Устройство аудита Syslog

Устройство аудита `syslog` записывает логи в syslog.

В настоящее время `syslog` не поддерживает настраиваемое место назначения и всегда отправляет логи локальному агенту. Это устройство поддерживается только в системах Unix. `syslog` не следует включать, если резервные экземпляры StarVault его не поддерживают.

Сообщения аудита, генерируемые для некоторых операций, могут быть довольно большими и превышать максимальный размер одного UDP-пакета. Если это возможно с демоном `syslog`, настройте TCP-приемник. В противном случае рассмотрите возможность использования файлового бэкэнда и настройки `syslog` на чтение записей из файла; или включите и файловый, и `syslog`, чтобы сбой при регистрации определенного сообщения непосредственно в `syslog` не привел к блокировке StarVault.

1. Примеры

Устройство аудита `syslog` можно включить следующей командой:

```bash
starvault audit enable syslog
```

Передать параметры конфигурации можно с помощью пар **K=V** (ключ=значение):

```bash
starvault audit enable syslog tag="starvault" facility="AUTH"
```

2. Конфигурация

Устройство аудита сокетов поддерживает общие параметры конфигурации, описанные на странице "Аудит"
Специфические параметры для устройства аудита сокет:

- `facility` (`string: «AUTH»`) — объект `syslog`, который необходимо использовать.
- `tag` (`string: «starvault»`) — тег `syslog`, который необходимо использовать.

Методы аутентификации. AppRole

Метод аутентификации **AppRole** позволяет машинам или приложениям аутентифицироваться с использованием ролей, определенных в StarVault. Открытая конструкция AppRole позволяет использовать различные рабочие процессы и конфигурации для управления большим количеством приложений. Этот метод аутентификации ориентирован на автоматизированные рабочие процессы (машинное оборудование и сервисы) и менее полезен для операторов-людей. Мы рекомендуем с методом аутентификации AppRole использовать пакетные токены.

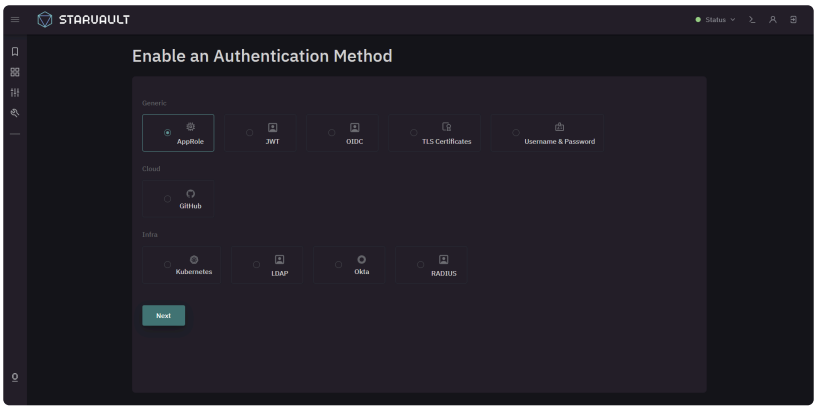
AppRole представляет собой набор политик StarVault и ограничений входа, которые должны быть выполнены для получения токена с этими политиками. Область применения может быть столь узкой или широкой, насколько это необходимо. AppRole может быть создан для конкретной машины, или даже для конкретного пользователя на этой машине, или для сервиса, распределенного по машинам. Требования к учетным данным для успешного входа зависят от ограничений, установленных на AppRole, связанном с учетными данными.

1. Настройка AppRole

Методы аутентификации должны быть настроены заранее, прежде чем пользователи или машины смогут пройти аутентификацию. Эти шаги обычно выполняются администратором или инструментом управления конфигурацией.

Для настройки AppRole выполните следующие действия:

1. Активируйте метод:
    - Через UI:
        a. Аутентифицируйтесь в пользовательском интерфейсе с достаточными правами.
        b. На странице **Access** нажмите **[Enable new method]**.
        c. Выберите AppRole и нажмите **[Next]**.



d. При необходимости измените путь в поле **Path** и параметры метода в группе **Hide Method Options**. Подробнее о параметрах см. в разделе Общие параметры для методов аутентификации.

e. Нажмите **[Enable Method]**.

