

Интеграция PostgresPro Enterprise TDE с Deckhouse Stronghold

Содержание

Интеграция PostgresPro Enterprise TDE с Deckhouse Stronghold	1
Содержание.....	1
1 ВВЕДЕНИЕ	2
Принцип работы.....	2
Схема взаимодействия	3
Аутентификация в Stronghold	3
Безопасность.....	3
2 ОПИСАНИЕ ТЕСТОВОЙ СРЕДЫ	4
3 УСТАНОВКА И НАСТРОЙКА Standalone STRONGHOLD	5
4 УСТАНОВКА И НАСТРОЙКА STRONGHOLD в среде Kubernetes	5
5 ПОДГОТОВКА POSTGRES PRO ENTERPRISE	5
6 ИНТЕГРАЦИЯ ЧЕРЕЗ STRONGHOLD AGENT	6
Схема взаимодействия с участием Agent.....	7
6.1 Настройка AppRole в Stronghold	8
6.2 Установка и настройка Stronghold Agent на сервере PostgresPro	8
Создание пользователя и директорий.....	8
Создание файла с переменными окружения	9
Сохранение Role ID и Secret ID	9
Создание конфигурационного файла Agent.....	9
6.3 Настройка systemd-сервиса Stronghold Agent	9
6.4 Добавление зависимости в сервис PostgresPro	9
6.5 Проверка работы команд с токеном из файла	9
6.6 Настройка параметров кодирования СУБД	10
6.7 Перезапуск PostgresPro	10
7 СОЗДАНИЕ ЗАЩИЩЕННОГО ТАБЛИЧНОГО ПРОСТРАНСТВА	10
8 РОТАЦИЯ КЛЮЧЕЙ	10
8.1 Ротация ключа табличного пространства	10
8.2 Обновление мастер-ключа в Stronghold Transit	10
ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ	11

1 ВВЕДЕНИЕ

СУБД Postgres Pro Enterprise, начиная с 17-й версии, поддерживает механизм **прозрачного защитного преобразование данных** (Transparent Data Encoding, или **TDE**), который позволяет обезличивать конфиденциальную или персональную информацию в выделенных табличных пространствах (tablespaces) и в журнале предзаписи (WAL). При активации TDE данные кодируются при записи на диск или в систему резервного копирования и раскодируются при чтении. Преимуществом этого механизма является то, что он не требует никаких изменений в приложении или на клиентах.

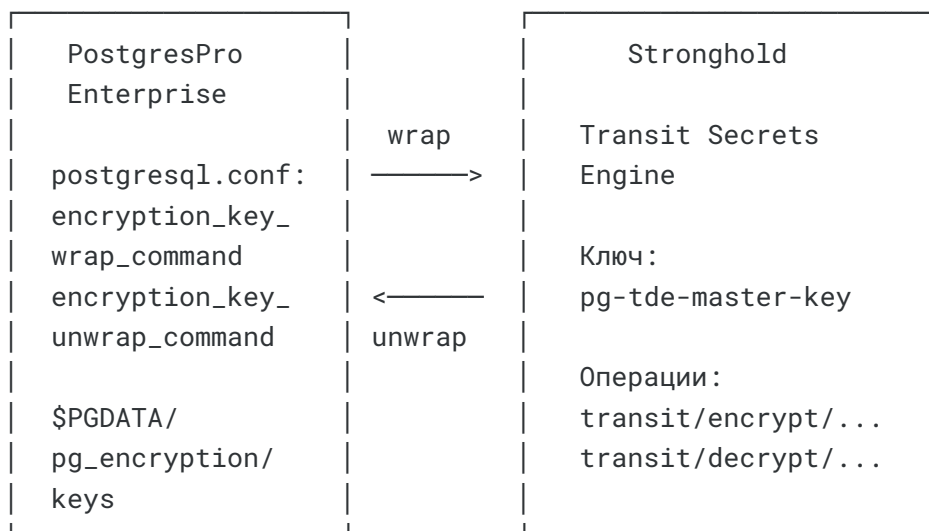
Для защиты используемых при этом ключей преобразования PostgresPro Enterprise может использовать внешнюю **систему управления ключами (KMS)**. В качестве такой системы в данном руководстве рассматривается **Deckhouse Stronghold** нашего стратегического партнера – компании «Флант». Эта KMS обеспечивает безопасное управление жизненным циклом секретов, имеет русскоязычный интерфейс, работает на российских операционных системах и полностью совместим с API HashiCorp Vault.

