Предельные и максимальные значения в StarVault

StarVault устанавливает предельные значения для размера некоторых полей и объектов. Предельные значения могут быть как верхними фиксированными, так и настраиваемыми. Так же базовое хранилище накладывает верхние ограничения на StarVault В статье перечисляются предельные значения, для планирования развертывания StarVault.

В некоторых случаях у системы могут возникнуть проблемы с производительностью до того, как будут достигнуты абсолютные пределы.

1. Предельные значения, связанные с хранилищем

1.1. Размер записи в хранилище

Максимальный размер объекта, записываемого в бэкенд хранения, определяется самим бэкендом.

По умолчанию предельный размер записи для бэкенда встроенного хранилища составляет 1 МиБ. Допустимый размер записи настраивается с помощью параметра `max_entry_size` в строке файла конфигурации хранилища.

Прежде чем вносить запись в Raft, StarVault автоматически разбивает любую запись в хранилище, размер которой больше 512 КиБ, но меньше `max_entry_size`.

Для развертываний StarVault, которые используют бэкенд хранения Consul, размер записи по умолчанию ограничен 512 КиБ. Размер по умолчанию устанавливает Consul, а не StarVault. Предельное значение размера записи задается с помощью параметра `kv_max_value_size` в Consul.

Однако Consul не разбивает записи в хранилище на части, как это делает StarVault, а хранит как одну большую запись. Даже небольшие изменения могут привести к удлинению циклов чтения-модификации-записи для записей хранилища, ухудшая производительность StarVault. Большие записи также могут нарушить работу кластера Consul, задерживая передачу сигналов подтверждения работоспособности, что приведет к частой и бессистемной передаче роли ведущего между узлами.

Другие предельные значения в StarVault зависят от максимального размера записи хранилища, как описано в следующих разделах. Чтобы избавиться от ошибки возникающей,

когда размер записи хранилища достигает максимума, выполните реконфигурацию StarVault или Consul и увеличьте максимальный размер записи хранилища.

1.2. Предельные значения для точек монтирования

Все точки монтирования механизма секретов и точки монтирования аутентификации должны умещаться в одной записи хранилища. Размер каждого объекта JSON с описанием точки монтирования составляет около 500 байт, но в сжатом виде и обычно занимает около 75 байт. Каждая из точек монтирования аутентификации, механизма секретов, чисто локальных методов аутентификации и чисто локального механизма секретов хранится отдельно, поэтому ограничение применяется к каждой из них отдельно.

Параметр	**Значение по умолчанию для Consul (512 КиБ)**	**Значение по умолчанию для встроенного хранилища (1 МиБ)**
Максимальное количество точек монтирования механизма секретов	~7000	~14000
Максимальное количество подключенных методов аутентификации	~7000	~14000
Максимальная длина точки монтирования	нет принудительного ограничения	нет принудительного ограничения

Указание особых параметров для каждой точки монтирования или использование длинных путей к точкам монтирования увеличивает объем пространства, требуемого для каждой точки монтирования.

Считывая конечные точки `sys/auth` и `sys/mounts` можно отследить количество точек монтирования.

Как вариант, используйте метрики телеметрии `vault.core.mount_table.num_entries` и `vault.core.mount_table.size`, чтобы отслеживать количество точек монтирования и размер каждой таблицы монтирования.

1.3. Предельные значения для объекта и группы

Метаданные, которые прикрепляются к объекту. "удостоверение" или группе объектов, подпадают под следующие ограничения:

Параметр	**Предельное значение, байт**
Количество пар ключ-значение в метаданных	64

Параметр	Предельное значение, байт
Размер ключа метаданных	128
Размер значения метаданных	512

StarVault разбивает объекты на сегменты, распределяя их между 256 записями хранилища. Как следствие, жесткое предельное значение: 128 МиБ пространства для объектов в Consul или 256 МиБ во встроенном хранилище с настройками по умолчанию. Псевдонимы объектов хранятся внутри элементов типа "Объект" и поэтому занимают тот же пул памяти. Определения объектов сжимаются в каждой записи хранилища, и размер до сжатия зависит от количества псевдонимов объектов и объема метаданных. Минимально заполненные объекты после сжатия занимают около 200 байт.

