Рекомендации

В разделе описаны рекомендации по работе со StarVault.

1. Содержание раздела

Механизмы управления секретами

1. Общие сведения

Механизмы управления секретами - это компоненты, которые хранят, генерируют или шифруют данные. Механизмы управления секретами невероятно гибкие, поэтому проще всего думать о них в терминах их функций. Механизмам управления секретами предоставляется некоторый набор данных, они выполняют некоторые действия с этими данными и возвращают результат.

В зависимости от требований механизмы управления секретами могут:

- Просто хранить и читать данные — подобно зашифрованному Memcached.
- Подключаться к различным сервисам и генерировать динамические учетные данные по требованию.
- Предоставлять шифрование как услугу, генерацию totp, сертификаты и многое другое.

Механизмы управления секретами активируются по пути в StarVault. Когда в StarVault поступает запрос, маршрутизатор автоматически направляет все, что имеет префикс маршрута, к соответствующему механизму. Таким образом, каждый механизм секретов определяет свои собственные пути и свойства. Для пользователя механизмы управления секретами ведут себя подобно виртуальной файловой системе, поддерживая такие операции, как чтение, запись и удаление.

2. Жизненный цикл механизмов управления секретами

Большинство механизмов секретов можно включать, отключать, настраивать и перемещать с помощью CLI, API или UI.

К этапам жизненного цикла механизмов относятся:

- **Enable** - активация механизма секретов по указанному пути. За некоторыми исключениями, механизмы могут быть включены по нескольким путям. Каждый механизм секретов изолирован в рамках своего пути. По умолчанию они активируются по их "типу" (например, "ssh" активируется по пути **ssh/**).
- **Disable** - отключение существующего механизма секретов. При отключении механизма аннулируются (если это поддерживается) все его секреты, а все данные, хранящиеся для этого механизма в физическом слое хранения, удаляются.

- **Move** - перемещение пути для существующего механизма секретов. В результате этого процесса все секреты аннулируются, поскольку аренда секретов связана с путем, где они были созданы. Конфигурационные данные, хранящиеся для механизма, сохраняются после перемещения.
- **Tune** - настройка глобальной конфигурации механизма секретов, например TTL.

После активации механизма управления секретами вы можете взаимодействовать с ним напрямую по его пути в соответствии с его собственным API. Используйте команду `starvault path-help`, чтобы определить пути, на которые он отвечает.

Обратите внимание, что точки монтирования в StarVault не могут конфликтовать друг с другом. Этот факт имеет два серьёзных последствия:

- Вы не можете иметь точку монтирования, которая начинается с уже существующей точки монтирования.
- Вы не можете создать точку монтирования с именем, которое является префиксом уже существующей точки монтирования.

Например, точки монтирования **foo/bar** и **foo/baz** могут сосуществовать, а **foo** и **foo/baz** нет.

2.1. Управление жизненным циклом

В следующей таблице представлены возможные операции с механизмами и способы их выполнения.

Операция	**Выполнение в UI**	**Выполнение в CLI**	**Дополнительная информация**
Просмотр списка доступных механизмов	На вкладке **Secrets**	`starvault secrets list`	При использовании CLI возможно управление выводом списка механизмов с помощью следующих опций: • `-detailed` - включает подробный вывод параметров механизмов, таких как TTL, параметры репликации, версия плагина и т.д. • `-format=<string>` - указывает способ представления вывода. Допустимые варианты: `table` (по умолчанию), `json` и `yaml`.

Операция	Выполнение в UI	Выполнение в CLI	Дополнительная информация
Активация	На вкладке **Secrets**	`starvault secrets enable [options] <secret-name>`	Как при использовании UI, так и CLI при активации можно задать дополнительные параметры механизма. Список параметров и их описание см. в разделе Общие параметры для механизмов.
Деактивация	На вкладке **Secrets** по кнопке **[Disable]** в дополнительном меню механизма ( )	`starvault secrets disable <path>`	
Перемещение	Недоступно	`starvault secrets move <old-path> <new-path>`	
Настройка	Недоступно	`starvault secrets tune <options> <path>`	Список параметров и их описание см. в разделе Общие параметры для механизмов.

2.2. Общие параметры для механизмов

В таблице ниже представлены возможные параметры для настройки механизмов и их описание.

Для конкретных механизмов могут быть доступны не все параметры. Подробнее см. в описании соответствующего механизма.

