Прокси

StarVault Proxy направлен на устранение первоначального препятствия для внедрения StarVault, предоставляя более масштабируемый и простой способ интеграции приложений с StarVault. StarVault Proxy действует как API Proxy для StarVault и может опционально разрешать или заставлять взаимодействующих клиентов использовать его автоматически аутентифицированный токен.

StarVault Proxy — это клиентский демон, предоставляющий следующие функции:

- Auto-Auth — автоматическая аутентификация в StarVault и управление процессом обновления токенов для локально извлеченных динамических секретов.
- API Proxy — действует как прокси для API StarVault, опционально используя (или принудительно используя) токен Auto-Auth.
- Кэширование — позволяет кэшировать на стороне клиента ответы, содержащие недавно созданные токены, и ответы, содержащие арендованные секреты, сгенерированные из этих недавно созданных токенов. Агент также управляет обновлениями кэшированных токенов и аренд.

1. Автоматическая аутентификация

StarVault Proxy позволяет легко аутентифицироваться в StarVault в самых разных средах. Информацию см. в документации автоматической аутентификации.

Функциональность автоматической аутентификации реализуется в блоке конфигурации `auto_auth`.

2. API-прокси

Основная цель StarVault Proxy — выступать в качестве API-прокси для StarVault, позволяя вам взаимодействовать с API StarVault через слушателя. Его можно настроить так, чтобы он опционально разрешал или принудительно включал автоматическое использование токена `Auto-Auth` для этих запросов.

Функциональность API Proxy реализуется в рамках определенного `listener`, и его поведение можно настроить с помощью блока конфигурации `api_proxy`.

3. Кэширование

StarVault Proxy позволяет кэшировать на стороне клиента ответы, содержащие недавно созданные токены, и ответы, содержащие арендованные секреты, сгенерированные из недавно созданных токенов. Для получения более подробной информации читайте документацию по кэшированию.

4. API

4.1. Выход

Эта конечная точка запускает отключение прокси. По умолчанию она отключена и может быть включена для каждого слушателя с помощью блока конфигурации `proxy_api`. Рекомендуется включать ее только на доверенных интерфейсах, так как для ее использования не требуется никакой авторизации.

Метод	Путь
`POST`	`/proxy/v1/quit`

4.2. Кэш

Подробную информацию об API кэширования смотрите на странице кэширования.

5. Конфигурация

5.1. Параметры команды

- `-log-level (string: "info")` — уровень детализации журнала. Поддерживаемые значения (в порядке убывания детализации): `trace`, `debug`, `info`, `warn` и `error`. Это также можно указать с помощью переменной среды `STARVAULT_LOG_LEVEL`.
- `-log-format (string: "standard")` — формат журнала. Поддерживаемые значения: `standard` и `json`. Это также можно указать с помощью переменной среды `STARVAULT_LOG_FORMAT`.
- `-log-file` — абсолютный путь, по которому StarVault Proxy должен сохранять сообщения журнала. Пути, заканчивающиеся разделителем пути, используют имя файла по умолчанию `proxy.log`. Пути, не заканчивающиеся расширением файла, используют расширения по умолчанию `.log`. Если файл журнала ротируется, StarVault Proxy добавляет текущую временную метку к имени файла во время ротации. Например:

`log-file`	Полный файл журнала	Измененный файл журнала
`/var/log`	`/var/log/proxy.log`	`/var/log/proxy-{timestamp}.log`
`/var/log/my-diary`	`/var/log/my-diary.log`	`/var/log/my-diary-{timestamp}.log`
`/var/log/my-diary.txt`	`/var/log/my-diary.txt`	`` `/var/log/my-diary-{timestamp}.txt ``

- `-log-rotate-bytes` — для указания количества байтов, которые должны быть записаны в журнал, прежде чем его нужно будет ротировать. Если не указано иное, ограничений на количество байтов, которые могут быть записаны в файл журнала, нет.

- `-log-rotate-duration` — для указания максимальной продолжительности записи журнала, прежде чем его нужно будет ротировать. Должно быть значение продолжительности, например 30 с. По умолчанию 24 ч.

- `-log-rotate-max-files` — для указания максимального количества старых архивов файлов журнала для хранения. По умолчанию 0 (файлы никогда не удаляются). Установите значение -1, чтобы отбрасывать старые файлы журнала при создании нового.