Определения групп хранятся отдельно в собственном пуле из 256 записей хранилища. Размер каждого элемента типа "Группа" зависит от количества членов группы и объема метаданных. Псевдонимы группы и информация о ее составе хранятся в каждом элементе типа "Группа". Группа без метаданных, которая содержит 10 объектов, будет занимать около 500 байт, а группа со 100 объектами будет занимать около 4 000 байт.

В таблице ниже приведены как наилучшие, так и консервативные оценки для объектов и групп. Значения указаны чуть меньше, чем тот объем, который помещается в один сегмент, так как первый же заполненный сегмент будет вызывать сбои. Максимальное значение уменьшится, если каждый объект будет содержать большой объем метаданных или каждая группа будет иметь большое количество членов.

Параметр	Значение по умолчанию для Consul (512 КиБ)	Значение по умолчанию для встроенного хранилища (1 МиБ)
Максимальное количество объектов типа "удостоверение" (наилучшая оценка – 200 байт на объект)	~610,000	~1,250,000
Максимальное количество объектов типа "удостоверение" (консервативная оценка – 500 байт на объект)	~250,000	~480,000
Максимальное количество объектов типа "удостоверение" (максимально разрешенный объем метаданных – 41 160 байт на объект)	670	2,400
Максимальное количество групп (10 объектов на группу)	~250,000	~480,000

Параметр	Значение по умолчанию для Consul (512 КиБ)	Значение по умолчанию для встроенного хранилища (1 МиБ)
Максимальное количество групп (100 объектов на группу)	~22,000	~50,000
Максимальное количество членов в группе	~11,500	~23,000

Количество объектов идентификации, которые отслеживаются с помощью телеметрии StarVault — `vault.identity.num_entities`.

Накладные расходы на обновление объектов и групп растут по мере увеличения количества объектов в каждом сегменте. Эти расходы отслеживаются с помощью метрик `vault.identity.upsert_entity_txn` и `vault.identity.upsert_group_txn`.

Не стоит создавать очень большие внутренние группы (больше 1 000 членов), поскольку список членов группы должен хранится в одной записи хранилища. Вместо этого попробуйте использовать внешние группы или разделить группу на подгруппы.

1.4. Предельные значения для токенов

Для каждого токена используется одна запись хранилища, поэтому максимальное количество активных токенов не ограничено. На поле метаданных токена не налагается никаких ограничений, кроме того, что токен должен помещаться в одну запись хранилища:

Параметр	Предельное значение
Количество пар ключ-значение в метаданных	без ограничений
Размер ключа метаданных	без ограничений
Размер значения метаданных	без ограничений
Общий объем метаданных токена	512 КиБ

1.5. Предельные размеры политик

Максимальный размер политики ограничен размером записи хранилища. Списки политик, которые используются в токенах или объектах, должны помещаться в одной записи хранилища.

Параметр	Значение по умолчанию для Consul (512 КиБ)	Значение по умолчанию для встроенного хранилища (1 МиБ)
Максимальный размер политики	512 КиБ	1 МиБ

Параметр	Значение по умолчанию для Consul (512 КиБ)	Значение по умолчанию для встроенного хранилища (1 МиБ)
Максимальное количество политик на токен	~14,000	~28,000
Максимальное количество политик на объект или группу	~14,000	~28,000

При каждом использовании токена StarVault собирает коллекцию прикрепленных политик: к токену, к объекту, к любым группам, в которые входит объект, и к рекурсивно к любым группам, в которые входят эти группы. Очень большое количество политик использовать можно, но это приведет к увеличению времени отклика StarVault. Отслеживайте метрику `vault.core.fetch_acl_and_token`, чтобы определить, не становится ли время, необходимое для составления списка контроля доступа, чрезмерно большим.