Опция команды CLI	Поле в UI	Описание параметра
`-allowed-managed-keys=<string>`		Используется для указания списка управляемых ключей, к которым конкретная точка монтирования механизма секретов имеет разрешение на доступ. Каждая указанная опция `-allowed-managed-keys`, включает один ключ, который будет добавлен в список разрешенных. Если нужно указать несколько ключей, необходимо использовать эту опцию несколько раз с разными ключами. Например: `starvault secrets tune -allowed-managed-keys=key1 -allowed-managed-keys=key2 <path>` В этом примере `key1` и `key2` являются именами ключей, к которым точка монтирования по пути `<path>` будет иметь доступ.
`-allowed-response-headers=<string>`		Используется для настройки списка заголовков ответа, которые разрешено возвращать клиенту при запросах к механизму секретов. Например: `starvault secrets tune -allowed-response-headers=Cache-Control -allowed-response-headers=Content-Type <path>` В этом примере заголовки `Cache-Control` и `Content-Type` разрешены для включения в ответы, отправляемые с точки монтирования по пути `<path>`.

Опция команды CLI	Поле в UI	Описание параметра
`-audit-non-hmac-request-keys=<string>`	**Request keys excluded from HMACing in audit**	Используется для указания списка ключей запроса, которые должны быть залогированы в журналах аудита в незашифрованном виде. По умолчанию, когда запрос записывается в журнал аудита, все его ключи и значения обычно защищены с помощью хэширования HMAC, чтобы предотвратить утечку конфиденциальной информации через журналы. Например: `starvault secrets tune -audit-non-hmac-request-keys=key1 -audit-non-hmac-request-keys=key2 <path>` В этом примере `key1` и `key2` - это ключи запроса, которые будут залогированы в журналах аудита без применения HMAC, для точки монтирования по пути `<path>`.
`-audit-non-hmac-response-keys=<string>`	**Response keys excluded from HMACing in audit**	Используется для указания списка ключей ответа, которые должны быть залогированы в журналах аудита в незашифрованном виде. Как и в случае с ключами запроса, когда ответ записывается в журнал аудита, все его ключи и значения обычно защищены с помощью хэширования HMAC, чтобы предотвратить утечку конфиденциальной информации через журналы. Например: `starvault secrets tune -audit-non-hmac-response-keys=key1 -audit-non-hmac-response-keys=key2 <path>` В этом примере `key1` и `key2` - это ключи в ответах, которые будут залогированы в журналах аудита без HMAC, для точки монтирования по пути `<path>`.

Опция команды CLI	Поле в UI	Описание параметра
`-default-lease-ttl=<duration>`	**Default Lease TTL**	Используется для установки времени жизни (TTL) по умолчанию для всех аренд (leases), выдаваемых механизмом секретов, к которому применяется эта настройка. TTL определяет, как долго секреты или токены будут действительны, прежде чем они истекут, если не будет выполнено их обновление. Если TTL истекает, и аренда не была продлена, то секреты автоматически аннулируются и становятся недоступными для использования. Пример: `starvault secrets tune -default-lease-ttl=1h <path>` В этом примере для точки монтирования по пути `<path>` устанавливается TTL по умолчанию равный одному часу (`1h`).
`-description=<string>`	**Description**	Описание механизма секретов.
`-force-no-cache` Опцию можно активировать только при включении механизма.		Используется для отключения кэширования для определенной точки монтирования механизма секретов. Когда эта опция включена, StarVault не будет кэшировать ответы от механизма, что означает, что каждый запрос к этой точке монтирования будет обрабатываться без использования кэшированных данных. Пример: `starvault secrets enable -force-no-cache kv`

Опция команды CLI	Поле в UI	Описание параметра
`-listing-visibility=<string>`	Опция **List method when unauthenticat-ed**	Определяет, какие ключи будут видны при выполнении запроса типа `list` в определенной точке монтирования механизма. Эта настройка влияет на то, какие данные пользователи могут видеть, когда они запрашивают список доступных ключей или путей в хранилище. Опция имеет два возможных значения: • `unauth` (соответствует активированной опции в UI) - ключи будут видны в списке даже без аутентификации. • `hidden` (соответствует деактивированной опции в UI) - ключи не будут отображаться в списке без соответствующих разрешений. Пример: `starvault secrets tune -listing-visibility=unauth <path>` В этом примере для точки монтирования по пути `<path>` устанавливается видимость списка ключей как `unauth`, что позволяет всем пользователям видеть список ключей в этой точке монтирования.
`-local` Опцию можно активировать только при включении механизма.	Опция **Local**	Используется для указания, что данный механизм секретов должен быть локальным для сервера, на котором он настроен. Это означает, что механизм не будет реплицироваться в кластерных установках StarVault, которые используют репликацию данных. Пример: `starvault secrets enable -local kv`