5.2. Параметры файла конфигурации

В настоящее время доступны следующие параметры конфигурации:

- `vault` (`vault`: <optional>) — указывает удаленный сервер StarVault, к которому подключается Proxy.
- `auto_auth` (`auto_auth`: <optional>) — указывает метод и другие параметры, используемые для функциональности Auto-Auth.
- `` `api_proxy (`api_proxy` `` : <optional>) — указывает параметры, используемые для функциональности API Proxy.
- `cache` (`cache`: <optional>) — указывает параметры, используемые для функциональности кэширования.
- `listener` (`listener`: <optional>) — указывает адреса и порты, на которых Proxy будет отвечать на запросы.

В `SIGHUP` (`kill -SIGHUP $(pidof StarVault)`) StarVault Proxy попытается перезагрузить конфигурацию слушателя TLS. Этот метод можно использовать для обновления сертификатов, используемых StarVault Proxy, без необходимости перезапускать его процесс.

- `pid_file (string: "")` — путь к файлу, в котором должен храниться идентификатор процесса (PID) прокси-сервера.
- `exit_after_auth (bool: false)` — если установлено значение `true`, прокси-сервер завершит работу с кодом `0` после одной успешной аутентификации, где успех означает, что токен был извлечен и все приемники успешно его записали.
- `disable_idle_connections (string array: [])` — список строк, отключающих неактивные соединения для различных функций в StarVault Proxy. Допустимые значения включают: `автоматическую аутентификацию` и `проксирование`. Также можно настроить, установив переменную среды `VAULT_PROXY_DISABLE_IDLE_CONNECTIONS` в виде строки, разделенной запятыми. Эта переменная среды переопределит любые значения, найденные в файле конфигурации.
- `disable_keep_alives (string array: [])` — список строк, отключающих поддержку соединений для различных функций в StarVault Agent. Допустимые значения включают: `автоматическую аутентификацию` и `проксирование`. Также можно настроить, установив переменную среды `VAULT_PROXY_DISABLE_KEEP_ALIVES` как строку, разделенную запятыми. Эта переменная среды переопределит любые значения, найденные в файле конфигурации.
- `template` (template: <optional>) — определяет параметры, используемые для шаблонизации секретов StarVault в файлах.
- `template_config` (template_config: <optional>) — определяет поведение механизма шаблонизации.
- `telemetry` (`telemetry`: <optional>) — определяет систему отчетов телеметрии. Список метрик, специфичных для Proxy, см. в разделе `Telemetry Stanza` ниже.
- `log_level` — эквивалентно флагу командной строки `-log-level`.

При `SIGHUP` (`kill -SIGHUP $(pidof StarVault)`) StarVault Proxy обновит уровень журнала до значения, указанного в файле конфигурации (включая переопределяющие значения, заданные с помощью CLI или параметров переменных среды).

- `log_format` - эквивалент флага командной строки `-log-format`.
- `log_file` - эквивалент флага командной строки `-log-file`.
- `log_rotate_duration` - флага командной строки `-log-rotate-duration`.
- `log_rotate_bytes` - эквивалент флага командной строки `-log-rotate-bytes`.
- `log_rotate_max_files` - эквивалент флага командной строки `-log-rotate-max-files`.

5.3. Блок конфигурации StarVault

Максимально может быть один блок хранилища верхнего уровня, и он будет иметь следующие записи конфигурации:

- `address (string: <optional>)` — адрес сервера StarVault для подключения. Это должно быть полное доменное имя (FQDN) или IP, например `https://starvault-fqdn:8200` или `https://172.16.9.8:8200`. Это значение можно переопределить, установив переменную среды `STARVAULT_ADDR`.
- `ca_cert (string: <optional>)` — путь на локальном диске к одному сертификату CA с кодировкой PEM для проверки сертификата SSL сервера StarVault. Это значение можно переопределить, установив переменную среды `STARVAULT_CACERT`.
- `ca_path (string: <optional>)` — путь на локальном диске к каталогу сертификатов CA с кодировкой PEM для проверки сертификата SSL сервера StarVault. Это значение можно переопределить, установив переменную среды `STARVAULT_CAPATH`.
- `client_cert` (`string: <optional>`) — путь на локальном диске к одному сертификату CA в кодировке PEM, который будет использоваться для аутентификации TLS на сервере StarVault. Это значение можно переопределить, установив переменную среды `STARVAULT_CLIENT_CERT`.
- `client_key` (`string: <optional>`) — путь на локальном диске к одному закрытому ключу в кодировке PEM, совпадающему с клиентским сертификатом из `client_cert`. Это значение можно переопределить, установив переменную среды `STARVAULT_CLIENT_KEY`.
- `tls_skip_verify` (`string: <optional>`) — отключение проверки сертификатов TLS. Использование этого параметра настоятельно не рекомендуется, поскольку он снижает безопасность передачи данных на сервер StarVault и с него. Это значение можно переопределить, установив переменную среды `STARVAULT_SKIP_VERIFY`.
- `tls_server_name` (`string: <optional>`) — имя для использования в качестве SNI хоста при подключении через TLS. Это значение можно переопределить, установив переменную среды `STARVAULT_TLS_SERVER_NAME`.