- Через CLI:

a. Используйте команду для активации метода аутентификации.

```
starvault auth enable [options] approle ①
```

① Подробнее о параметрах см. в разделе Общие параметры для методов аутентификации

2. Создайте именованную роль с необходимыми параметрами:

```
starvault write auth/approle/role/<role-name> [options] ①
```

① Замените `<role-name>` на имя создаваемой роли

Для настройки параметров роли используйте необходимые опции. Подробнее о доступных параметрах роли см. в таблице

3. Определите RoleID для AppRole

```
starvault read auth/approle/role/demo-role/role-id

Key        Value
---        -----
role_id    6dd13b9f-024e-5ffd-f516-bb12a119a930
```

RoleID является идентификатором, который выбирает AppRole, с которым сравниваются другие учетные данные. При аутентификации через конечную точку `login` этого метода аутентификации RoleID всегда является обязательным аргументом

4. Получите SecretID, выданный для AppRole:

```
starvault write -f auth/approle/role/demo-role/secret-id

Key                   Value
---                   -----
secret_id             82b1c63c-368a-4985-48d1-0f2784bbd200 ①
secret_id_accessor    8a9b733b-c591-3bee-ded6-c1aa458dd602
secret_id_num_uses    40
secret_id_ttl         10m
```

① Необходимое значение

SecretID - это учетные данные, которые по умолчанию требуются для любого входа (через secret_id). Для продвинутого использования требование SecretID может быть отключено через параметр bind_secret_id роли AppRole, позволяя машинам с знанием только RoleID или соответствующим другим установленным ограничениям получать токен.

2. Управление ролями AppRole

В таблице ниже представлены команды для управления ролями в методе **AppRole**.

Команда	**Описание**
`starvault write auth/<approle:path>/role/<role-name> <key=value list>`	Создание новой роли или изменение существующей по пути `auth/<approle:path>/role/<role-name>`. Параметры роли передаются в формате `key=value`. Для добавления нескольких ключей со значениями, их необходимо перечислить через пробел. Подробнее о доступных параметрах роли см. в таблице
`starvault read [-format=<string>] auth/<approle:path>/role/<role-name>`	Возвращает параметры роли с именем <role-name> в методе **AppRole** по пути `<approle:path>`. С помощью опции `-format=<string>` можно указать формат представления данных. Допустимые значения: `table` (по умолчанию), `yaml`, 'json', 'raw'.
`starvault list [-format=<string>] auth/<approle:path>/role`	Возвращает список ролей в методе **AppRole** по пути `<approle:path>`. С помощью опции `-format=<string>` можно указать формат представления данных. Допустимые значения: `table` (по умолчанию), `yaml`, 'json'.
`starvault delete auth/<approle:path>/role/<role-name>`	Удаляет роль с именем <role-name> в методе **AppRole** по пути `<approle:path>`.

Параметры, которые можно передать при создании/изменении роли представлены в следующей таблице:

Опция	**Описание**
`bind_secret_id`	Логическое значение. Определяет, будет ли для аутентификации через данную роль требоваться Secret ID.
`secret_id_bound_cidrs`	Список значений, разделенных запятыми. Определяет список CIDR-адресов, с которых разрешено использовать Secret ID для аутентификации.