1.6. Хранилище пар ключ-значение с поддержкой версионирования (механизм секретов kv-v2)

Параметр	Предельное значение
Количество секретов	без ограничений, определяется доступным объемом хранилища
Максимальный размер одной версии секрета	чуть меньше размера одной записи в хранилище (512 или 1024 КиБ)
Количество версий секрета	по умолчанию 10; настраивается для каждого секрета или точки монтирования
Максимальное количество версий (не проверяется при настройке)	не менее 24 000

Каждая версия секрета должна умещаться в одной записи хранилища. Пары ключ-значение перед хранением преобразуются в формат JSON.

Метаданные каждой версии занимают 21 байт и должны помещаться в одной записи хранилища отдельно от хранимых данных.

Каждый секрет также имеет метаданные, не зависящие от версии. Эти данные могут содержать поле `custom_metadata` с парами ключ-значение, которые предоставил пользователь. StarVault устанавливает следующие предельные значения для пользовательских метаданных:

Параметр	Предельное значение, байт
Количество пар ключ-значение пользовательских метаданных	64
Размер ключа пользовательских метаданных	128
Размер значения пользовательских метаданных	512

1.7. Механизм секретов Transit

Максимальный размер зашифрованного или незашифрованного текста Transit ограничен максимальным размером запроса StarVault, как описано ниже.

Все архивные версии одного ключа должны умещаться в одну запись хранилища. Это предельное значение зависит от размера ключа.

Тип ключа	Значение по умолчанию для Consul (512 КиБ)	Значение по умолчанию для встроенного хранилища (1 МиБ)
aes128-gcm96	2008	4017
aes256-gcm96	1865	3731
chacha-poly1305	1865	3731
ed25519	1420	2841
ecdsa-p256	817	1635
ecdsa-p384	659	1318
ecdsa-p523	539	1078
1024-bit RSA	169	333
2048-bit RSA	116	233
4096-bit RSA	89	178

2. Другие предельные значения

2.1. Размер запроса

Максимальный размер HTTP-запроса в StarVault ограничивается параметром `max_request_size` в строке конфигурации обработчика. По умолчанию равен 32 МиБ.

Это значение, за вычетом размера самого HTTP-запроса, задает верхнее предельное значение для операции Transit и максимальный размер секретов в формате ключ-значение.

2.2. Длительность запроса

Максимальная продолжительность операции StarVault определяется значением `max_request_duration`, которое по умолчанию равно 90 секундам. Если определенному механизму секретов требуется больше времени для выполнения операции на удаленном сервисе, клиент StarVault увидит сбой.

Переменная окружения `VAULT_CLIENT_TIMEOUT` также задает максимальную продолжительность на стороне клиента, которая по умолчанию составляет 60 секунд.

2.3. Предельные значения для кластеров и репликаций

На максимальный размер кластера или максимальное количество реплик, связанных с ведущим кластером, на практике не налагается никаких ограничений. Однако каждая реплика или резервный рабочий узел требуют от активного узла значительных дополнительных ресурсов, поскольку каждая запись дублируется на все резервные узлы. На повторную одновременную синхронизацию нескольких реплик также уходит много ресурсов.

Следите за загрузкой процессора и сети активного узла StarVault и за задержкой между последней записью в журнал WAL и в журнал WAL реплик, чтобы определить превышение максимального количества реплик.

Параметр	**Предельное значение**
Максимальный размер кластера	без ограничений, определяется возможностями активного узла
Максимальное количество реплик аварийного восстановления	без ограничений, определяется возможностями активного узла
Максимальное количество реплик рабочих узлов	без ограничений, определяется возможностями активного узла

2.4. Предельные значения для аренды

Для предельного значения аренды настраиваются общесистемные параметры `maximum TTL` и `maximum TTL per mount point`.

Хотя технического максимального значения не существует, большое время или количество аренд снижают производительность системы. Рекомендуем устанавливать по умолчанию короткие значения времени жизни для токенов и времени аренды, чтобы не накапливалось

большое количество неистекших аренд и не происходило одновременное истечение срока действия нескольких аренд сразу.