Совместное решение обеспечивает дополнительную защиту ключей, стандартизацию работы с секретами в организации, возможность использования только отечественного ПО, включая редакции, сертифицированные ФСТЭК РФ.

Принцип работы

Ключи защитного преобразования табличных пространств и WAL находятся в специальном файле в каталоге СУБД `$PGDATA/pg_encryption/keys`. Это хранилище ключей защищено и преобразовано с помощью **мастер-ключа**, который генерируется и хранится в Stronghold Transit Secrets Engine. При рестарте и запуске СУБД PostgresPro вызывает внешнюю команду (`encryption_key_unwrap_command`) для раскодирования файла ключей с помощью Stronghold, а при создании новых ключей – команду кодирования (`encryption_key_wrap_command`). Ключи в открытом виде помещаются в память СУБД и используются для работы с данными в выделенных табличных пространствах и в журнале предзаписи.

Схема взаимодействия



Аутентификация в Stronghold

Взаимодействие между PostgresPro и Stronghold происходит по протоколу HTTPS, но перед началом работы сервис СУБД должен аутентифицироваться в KMS.

Аутентификация выполняется через **Stronghold Agent** с настроенным методом аутентификации **AppRole**. Stronghold Agent получает токен через AppRole, записывает в файл, доступный на чтение пользователю `postgres` (через переменную окружения `VAULT_TOKEN_FILE`). Токен автоматически обновляется при истечении времени жизни (TTL).

Безопасность

Связка TDE и Stronghold Agent обеспечивает разделение доступа к ключам и снижает риски при компрометации отдельных компонентов.

Доступ сервисов к ключам и учётным данным:

Компонент	Мастер-ключ (Stronghold Transit)	Role ID / Secret ID	Токен (коротко-живущий)	Файл ключей <code>pg_encryption/keys</code>	Расшифрованные ключи табличных пространств
Stronghold	Хранит, ключ не покидает хранилище	Нет доступа	Выдаёт по запросу	Нет доступа	Нет доступа
Stronghold Agent	Нет доступа	Читает (для аутентификации)	Получает и записывает в файл	Нет доступа	Нет доступа
PostgresPro (процесс <code>postgres</code>)	Нет доступа	Нет доступа	Читает из файла	Читает (зашифрованный файл)	В памяти только при работе (результат <code>unwrap</code>)

Ключевые принципы:

- **Мастер-ключ никогда не покидает Stronghold.** Stronghold выполняет только операции шифрования и дешифрования (transit/encrypt, transit/decrypt). Исходное значение ключа не передаётся ни PostgresPro, ни Agent.
- **PostgresPro не хранит долгоживущих секретов.** У процесса есть доступ только к короткоживущему токenu (TTL 1–4 ч), обновляемому Agent. Role ID и Secret ID лежат в файлах, доступных только root и stronghold-agent; пользователь `postgres` их не читает.
- **Политика доступа токена ограничена.** Токен, выдаваемый для AppRole `pg-tde-role`, даёт права только на `transit/encrypt/pg-tde-master-key` и `transit/decrypt/pg-tde-master-key`. Доступ к другим секретам, ключам или админ-функциям Stronghold отсутствует.

Минимизация угроз ИБ:

- **Кража или копирование диска СУБД** — данные в защищенных табличных пространствах хранятся в закодированном виде. Файл `pg_encryption/keys` тоже зашифрован мастер-ключом; без доступа к Stronghold расшифровать данные нельзя.
- **Утечка файла ключей** — похищение только файла `pg_encryption/keys` бесполезно: для расшифровки нужен мастер-ключ, который хранится исключительно в Stronghold.
- **Компрометация сервера PostgresPro** — злоумышленник получает короткоживущий токен и закодированные данные. После истечения срока действия токена получить новый токен без Role ID и Secret ID (которые читает только Agent) нельзя; к тому же для расшифровки данных нужна работающая связь со Stronghold в момент `unwrap`.
- **Компрометация токена** — при утечке токена злоумышленник может вызывать только `encrypt/decrypt` для одного ключа в ограниченное время, без доступа к другим секретам и без извлечения самого мастер-ключа.