Опция команды CLI	Поле в UI	Описание параметра
`-max-lease-ttl=<duration>`	**Max Lease TTL**	Устанавливает максимальное время жизни (TTL) для аренды, выдаваемой механизмом секретов или методом аутентификации. Это ограничение применяется ко всем секретам и токенам, выданным через точку монтирования, и задаёт верхний предел времени, на который можно продлить аренду до её истечения. Пример: `starvault secrets tune -max-lease-ttl=24h <path>` В этом примере для точки монтирования по пути `<path>` устанавливается максимальное TTL равное 24 часам (`24h`). Это означает, что ни одна аренда, выданная через эту точку монтирования, не сможет быть продлена сверх этого времени.
`-passthrough-request-headers=<string>`	**Allowed passthrough request headers**	Используется для указания списка HTTP заголовков запроса, которые должны быть переданы через StarVault к удалённому ресурсу или сервису. Например: `starvault secrets tune -passthrough-request-headers=X-Custom-Header1,X-Custom-Header2 <path>` В этом примере заголовки `X-Custom-Header1` и `X-Custom-Header2` будут переданы через StarVault к внешнему сервису для точки монтирования по пути `<path>`.
`-path=<string>` Опцию можно активировать только при включении механизма.	**Path**	Определяет путь точки монтирования, где будет активирован определенный механизм секретов. Например: `starvault secrets enable -path=custom/path kv` В этом примере механизм типа `kv` активируется по пользовательскому пути `custom/path`.

Опция команды CLI	Поле в UI	Описание параметра
`-plugin-name=<string>` Опцию можно активировать только при включении механизма.		Используется для указания имени плагина, который будет использоваться при активации механизма секретов. Это имя должно соответствовать имени плагина, как оно зарегистрировано в системе StarVault. Например: `starvault secrets enable -path=my-custom-secrets -plugin-name=my-plugin plugin` В этом примере плагин с именем `my-plugin` активируется в качестве механизма секретов по пути `my-custom-secrets`.
`-plugin-version=<string>`		Используется для указания версии плагина, который вы хотите использовать при активации механизма секретов. Указание версии плагина необходимо, когда доступно несколько версий плагина и требуется контроль над тем, какая именно версия должна быть задействована. Пример: `starvault secrets enable -path=my-custom-secrets -plugin-name=my-plugin -plugin-version=1.2.3 plugin` В этом примере плагин с именем `my-plugin` и версией `1.2.3` активируется в качестве механизма секретов по пути `my-custom-secrets`.

Опция команды CLI	Поле в UI	Описание параметра
`-seal-wrap` Опцию можно активировать только при включении механизма.	Опция **Seal wrap**	Используется для включения функции обертывания печати (Seal Wrapping) для всего механизма секретов или определенных ключей внутри него. **Seal Wrapping** — это механизм, который использует возможности автоматического запечатывания (Auto-Seal) хранилища для дополнительной защиты конфиденциальных данных. Когда опция `-seal-wrap` включена, данные, связанные с механизмом секретов, оборачиваются дополнительным слоем шифрования, который обеспечивается функцией Auto-Seal. Это означает, что даже если кто-то получит доступ к физическому хранилищу данных, он не сможет прочитать защищенные таким образом секреты без распечатывания (unsealing) хранилища. Пример: `starvault secrets enable -seal-wrap -path=my-secrets kv` В этом примере для механизма секретов типа `kv`, активированного по пути `my-secrets`, включена функция **Seal Wrapping**, обеспечивающая дополнительный уровень защиты для данных, хранящихся в этом механизме.
`-version=<int>`	**Version**	Используется для указания версии механизма секретов KV, которую необходимо активировать. Пример: `starvault secrets enable -version=2 kv` В этом примере активируется механизм KV версии 2. KV допускает повышение версии с **v1** до **v2** с помощью `tune`, но понижение с **v2** на **v1** невозможно.

3. Представление барьера

Механизмы секретов получают представление барьера на настроенное физическое хранилище StarVault. Это очень похоже на chroot.