Параметр	Предельное значение
Максимальное количество аренд	Рекомендованное предельное значение — 256 000
Максимальная продолжительность аренды или токена	768 часов по умолчанию

Текущее количество неистекших аренд отслеживактся с помощью метрики `vault.expire.num_leases`.

2.5. Предельные значения для внешних плагинов

Система плагинов запускает отдельный процесс, инициируемый StarVault, который взаимодействует посредством RPC. Для каждого механизма секретов и метода аутентификации, подключенных как внешний плагин, StarVault будет запускать процесс на хост-системе. Для механизма секретов баз данных внешние плагины баз данных будут запускать процесс для каждого настроенного подключения.

Независимо от типа плагина, каждый из этих процессов будет нагружать систему используя следующие ресурсы: процессорное время, память, сеть и файловые дескрипторы. Не существует конкретных ограничений на количество механизмов секретов, методов аутентификации или настроенных подключений к базе данных. В конечном итоге это зависит от использования ресурсов конкретным плагином, интенсивности вызовов плагина и доступных ресурсов в системе. Для однотипных плагинов каждый дополнительный процесс будет линейно увеличивать использование ресурсов. Таким образом, предполагается, что однотипные плагины используются одинаково.

Модель безопасности StarVault

Модель безопасности StarVault очень важна в силу специфики самого хранилища StarVault и конфиденциальности данных, которыми управляет. Цель модели безопасности StarVault — обеспечить конфиденциальность, целостность, доступность, контроль и учет, а также аутентификацию.

Это означает, что хранимые и передаваемые данные должны быть защищены от перехвата и искажения. Чтобы получить доступ к данным или вносить изменения в политики, клиенты должны надлежащим образом аутентифицироваться и авторизоваться. Каждое взаимодействие можно проверить и однозначно проследить до исходного объекта.Система должна быть устойчивой к преднамеренным попыткам обойти средства контроля доступа.

1. Модель угроз

Ниже показано, что входит в модель угроз StarVault:

Перехват любого взаимодействия с StarVault. Взаимодействие клиента с StarVault, StarVault с бэкендом хранения и между узлами кластера StarVault должно быть защищено от перехвата.

- Искажение хранимых или передаваемых данных. Если обнаружено искажение, то StarVault прерывает обработку транзакции.
- Доступ к данным или средствам контроля без аутентификации или авторизации. Все запросы обрабатываются в соответствии с применимыми политиками безопасности.
- Бесконтрольный доступ к данным или средствам контроля. Если ведется журнал аудита, то запросы и ответы регистрируются до того, как клиент получит какие-либо секретные материалы.
- Конфиденциальность хранимых секретов. Любые данные, которые StarVault оставляет в бэкенде хранения, должны быть защищены от перехвата. На практике это означает, что все хранимые данные должны быть зашифрованы.
- Доступность секретных материалов в случае сбоя. StarVault может работать в конфигурации высокой доступности, чтобы избежать потери доступности. Перечисленное ниже не входит в модель угроз StarVault:
- Защита от получения произвольного контроля над бэкендом хранения. Злоумышленник, имеющий возможность выполнять произвольные операции с бэкендом хранения, угрожает безопасности разными способами, от которых трудно или невозможно защититься. Например, злоумышленник может удалить или повредить все содержимое бэкенда хранения, что приведет к полной потере данных для StarVault. Контроль

операций чтения позволит злоумышленнику сделать снимок в хорошо известном состоянии и откатывать изменения состояния, если это будет ему выгодно.