Таким образом **разделение ролей** (Stronghold хранит мастер-ключ, Agent выдаёт ограниченный токен, PostgresPro использует только этот токен и зашифрованный файл ключей) и **принцип наименьших привилегий** делают использование TDE с Stronghold Agent безопасным с точки зрения хранения ключей и доступа к ним.

2 ОПИСАНИЕ ТЕСТОВОЙ СРЕДЫ

В рамках программы взаимной сертификации технологических решений интеграционные испытания проводились для двух сценариев развертывания Stronghold:

- Standalone-решение на ОС Astra Linux 1.7.7 / Ubuntu 22.04 / RHEL 9
- контейнеризированное приложение в Deckhouse Kubernetes Platform версии 1.71

Сервер PostgresPro Enterprise версии 17.7.1 был размещен в той же сети, и на нем был установлен [stronghold CLI](#)

Точные параметры настройки могут получить заказчики PostgresPro и Deckhouse Stronghold, обратившись за протоколом испытаний в службы технической поддержки организаций.

3 УСТАНОВКА И НАСТРОЙКА Standalone STRONGHOLD

В данном разделе представлены выдержки из [протокола](#) с этапами установки Stronghold на выделенном сервере (VM/bare-metal).

- Установка пакетов Stronghold
- Настройка systemd-сервиса
- Подготовка TLS-сертификатов
- Создание конфигурационного файла и запуск сервиса
- Инициализация сервиса, Перевод системы в рабочее состояние, Авторизация с root-токеном
- Включение Transit Secrets Engine и создание мастер-ключа для PostgresPro TDE
- Создание политики для PostgresPro TDE

4 УСТАНОВКА И НАСТРОЙКА STRONGHOLD в среде Kubernetes

В данном разделе представлены выдержки из [протокола](#) с этапами настройки Stronghold в составе Deckhouse Kubernetes Platform (DKP):

- Включение модуля Stronghold в DKP
- Проверка статуса Stronghold
- Предоставление доступа к Stronghold извне кластера
- Получение root-токена
- Включение Transit и создание мастер-ключа для PostgresPro TDE
- Создание политики для PostgresPro TDE

5 ПОДГОТОВКА POSTGRESPRO ENTERPRISE

В данном разделе представлены выдержки из [протокола](#) с этапами настройки сервера PostgresPro:

- Установка пакета pgpro_tde
- Настройка shared_preload_libraries
- Установка Stronghold CLI
- Копирование CA-сертификата Stronghold
- Проверка сетевой доступности Stronghold