Когда механизм секретов активируется, генерируется случайный UUID. Этот UUID становится корнем данных для этого механизма. Каждый раз, когда этот механизм записывает данные в физическое хранилище, они снабжаются префиксом UUID папки. Поскольку слой хранения StarVault не поддерживает относительный доступ (например, **../**), это делает невозможным для активированного механизма доступ к другим данным.

Это важная особенность безопасности StarVault - даже вредоносный механизм не сможет получить доступ к данным любого другого механизма.

4. Содержание раздела

Хранилище

Как описано на странице Устройство StarVault в разделе **Архитекура**, бэкенд хранения StarVault представляет собой недоверенное хранилище, используемое исключительно для хранения зашифрованной информации.

1. Поддерживаемые бэкенды хранения

В StarVault доступны другие варианты хранилищ – дополнительную информацию см. в разделе Конфигурирование хранилища.

2. Резервные копии

Ввиду чрезвычайной гибкости возможных конфигураций хранилища StarVault сложно дать точные рекомендации по резервному копированию StarVault.

При выполнении резервного копирования StarVault следует учесть два момента:

1. Зашифрованные данные StarVault в бэкенде хранения
2. Файлы конфигурации и скрипты управления для запуска сервера StarVault.

Задумайтесь над вопросом: от какой беды пытаетесь защититься, сохраняя резервную копию?

2.1. Цель резервного копирования

Во время разработки стратегии постоянного резервного копирования и аварийного восстановления важно ответить на вопрос: "Зачем делать резервную копию?".

Сделать резервную копию рекомендуется перед обновлением, поскольку понижение версии хранилища StarVault не всегда возможно. Резервное копирование рекомендуется делать всякий раз, когда планируется внесение серьезных изменений в кластер.

В частности, рекомендуем делать резервные копии до, а не во время операций записи в API `/sys` (за исключением конечных точек `/sys/leases`, , `/sys/tools`, `/sys/wrapping`, `/sys/policies` и `/sys/pprof`). К примерам рабочих процессов, которые записывают данные в API `/sys`, относятся обновления и повторные ключи. В будущем для бэкенда встроенного хранилища эта инструкция может измениться.

Резервное копирование также может помочь в случае случайного удаления или изменения данных. Однако в этом случае возникают некоторые нюансы. Если, например, сегодня в 10 утра восстановитесь с резервной копии, сделанной в 5 утра и содержащей правильные данные, то будут потеряны данные, записанные с 5 и 10 утра.

Не рекомендуем использовать резервные копии как защиту от сбоя отдельной машины. Серверы StarVault могут работать в кластерах, поэтому для защиты от сбоя сервера рекомендуем запускать StarVault в режиме высокой доступности. Кластер StarVault может охватывать несколько зон доступности в пределах региона.

При использовании StarVault в режиме высокой доступности резервное копирование может помочь защититься от сбоя ЦОД.

В конечном счете резервное копирование не является заменой режиму высокой доступности. При разработке плана восстановления после (или защиты от) сбоя резервное копирование и режим высокой доступности следует рассматривать как ключевые компоненты этого плана.

2.2. Резервное копирование сохраненных данных

В идеале резервное копирование и восстановление нужно выполнять, когда StarVault не подключено к сети (офлайн). Если это невозможно, рекомендуем использовать бэкенд хранения, который поддерживает атомарные моментальные снимки (например, Встроенное хранилище).

Если бэкенд хранения не поддерживает атомарные моментальные снимки, тогда рекомендуем создавать только автономные (офлайн) резервные копии.

2.3. Конфигурирование

Помимо резервного копирования зашифрованных данных StarVault через бэкенд хранения, также можете сохранить файлы конфигурации сервера, скрипты для управления службой StarVault и убедиться, что сможете переустановить установленные пользователем плагины. Местоположение этих файлов будет зависеть установки StarVault.

Хотя резервная копия или моментальный снимок данных StarVault из бэкенда хранения зашифрованы, какие-то элементы конфигурации могут быть конфиденциальными (например, токен StarVault для автоматического распечатывания средствами Transit (Transit Autounseal) или закрытый ключ TLS в конфигурации). Присутствие этой информации в резервных копиях означает, что ее необходимо тщательно защищать.

3. Содержание раздела

- Распечатывание хранилища
- Ротация ключей
- Встроенное хранилище