- Защита от утечки информации о существовании секретных материалов. Злоумышленник, способный читать данные из бэкенда хранения, может узнать о факте существования и хранения секретных материалов, даже если те хранятся с соблюдением режима конфиденциальности.
- Защита от анализа памяти работающего StarVault. Если злоумышленник способен проверить состояние памяти работающего экземпляра StarVault, то конфиденциальность данных скомпрометирована.
- Защита от недостатков внешних систем или служб, используемых StarVault. Некоторые методы аутентификации или механизмы секретов делегируют конфиденциальные операции системам, внешним по отношению к StarVault. Если злоумышленник скомпрометировал учетные данные или иным образом эксплуатировал уязвимость в этих внешних системах, то конфиденциальность или целостность данных могут быть нарушены.
- Защита от вредоносных плагинов или выполнения кода на базовом хосте. Если злоумышленник получил права на выполнение кода или на запись на базовом хосте, то конфиденциальность или целостность данных могут быть нарушены.
- Защита от недостатков клиентов или систем, которые обращаются к StarVault. Если злоумышленник способен скомпрометировать клиент StarVault (например, систему или браузер) и получить учетные данные этого клиента в StarVault, то может получить доступ к StarVault с уровнем прав, выданных этому клиенту.
- Защита от администраторов StarVault, предоставляющих вредоносные конфигурационные данные или данные, которые сделают конфигурацию уязвимой. Любые данные, предоставляемые административным конечным точкам StarVault в качестве значений конфигурации (например, конфигурации механизмов секретов), или файлы конфигурации StarVault должны проверяться. Если злоумышленник сделает запись в конфигурацию StarVault, то конфиденциальность или целостность данных могут быть нарушены.

2. Обзор внешних угроз

Архитектура StarVault охватывает три различные системы:

- Клиент: обращается к StarVault через API.
- Сервер: предоставляет API и обслуживает запросы.
- Бэкенд хранения: используется сервером для чтения и записи данных.

Между клиентом и сервером StarVault нет взаимного доверия. Клиенты используют TLS для проверки удостоверений сервера и организации безопасного канала связи. Серверы

требуют, чтобы клиент предоставлял токен клиента для каждого запроса, который используется для идентификации клиента. Клиент, который не предоставляет токен, делает только запросы на вход в систему.

Весь серверный трафик между экземплярами StarVault в пределах кластера (т.е. высокая доступность, или встроенное хранилище) использует взаимно аутентифицированный TLS для обеспечения конфиденциальности и целостности передаваемых данных. Узлы аутентифицируются до добавления в кластер с помощью запроса на распечатывание или одноразового токена активации.

Используемые StarVault бэкенды хранения также изначально не являются доверенными. StarVault использует защитный барьер для запросов к бэкенду. Защитный барьер автоматически шифрует данные, поступающие из StarVault, 256-битным шифром по стандарту Advanced Encryption Standard (AES) в режиме счетчика Галуа (GCM) с использованием 96-битных одноразовых чисел. Одноразовое число генерируется случайным образом для каждого шифруемого объекта. Когда данные считываются с защитного барьера, тег аутентификации GCM проверяется во время расшифровки, позволяя обнаружить искажения.

В зависимости от используемого бэкенда StarVault может взаимодействовать с бэкендом по TLS, обеспечивая дополнительный уровень безопасности. Иногда, например, в случае файлового бэкенда, это неприменимо. Поскольку бэкенды хранения не являются доверенными, злоумышленник получит доступ только к зашифрованным данным, даже если взаимодействие с бэкендом будет перехвачено.

3. Обзор внутренних угроз

В системе StarVault первостепенное внимание уделяется защите от попыток неавторизованного доступа к секретным материалам. Если злоумышленник уже получил некий уровень доступа к StarVault и может аутентифицироваться, то это – внутренняя угроза.

При первой аутентификации клиента в StarVault метод аутентификации проверяет удостоверение клиента и возвращает список связанных политик контроля доступа на основе ACL. Связанные политики задаются операторами StarVault заранее. Например, пользователи LDAP из команды «engineering» могут быть сопоставлены с политиками StarVault «engineering» и «ops». Затем StarVault генерирует клиентский токен, который представляет собой случайно сгенерированное сериализованное значение, и сопоставляет со списком политик. Этот клиентский токен затем возвращается клиенту.