Опция	Описание
`secret_id_num_uses`	Целочисленное значение. Количество раз, которое Secret ID может быть использовано для аутентификации.
`secret_id_ttl`	Строковое значение. Время жизни (TTL) для Secret ID, связанного с ролью.
`local_secret_ids`	Логическое значение. Определяет, будут ли Secret ID, связанные с данной ролью, локальными для сервера StarVault, на котором они были созданы.
`token_ttl`	Целочисленное или строковое значение. Определяет время жизни (TTL) токенов, выдаваемых для данной роли AppRole. По истечении этого времени токен будет автоматически отозван, если только он не будет продлен.
`token_max_ttl`	Целочисленное или строковое значение. Максимальное время жизни сгенерированных токенов. Текущее значение этого параметра будет указано при обновлении.
`token_policies`	Список значений, разделенных запятыми. Используется для указания списка политик, которые будут присвоены токенам, выданным при аутентификации с использованием этой роли.
`token_bound_cidrs`	Список значений, разделенных запятыми. Определяет список CIDR-адресов, для которых будет действителен выданный токен. Это ограничение по IP-адресам устанавливает, что токен может использоваться только с определенных IP-адресов или диапазонов IP-адресов, что повышает безопасность, ограничивая возможность использования токена
`token_explicit_max_ttl`	Целочисленное или строковое значение. Задает максимальное время жизни (TTL) для токенов, выданных с использованием этой роли. Это значение устанавливает жесткий верхний предел времени жизни токена, который не может быть превышен даже при продлении токена.
`token_no_default_policy`	Логическое значение. Определяет, будет ли новому токену автоматически присваиваться политика по умолчанию. Если `token_no_default_policy` установлен в `true`, то при создании токена политика по умолчанию не будет присвоена.
`token_num_uses`	Целочисленное значение. Максимальное количество раз, которое может быть использован сгенерированный токен (в течение срока его действия). 0 означает неограниченное количество раз. Если требуется, чтобы токен имел возможность создавать дочерние токены, необходимо установить это значение равным 0.
`token_period`	Целочисленное или строковое значение. Определяет период, в течение которого выданный токен может быть продлен бесконечное количество раз, пока политики, связанные с токеном, остаются неизменными.

Опция	Описание
`token_type`	Строковое значение. Определяет тип токена, который будет выдан при аутентификации с использованием этой роли. Допустимые значения: • `service` - стандартный тип токена, который может быть продлен в соответствии с его политиками и TTL. Токены типа "service" подходят для большинства приложений и сервисов, которые требуют долгосрочного доступа и возможности продления токена. • `batch` - легковесный тип токена, который хранится в памяти и не записывается в хранилище данных StarVault. Токены типа "batch" не могут быть продлены или отозваны индивидуально, и они исчезают при перезагрузке или выключении сервера StarVault. Они подходят для краткосрочных операций и сценариев с большим объемом токенов, где требуется меньшая нагрузка на хранилище. • `default` - Если параметр не задан, будет использоваться тип токена по умолчанию, который в большинстве случаев является типом "service".

Пример создания роли:

```
starvault write auth/approle/role/demo-role \
    secret_id_ttl=10m \
    token_num_uses=10 \
    token_ttl=20m \
    token_max_ttl=30m \
    secret_id_num_uses=40
```

Методы аутентификации. TLS сертификаты

Данный механизм может использовать внешние сертификаты X.509 в качестве части проверки TLS или подписания. Проверка подписей по сертификатам X.509, использующим SHA-1, является устаревшей и больше не используется без обходного пути. Дополнительную информацию см. в разделе FAQ об устаревании.

Метод аутентификации `cert` позволяет выполнять аутентификацию с помощью клиентских сертификатов SSL/TLS, подписанных либо центром сертификации, либо самоподписанных. Сертификаты клиентов SSL/TLS должны содержать расширение `ExtKeyUsage` с параметром использования `ClientAuth` или `Any`.

Доверенные сертификаты и центры сертификации настраиваются непосредственнов методе аутентификации, задавая путь `certs/`. Этот метод не может считывать доверенные сертификаты из внешнего источника.

Сертификаты CA ассоциируются с ролью; имена ролей и имена CRL нормализованы к нижнему регистру.

Обратите внимание, что для использования этого метода авторизации значения `tls_disable` и `tls_disable_client_certs` в конфигурации StarVault должны быть равны `false`. Это связано с тем, что сертификаты отправляются через TLS-соединение.

1. Проверка отзыва

Метод поддерживает проверку отзыва.

Авторизованный пользователь может отправить CRL в формате PEM, идентифицированные по заданному имени; их можно обновлять или удалять по своему усмотрению. Также авторизованный пользователь может задать URL-адрес доверенной точки распространения CRL и попросить StarVault получить CRL по мере необходимости.

Когда CRL присутствуют, во время аутентификации клиента:

- Если клиент представляет любую цепочку, в которой ни один сертификат не соответствует отозванному серийному номеру, аутентификация разрешена.
- Если клиент не представил ни одной цепочки без отозванного серийного номера, аутентификация запрещена.

