
    </result>
    <!-- включаем поддержку Vault -->
    <providers>
      <provider config-tag="roox-vault"
        config-
class="com.rooxteam.config.vault.VaultConfigurationProvider"/>
    </providers>
    <!-- endof: включаем поддержку Vault -->
  </header>

  <!-- добавляем Vault как источник конфигурационных свойств -->
  <roox-vault base-url="http://vault:8200"
    role-id="ad5d0a6e-4acd-4f53-ba5f-245b115138c6"
    role-secret-id="0b441273-7f01-4325-aa61-09bf35148878"
    path="uidm_secrets/uidm-test"/>
  <!-- endof: добавляем Vault как источник конфигурационных свойств -->

  <properties fileName="custom.properties" encoding="UTF-8">
    <reloadingStrategy refreshDelay="60000"
      config-
class="org.apache.commons.configuration.reloading.FileChangedReloadingStrategy"/>
  </properties>
</configuration>
```

Настройка конфигурационных свойств

В файлах конфигурации (`env.properties` , `custom.properties` и др.) значения параметров заменяются на конструкции вида `${vault.<key>}` , где:

- `vault` — служебная часть, указывающая на использование Vault;
- `<key>` — имя ключа в Vault, заданное при его конфигурации.

Пример:

```
com.rooxteam.webapi.database.password=${vault.com.rooxteam.webapi.database.password}
```