При каждом запросе клиент предоставляет этот токен, после чего StarVault убеждается, что токен действителен и не был отозван или просрочен, и создает ACL на основе связанных политик. По умолчанию StarVault использует строгую стратегию "все, что не разрешено,

запрещено". Это означает, что, если связанная политика не допускает определенное действие, то оно будет запрещено. Каждая политика определяет предоставленный уровень доступа к тому или иному пути в StarVault. При объединении политик (если с клиентом связано несколько политик) используется наивысший разрешенный уровень доступа. Например, если политика «engineering» дает права доступа на чтение/обновление к пути «eng/», а политика «ops» дает права доступа на чтение к пути «ops/», то пользователь получает сразу и первые права и вторые. Политика сопоставляется с самой детализированной политикой, которая может быть точным соответствием или шаблоном поиска (glob) с самым длинным префиксом. Дополнительную информацию см. в разделе «Синтаксис политики».

Определенные операции разрешают только root-пользователи — отдельная политика, встроенная в StarVault. Это похоже на концепцию root-пользователя в Unix-подобных системах или администратора в Windows. В случаях, когда клиентам предоставляются root-токены или задана связанная root-политика, StarVault поддерживает понятие привилегии «sudo». В рамках политики пользователям могут быть предоставлены привилегии «sudo» для определенных путей, чтобы они могли по-прежнему выполнять операции, критичные с точки зрения безопасности, но при этом у них не было глобального root-доступа к StarVault.

Наконец, StarVault поддерживает использование правила двух лиц для распечатывания с помощью схемы разделения секрета Шамира. StarVault запускается в запечатанном состоянии. Это означает, что ключ шифрования, необходимый для чтения и записи из бэкенда хранения, еще не известен. Для распечатывания нужно предоставить корневой ключ, чтобы получить ключ шифрования. Риск передачи корневого ключа заключается в том, что один злоумышленник, имеющий к нему доступ, может расшифровать весь StarVault. Вместо этого метод Шамира позволяет разделить корневой ключ на несколько частей. Количество частей и необходимое пороговое значение настраивается, но по умолчанию StarVault генерирует 5 частей, и для воссоздания корневого ключа достаточно предоставить любые 3 из них.

Метод разделения секрета позволяет избежать ситуации, когда держатель корневого ключа пользуется абсолютным доверием, и позволяет вообще не хранить корневой ключ. Корневой ключ можно получить только путем воссоздания его из частей. Сами части используются только для распечатывания, для выполнения каких-либо запросов к StarVault они бесполезны. После распечатывания для всех запросов используются стандартные механизмы ACL.

Если провести аналогию, банк помещает сейфовые ячейки в хранилище. У каждой ячейки есть ключ, а дверь хранилища закрыта и ключом, и комбинацией. Хранилище сделано из стали и бетона, так что попасть внутрь можно только через дверь. Аналогия с StarVault заключается в том, что криптосистема — конструкция из стали и бетона, защищающая данные. Конечно можно пробурить туннель в бетоне или подобрать ключи шифрования, только это займет слишком много времени.

Для открытия банковского хранилища требуются два фактора: ключ и комбинация. Аналогично, StarVault требует предоставления нескольких частей для воссоздания корневого ключа. Даже после вскрытия хранилища открыть каждую сейфовую ячейку можно лишь ключом, который предоставляет владелец, и аналогичным образом система StarVault ACL защищает все хранящиеся секреты.

Методы аутентификации. Общие сведения

Методы аутентификации — это компоненты в StarVault, которые выполняют аутентификацию и отвечают за присвоение идентификатора и набора политик пользователю. Во всех случаях StarVault обеспечивает аутентификацию как часть обработки запроса. В большинстве случаев StarVault делегирует администрирование и принятие решений о аутентификации соответствующему настроенному внешнему методу аутентификации (например, Kubernetes, LDAP и т.д.).

Наличие нескольких методов аутентификации позволяет выбрать наиболее подходящий для каждого отдельного случая использования StarVault.

Например, для серверов рекомендуемым выбором является метод **AppRole**.

1. Жизненный цикл методов аутентификации

Большинство методов аутентификации можно включать, отключать, настраивать и перемещать с помощью CLI, API или UI.