Этот метод обеспечивает хорошую безопасность и в то же время допускает гибкость. Например, если промежуточный ЦС (центр сертификации) будет отозван, клиент может быть настроен с двумя цепочками сертификатов: одна содержит начальный промежуточный ЦС в пути, а другая — замену. Если первоначальный промежуточный центр сертификации будет отозван, цепочка, содержащая замену, позволит клиенту успешно пройти аутентификацию.

N.B.: Сопоставление выполняется только по серийному номеру. Для большинства ЦС, включая метод `pki` в StarVault, можно успешно использовать несколько CRL, поскольку серийные номера глобально уникальны. Однако, поскольку в RFC указано, что серийные номера должны быть уникальными только для каждого CA, некоторые CA выдают серийные номера в порядке очереди, что может привести к конфликтам при попытке использовать CRL от двух таких CA в одном монтировании метода. Обходным решением в этом случае является монтирование нескольких копий метода `cert`, настройка каждой из них с одним CA/CRL и подключение клиентов к соответствующему монтированию.

Кроме того, если точка распространения CRL не задана, то метод не будет самостоятельно получать CRL, и время, указанное в CRL для следующего обновления, не учитывается. Если CRL больше не используется, администратор должен удалить его из метода.

В дополнение к автоматическому или ручному управлению CRL, для настроенного сертификата может быть включен OCSP, в этом случае StarVault будет запрашивать OCSP-сервер, указанный в представленном сертификате или настроенный в методе auth, для проверки отзыва.

2. Аутентификация

2.1. С помощью CLI

Ниже приведена аутентификация по роли `web` cert путем предоставления сертификата (`cert.pem`) и ключа (`key.pem`), подписанных ЦС, связанным с ролью `web` cert. Обратите внимание, что имя web связано с приведенным ниже примером конфигурации, записывающим путь `auth/cert/certs/web`. Если имя роли сертификата не указано, метод auth будет пытаться аутентифицироваться по всем доверенным сертификатам.

Значение `-ca-cert` здесь используется для сертификата ЦС Слушателя StarVault TLS, а не для ЦС, выпустившего сертификат аутентификации клиента. Это значение можно опустить, если ЦС, используемый для выдачи сертификата сервера StarVault, является доверенным для локальной системы, выполняющей эту команду.

BASH

```
starvault login \
    -method=cert \
```

```
    -ca-cert=starvault-ca.pem \
    -client-cert=cert.pem \
    -client-key=key.pem \
    name=web
```

2.2. С помощью API

Конечной точкой для входа в систему является `/login`. Клиент подключается с помощью своего сертификата TLS, и когда конечная точка login будет достигнута, метод auth определит, есть ли подходящий доверенный сертификат для аутентификации клиента. В качестве опции можно указать роль одного сертификата для аутентификации.

Значение `--cacert` здесь используется для сертификата ЦС Слушателя StarVault TLS, а не для ЦС, выпустившего сертификат аутентификации клиента. Это значение можно опустить, если ЦС, используемый для выдачи сертификата сервера StarVault, является доверенным для локальной системы, выполняющей эту команду.

BASH

```
curl \
    --request POST \
    --cacert starvault-ca.pem \
    --cert cert.pem \
    --key key.pem \
    --data '{"name": "web"}' \
    https://127.0.0.1:8200/v1/auth/cert/login
```

3. Конфигурация

Методы аутентификации должны быть настроены заранее, прежде чем пользователи или машины смогут пройти аутентификацию. Эти шаги обычно выполняются оператором или средствами управления конфигурацией.

1. Включите метод аутентификации с помощью сертификата:

   BASH

   ```
   starvault auth enable cert
   ```

2. Настройте метод с помощью доверенных сертификатов, которым разрешена аутентификация:

   BASH

   ```
   starvault write auth/cert/certs/web \
       display_name=web \
       policies=web,prod \
       certificate=@web-cert.pem \
       ttl=3600
   ```

При этом создается новый доверенный сертификат "web" с тем же именем отображения и политиками "web" и "prod". Сертификат (открытый ключ), используемый для проверки клиентов, задается файлом "web-cert.pem". Наконец, можно указать необязательное значение `ttl` в секундах, чтобы ограничить продолжительность аренды.

4. API

Метод аутентификации с помощью сертификата TLS имеет полноценный HTTP API. Более подробная информация приведена в разделе API сертификата TLS.