Повтор блока конфигурации

Блок конфигурации `хранилища` может содержать конфигурацию `повтора`, которая управляет тем, как обрабатываются неудачные запросы хранилища. Однако у автоматической аутентификации есть собственное понятие повтора, и этот раздел на него не влияет.

Далее представлены варианты конфигурации повтора:

- `num_retries (int: 12)` — указывает, сколько раз будет повторен неудачный запрос. Значение `0` соответствует значению по умолчанию, т. е. 12 повторов. Значение `-1`

отключает повторы. Переменная окружения `STARVAULT_MAX_RETRIES` переопределяет эту настройку.

Запросы, исходящие из кэша прокси-сервера, будут повторяться только в том случае, если они привели к определенным кодам результата HTTP: любой код 50x, кроме 501 («не реализовано»), а также 412 («предварительное условие не выполнено»). Запросы, исходящие из подсистемы шаблонов, повторяются независимо от сбоя.

5.4. Блок конфигурации слушателя (listener)

StarVault Proxy поддерживает один или несколько блоков конфигурации слушателя. Слушатели могут быть настроены с кэшированием или без него, но будут использовать кэш, если он был настроен, и включат API proxy. В дополнение к стандартной конфигурации слушателя, конфигурация слушателя Proxy также поддерживает следующее:

- `require_request_header (bool: false)` — требует, чтобы все входящие HTTP-запросы на этом слушателе имели запись заголовка `X-Vault-Request: true`. Использование этой опции обеспечивает дополнительный уровень защиты от атак Server Side Request Forgery. Запросы на слушателе, не имеющие надлежащего заголовка `X-Vault-Request`, будут отклонены с кодом статуса HTTP-ответа `412: Precondition Failed`.
- `role (string: default)` — `role` определяет, какие API обслуживает слушатель. Его можно настроить на `metrics_only` для обслуживания только метрик или на роль по умолчанию `default`, которая обслуживает все (включая метрики). `require_request_header` не применяется к слушателям `metrics_only`.
- `proxy_api (<optional>)` — управляет дополнительными конечными точками Proxy API.

5.4.1. Блок конфигурации proxy_api

- `enable_quit (bool: false)` — если установлено значение `true`, прокси-сервер включит API выхода.

5.5. Блок конфигурации телеметрии (telemetry)

StarVault Proxy поддерживает конфигурацию телеметрии и собирает различные показатели: времени выполнения, производительности, автоматической аутентификации и состоянии кэша. Далее в таблице представлены метрики:

Метрика	Описание	Тип
`vault.proxy.auth.failure`	Количество неудачных попыток аутентификации	Счетчик

Метрика	Описание	Тип
`vault.proxy.auth.success`	Количество успешных аутентификаций	Счетчик
`vault.proxy.proxy.success`	Количество успешно проксированных запросов	Счетчик
`vault.proxy.proxy.client_error`	Количество запросов, на которые StarVault вернул ошибку	Счетчик
`vault.proxy.proxy.error`	Количество запросов, которые не удалось обработать прокси-серверу	Счетчик
`vault.proxy.cache.hit`	Количество попаданий в кэш	Счетчик
`vault.proxy.cache.miss`	Количество промахов кэша	Счетчик

6. Запуск прокси-сервера StarVault

Чтобы запустить StarVault Proxy:

1. Скопируйте двоичный файл StarVault туда, где запускается клиентское приложение (виртуальная машина, модуль Kubernetes и т. д.)
2. Создайте файл конфигурации StarVault Proxy. (Пример конфигурации см. в разделе Пример конфигурации.)
3. Запустите StarVault Proxy с файлом конфигурации.