К этапам жизненного цикла методов аутентификации относятся:

- **Enable** - активация метода аутентификации по указанному пути. За некоторыми исключениями, методы могут быть включены по нескольким путям. По умолчанию методы аутентификации монтируются в **auth/<type>**. Например, после активации метода "ldap", взаимодействовать с ним можно по адресу **auth/ldap**.
- **Disable** - отключение существующего метода аутентификации. При отключении метода аутентификации, все пользователи, прошедшие аутентификацию с помощью этого метода, автоматически выходят из системы.
- **Move** - перемещение существующего метода аутентификации в другой путь. В результате этого процесса все аренды аннулируются. Конфигурационные данные, хранящиеся для метода аутентификации, сохраняются после перемещения.
- **Tune** - настройка глобальной конфигурации метода аутентификации, например TTL.

После активации метода аутентификации 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.

1.1. Управление жизненным циклом

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

Операция	Выполнение в UI	Выполнение в CLI	Дополнительная информация
Просмотр списка доступных методов	На странице **Access**	`starvault auth list`	При использовании CLI возможно управление выводом списка методов с помощью следующих опций: • `-detailed` - включает подробный вывод параметров методов, таких как TTL, параметры репликации, версия плагина и т.д. • `-format=<string>` - указывает способ представления вывода. Допустимые варианты: `table` (по умолчанию), `json` и `yaml`.
Активация	На странице **Access** путем нажатия **[Enable new method]**	`starvault auth enable [options] <auth-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 методов аутентификации.
Деактивация	На странице **Access** в расширенном меню метода (⋮) по кнопке **[Disable]**	`starvault auth disable <path>`	
Перемещение	Недоступно	`starvault auth move <old-path> <new-path>`	

Операция	Выполнение в UI	Выполнение в CLI	Дополнительная информация
Настройка	***Вариант 1*** На странице **Access** в расширенном меню метода ( ) по кнопке **[Edit configuration]** . ***Вариант 2*** На вкладке **Configuration** в подробном представлении метода по кнопке **[Configure]**.	`starvault auth tune <options> <path>`	Список параметров и их описание см. в разделе Общие параметры для методов аутентификации.

1.2. Общие параметры для методов аутентификации

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

Для конкретных методов могут быть доступны не все параметры. Подробнее см. в описании соответствующего метода.

Опция команды CLI	Поле в UI	Описание параметра
`-allowed-response-headers=<string>`		Используется для настройки списка заголовков ответа, которые разрешено возвращать клиенту при запросах к методу аутентификации. Например: `starvault auth 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 auth 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 auth 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 методом аутентификации, к которому применяется эта настройка. Это время, на которое выдается токен или другие учетные данные после успешной аутентификации пользователя. Как только TTL истекает, аренда (и соответствующие учетные данные) становятся недействительными, если они не были продлены. Пример: `starvault auth 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 метода аутентификации.
`-external-entropy-access` Опцию можно активировать только при включении метода.		Используется для включения доступа к внешнему источнику энтропии при генерации криптографических ключей, токенов или других случайных значений, которые требуют высокого уровня случайности. Пример: `starvault auth enable --external-entropy-access=true <method-name>` Эта команда включает доступ к внешнему источнику энтропии для метода <method-name> при его активации.

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 метода. Эта настройка влияет на то, какие данные пользователи могут видеть, когда они запрашивают список доступных ключей или путей. 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 auth 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 метода.	Опция **Local**	Используется для указания, что данный метод аутентификации должен быть локальным для сервера, на котором он настроен. Это означает, что метод не будет реплицироваться в кластерных установках StarVault, которые используют репликацию данных. Пример: `starvault auth enable -local <method-name>`

Опция команды CLI	Поле в UI	Описание параметра
`-max-lease-ttl=<duration>`	**Max Lease TTL**	Устанавливает максимальное время жизни (TTL) для аренды, выдаваемой методом аутентификации. Это ограничение применяется ко всем токенам и учетным данным, выданным через точку монтирования, и задаёт верхний предел времени, на который можно продлить аренду до её истечения. Пример: `starvault auth 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 auth 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 метода.	**Path**	Определяет путь точки монтирования, где будет активирован метод аутентификации. Например: `starvault auth enable -path=custom/path userpass` В этом примере метод типа `userpass` активируется по пользовательскому пути `custom/path` .