6 ИНТЕГРАЦИЯ ЧЕРЕЗ STRONGHOLD AGENT

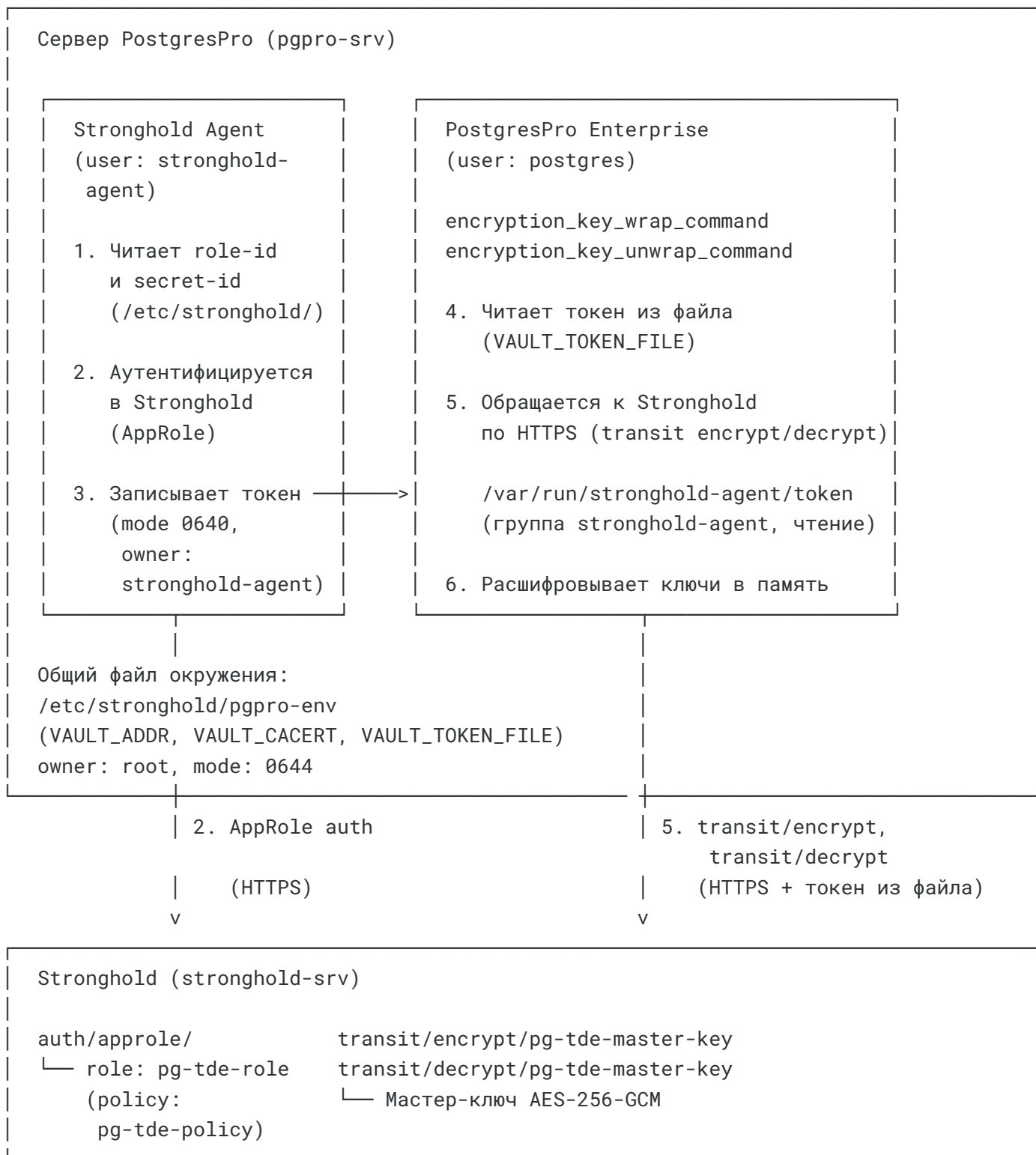
В данном разделе представлены выдержки из [протокола](#) с этапами установки и настройки Stronghold Agent на сервере PostgresPro, а также – необходимые для этого действия на сервере Stronghold.

Stronghold Agent на сервере PostgresPro выполняет следующие функции:

- **Автоматическая аутентификация** через AppRole (без хранения долгоживущих токенов на диске);
- **Автоматическое обновление** токена при приближении к истечению его срока жизни;
- **Запись токена в файл** – Agent помещает текущий токен в файл (например, `/var/run/stronghold-agent/token`) с правами, позволяющими пользователю `postgres` читать его. PostgresPro при вызове `wrap/unwrap` обращается **напрямую к Stronghold** (по HTTPS), используя путь к файлу с токеном через переменную `VAULT_TOKEN_FILE`. Локальный API без авторизации не используется.

Подробнее о Stronghold Agent можно узнать [по ссылке](#).

Схема взаимодействия с участием Agent



Для безопасного обмена секретами в файловой системе сервера PostgresPro должны быть установлены привилегии, аналогичные приведенным в таблице значениям.

Владение файлами и права доступа:

Файл / каталог	Владелец	Группа	Права	Кто читает	Кто пишет
/etc/stronghold/role-id	root	stronghold-agent	0640	Agent	Администратор (root)
/etc/stronghold/secret-id	root	stronghold-agent	0640	Agent	Администратор (root)
/etc/stronghold/pgpro-env	root	root	0644	Agent, PostgresPro	Администратор (root)
/var/run/stronghold-agent/token	stronghold-agent	stronghold-agent	0640	Agent, PostgresPro (группа stronghold-agent)	Agent
/etc/stronghold/agent.hcl	root	stronghold-agent	0640	Agent	Администратор (root)

6.1 Настройка AppRole в Stronghold

Как уже было описано ранее, перед началом работы сервис СУБД должен аутентифицироваться в KMS с настроенным методом аутентификации **AppRole**. Это требует на сервере Stronghold (или через `kubectl exec`):

- Включение метода аутентификации AppRole
- Создание роли для PostgresPro TDE
- Получение Role ID и Secret ID (одноразового wrapping-токена), которые будут использованы на сервере PostgresPro

6.2 Установка и настройка Stronghold Agent на сервере PostgresPro

Создание пользователя и директорий

Пользователь `postgres` должен иметь возможность читать файл с токеном, который создаёт Agent. Для этого добавьте `postgres` в группу `stronghold-agent` и перезапустите PostgresPro.