6.1. Пример

```bash
starvault proxy -config=/etc/starvault/proxy-config.hcl
```

Чтобы получить помощь, выполните:

```bash
starvault proxy -h
```

Как и в случае со StarVault, флаг `-config` можно использовать тремя различными способами:

- Используйте флаг один раз, чтобы указать путь к одному конкретному файлу конфигурации.
- Используйте флаг несколько раз, чтобы указать несколько файлов конфигурации, которые будут составлены во время выполнения.

- Используйте флаг, чтобы указать каталог файлов конфигурации, содержимое которых будет составлено во время выполнения.

7. Пример конфигурации

Ниже приведен пример конфигурации с сильно измененнными значениями:

```
pid_file = "./pidfile"

vault {
  address = "https://starvault-fqdn:8200"
  retry {
    num_retries = 5
  }
}

auto_auth {
  method "approle" {
    mount_path = "auth/approle"
    config = {
      role_id_file_path = "/etc/starvault/role_id"
      secret_id_file_path = "/etc/starvault/secret_id"
    }
  }

  sink "file" {
    config = {
      path = "/tmp/file-foo"
    }
  }

  sink "file" {
    wrap_ttl = "5m"
    aad_env_var = "TEST_AAD_ENV"
    dh_type = "curve25519"
    dh_path = "/tmp/file-foo-dhpath2"
    config = {
      path = "/tmp/file-bar"
    }
  }
}

cache {
  // An empty cache stanza still enables caching
}

api_proxy {
  use_auto_auth_token = true
}
```

```
listener "unix" {
  address = "/path/to/socket"
  tls_disable = true

  agent_api {
    enable_quit = true
  }
}

listener "tcp" {
  address = "127.0.0.1:8100"
  tls_disable = true
}
```

Методы автоматической аутентификации. JWT

Метод `jwt` считывает JWT из файла и отправляет его методу JWT Auth.

1. Конфигурация

- `path (string: required)` - путь к файлу JWT
- `role (string: required)` - роль для аутентификации в StarVault.
- `remove_jwt_after_reading (bool: optional, defaults to true)` - это значение может быть установлено в `false`, чтобы отключить стандартное поведение удаления JWT после его чтения.
- `remove_jwt_follows_symlinks (bool: optional, defaults to false)` - это значение может быть установлено в `true`, чтобы следовать симлинкам при удалении JWT после его прочтения при выполнении поведения `remove_jwt_after_reading`. Если установлено значение `false`, будет удалена симссылка, а не JWT. Ничего не делает, если `remove_jwt_after_reading` равно `false`.
- `jwt_read_period (duration: «0.5s», optional)` - период времени, после которого Агент будет пытаться прочитать JWT, хранящийся по адресу `path`. По умолчанию равен `1м`, если `remove_jwt_after_reading` установлен в `true`, или `0,5c` в противном случае. Используются строки формата duration.

Методы автоматической аутентификации. Kerberos

Метод автоматической аутентификации `kerberos` обеспечивает автоматический механизм получения токена StarVault для субъектов Kerberos. Он считывает конфигурационную и идентификационную информацию из окружающей среды и использует ее для аутентификации в StarVault.

Подробнее об этом методе аутентификации см. в разделе Метод аутентификации Kerberos.

1. Конфигурация

- `krb5conf_path (string: requiredo)` - это путь к правильному файлу `krb5.conf`, описывающему взаимодействие со средой Kerberos.
- `keytab_path (string: required)` - это путь к `keytab`, в которой хранится запись для субъекта, аутентифицирующегося в StarVault. Файлы `keytab` должны быть защищены от других пользователей на общем сервере с помощью соответствующих разрешений на файлы.
- `username (string: required)` - это имя пользователя для записи в `keytab`, которое будет использоваться для входа в Kerberos. Это имя пользователя должно совпадать с учетной записью службы в LDAP.
- `service (string: required)` - это имя участника службы, которое будет использоваться при получении заявки на обслуживание для получения токена SPNEGO. Эта служба должна существовать в LDAP.
- `realm (string: required)` - это имя области Kerberos. Эта область должна соответствовать домену UPN, настроенному для подключения LDAP. При проверке учитывается регистр символов.
- `disable_fast_negotiation (bool: optional)` - предназначен для отключения метода авторизации Kerberos, который по умолчанию использует быстрое согласование. FAST - это платформа предварительной аутентификации для Kerberos. Она включает механизм туннелирования обмена данными перед аутентификацией с использованием защищенных сообщений KDC. FAST обеспечивает повышенную устойчивость к пассивным атакам с подбором пароля. Некоторые распространенные реализации Kerberos не поддерживают быстрое согласование. Значение по умолчанию равно `false`.