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

Опция команды CLI	Поле в UI	Описание параметра
`-seal-wrap` Опцию можно активировать только при включении метода.	Опция **Seal wrap**	Используется для включения функции обертывания печати (Seal Wrapping) для всего метода аутентификации или определенных данных 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 методом аутентификации, 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 учетные данные или токены без распечатывания (unsealing) хранилища. Пример: `starvault auth enable -seal-wrap -path=my-auth userpass` В этом примере для метода аутентификации типа `userpass`, активированного по пути `my-auth`, включена функция **Seal Wrapping**, обеспечивающая дополнительный уровень защиты для данных, хранящихся в этом методе.

Опция команды CLI	Поле в UI	Описание параметра
`-token-type=<string>`	Меню **Token Type**	Используется для определения типа токена, который будет выдан после успешной аутентификации пользователя. Тип токена влияет на возможности и поведение токена в системе StarVault. Существует несколько типов токенов, которые можно указать с помощью этой опции: • **service** - стандартный тип токена в StarVault. Токены типа "service" могут иметь связанные с ними дочерние токены и политики. Они могут быть использованы для длительных сессий и обычно используются для взаимодействия с StarVault. • **batch** - являются легковесными и имеют ограниченный набор возможностей по сравнению с токенами типа "service". Они не хранятся в постоянном хранилище StarVault и не имеют возможности создавать дочерние токены. Они идеально подходят для краткосрочных операций, которые требуют меньшей степени управления и аудита. • **default** - тип токена будет определяться глобальными настройками StarVault или параметрами по умолчанию для конкретного метода аутентификации.

Опция команды CLI	Поле в UI	Описание параметра
`-user-lockout-counter-reset-duration=<duration>`		Определяет период времени, по истечении которого счетчик неудачных попыток входа (автоматической блокировки пользователя) будет сброшен. Эта настройка используется в сочетании с механизмами блокировки учетных записей для предотвращения подбора паролей и других видов атак. Пример: `starvault auth tune -user-lockout-counter-reset-duration=1h path/to/auth/method` В этом примере для метода аутентификации, находящегося по пути **path/to/auth/method**, устанавливается продолжительность сброса счетчика блокировки равной одному часу (`1h`). Если пользователь вводит неверные учетные данные, счетчик неудачных попыток будет сброшен, если он не пытается войти в течение одного часа после последней неудачной попытки.
`-user-lockout-disable`		Используется для отключения механизма блокировки пользователей после определенного количества неудачных попыток аутентификации. Это может быть полезно в средах, где блокировка пользователей не требуется или где администраторы предпочитают другие методы защиты от атак по подбору паролей. Пример: `starvault auth tune -user-lockout-disable=true path/to/auth/method`

Опция команды CLI	Поле в UI	Описание параметра
`-user-lockout-duration=<duration>`		Задаёт продолжительность блокировки пользователя после превышения допустимого количества неудачных попыток входа. Пример: `starvault auth tune -user-lockout-duration=30m path/to/auth/method` В этом примере для метода аутентификации по пути **path/to/auth/method** устанавливается продолжительность блокировки пользователя равной 30 минутам (`30m`). Это означает, что после превышения допустимого числа неудачных попыток входа, пользователь будет заблокирован на 30 минут.
`-user-lockout-threshold=<uint>`		Устанавливает пороговое значение неудачных попыток входа, после которого пользователь будет временно заблокирован. Это мера безопасности предназначена для предотвращения атак по подбору паролей и других видов несанкционированного доступа. Пример: `starvault auth tune -user-lockout-threshold=5 path/to/auth/method` В этом примере для метода аутентификации по пути **path/to/auth/method** устанавливается пороговое значение в 5 неудачных попыток входа. Если пользователь вводит неверные учетные данные более пяти раз подряд, он будет заблокирован в соответствии с настройками продолжительности блокировки.