Создание файла с переменными окружения

Чтобы не дублировать настройки, адрес сервера Stronghold и путь к СА-сертификату задаются один раз в файле переменных окружения. Этот файл будет использовать PostgresPro при вызове wrap/unwrap, а сервис Agent загружает его при старте.

Сохранение Role ID и Secret ID

Необходимо создать файлы с учетными данными, полученными на шаге 6.1 «Настройка AppRole в Stronghold», которые не должны содержать лишних символов (пробелы, переносы строк), и установить к ним права доступа.

Создание конфигурационного файла Agent

Создайте [agent.hcl](#), указав в нем следующие параметры:

- Метод подключения к серверу Stronghold
- Адрес на сервере PostgresPro, куда будет записываться токен
- Настройки журналирования

и установите права доступа к нему

6.3 Настройка systemd-сервиса Stronghold Agent

- Создайте файл с настройками сервиса, обеспечивающими запуск агента до запуска СУБД и перезапуск в случае необходимости
- Активируйте и запустите Agent
- Проверьте его статус
- Проверьте, что Agent аутентифицировался и записал токен в файл

6.4 Добавление зависимости в сервис PostgresPro

Чтобы PostgresPro гарантированно запускался после Stronghold Agent, создайте drop-in файл с настройками порядка сервисов

6.5 Проверка работы команд с токеном из файла

Убедитесь, что кодирование текста и его декодирование работают при обращении к Stronghold с токеном из файла, используя те же команды, что будут в [encryption_key_wrap_command](#) / [encryption_key_unwrap_command](#)

6.6 Настройка параметров кодирования СУБД

Подключитесь к PostgresPro с правами суперпользователя и задайте параметры кодирования через `ALTER SYSTEM SET`. Обращение идёт к Stronghold по HTTPS, токен подставляется из файла, который обновляет Agent.

6.7 Перезапуск PostgresPro

Проверьте статус сервиса СУБД после перезапуска и убедитесь, что хранилище ключей создано и защищено

7 СОЗДАНИЕ ЗАЩИЩЕННОГО ТАБЛИЧНОГО ПРОСТРАНСТВА

После успешной настройки TDE в СУБД можно создавать защищенные табличные пространства, переносить туда существующие файлы отношений или генерировать новые.

Если защитное преобразование работает, на уровне файловой системы вы увидите случайные байты вместо читаемых данных.

8 РОТАЦИЯ КЛЮЧЕЙ

8.1 Ротация ключа табличного пространства

При ротации генерируется новый ключ для табличного пространства. Новые данные будут преобразовываться с помощью нового ключа, а для прочтения старых по-прежнему применяться старый ключ до их обновления.

8.2 Обновление мастер-ключа в Stronghold Transit

Если мастер-ключ в Stronghold скомпрометирован или истёк срок его жизни, выполните следующие действия:

- Создайте новый мастер-ключ в Stronghold Transit
- Перекодируйте файл ключей PostgresPro
- Обновите `postgresql.conf`, заменив старое имя ключа на новое
- Обновите политику Stronghold
- Перезапустите PostgresPro

Вместо создания нового именованного ключа можно выполнить ротацию версии ключа внутри Stronghold Transit. В этом случае Transit автоматически будет использовать новую версию ключа для кодирования, а старые версии — для декодирования. Изменения в `postgresql.conf` не потребуются.

ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ

В последующих разделах [протокола](#)

- представлены различные проверки (например, тест поведения СУБД при недоступности Stronghold);
- приведены рекомендации по эксплуатации (например, в областях безопасности, резервного копирования, мониторинга, обеспечения высокой доступности);
- перечислены последствия недоступности Stronghold для различных операций;
- присутствует справочная информация по наиболее часто используемым командам управления Stronghold Transit, политиками и токенами, AppRole, и настройки TDE
- имеются подсказки для устранения неполадок