Utrzymanie

Kopia, aktualizacja i częste problemy.

- Kopia zapasowa i aktualizacja Vaultwardena
- Vaultwarden: częste problemy

Kopia zapasowa i aktualizacja Vaultwardena

Poziom: operacyjny
Dotyczy: Vaultwarden 1.37.3 w Dockerze, baza SQLite na wolumenie **Czas: ~35 min**

Cała zawartość sejfu Zielonej mieści się w jednym pliku SQLite i kilku katalogach na wolumenie. Zwykłe skopiowanie tego pliku podczas pracy kontenera potrafi dać uszkodzoną kopię, więc kopię bazy robimy narzędziem `sqlite3` w trybie `.backup`, nie poleceniem `cp`. Aktualizacja to podmiana tagu obrazu, ale bez świeżej kopii sprzed niej wycofanie zmian po migracji bazy jest bardzo trudne.

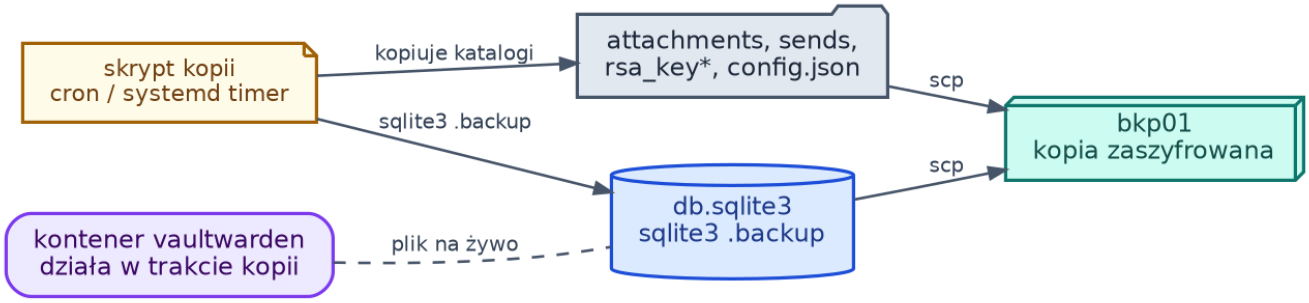
Skróty użyte na tej stronie (2)

Skrót	Co znaczy
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą
SQLite	mała baza danych zapisana w jednym pliku; Home Assistant trzyma w niej historię w pliku home-assistant_v2.db

Zanim zaczniemy

Wymaganie	Uwaga
Pakiet `sqlite3` na `docker01`	`sudo dnf install -y sqlite`, sprawdzamy `sqlite3 --version`
Dostęp do katalogu `/docker/apps/vaultwarden`	konto z prawem odczytu wolumenu kontenera na hoście
Miejsce na `bkp01` na regularne kopie	patrz Świat przykładów: Zielona po adres i rolę serwera
Znajomość obecnego tagu obrazu	`docker compose images` w `/docker/compose/vaultwarden/`

Cała droga danych, od pliku na wolumenie do serwera kopii:



Kopia bazy SQLite przez sqlite3 .backup i osobno skopiowane katalogi, wysłane na bkp01 (kliknięcie otwiera pełny rozmiar)

Uważaj: aktualizacja Vaultwardena migruje schemat bazy SQLite przy pierwszym starcie nowej wersji, podobnie jak Gitea migruje swoją bazę PostgreSQL. Migracja jest jednokierunkowa: cofnięcie tagu obrazu do starszej wersji po tym, jak migracja już przeszła, zostawia starego Vaultwardena wobec schematu bazy, którego nie rozumie. Jedyna pewna droga wycofania po migracji to odtworzenie z kopii sprzed aktualizacji.

Przebieg

1. Dlaczego nie kopiujemy pliku na żywo

> **Pojęcie: WAL.** Tryb dziennika zapisu w SQLite, w którym świeże zmiany trafiają najpierw do osobnego pliku `db.sqlite3-wal`, a do głównego pliku bazy wracają dopiero przy okazji punktu kontrolnego. Kontener Vaultwarden trzyma bazę otwartą cały czas, więc zwykłe `cp db.sqlite3 kopia.sqlite3` w dowolnym momencie może skopiować sam plik główny bez świeżych zmian z `db.sqlite3-wal`, dając kopię, która wygląda poprawnie, ale brakuje w niej najnowszych zapisów, albo w gorszym wypadku kopię niespójną w środku transakcji.

Polecenie `.backup` narzędzia `sqlite3` korzysta z tego samego mechanizmu blokad co sama baza, więc bezpiecznie czyta spójny stan danych nawet wtedy, gdy Vaultwarden akurat coś zapisuje. Nie musimy w tym celu zatrzymywać kontenera.

2. Wykonanie kopii bazy: sqlite3 .backup

```
mkdir -p /docker/apps/vaultwarden/kopie
sqlite3 /docker/apps/vaultwarden/db.sqlite3 ".backup '/docker/apps/vaultwarden/kopie/db-2026-09-28.sqlite3'"
```

Przykład sprawdzenia, że plik kopii rzeczywiście powstał i ma sensowny rozmiar:

```
ls -lh /docker/apps/vaultwarden/kopie/
```

Przykład:

```
-rw-r--r-- 1 root root 2.3M Sep 28 03:00 db-2026-09-28.sqlite3
```

3. Kopia pozostałych katalogów i plików

```
tar -czf /docker/apps/vaultwarden/kopie/dane-2026-09-28.tar.gz \
  -C /docker/apps/vaultwarden attachments sends rsa_key.pem rsa_key.pub.pem config.json
```

`attachments` i `sends` trzymają załączniki dodane do haseł i pliki wysłane funkcją Send. Pliki `rsa_key*` to klucze, którymi Vaultwarden podpisuje tokeny sesji: bez nich po odtworzeniu wszystkie istniejące sesje i część integracji trzeba by zestawić od nowa. `config.json` trzyma ustawienia zapisane z poziomu panelu `/admin`, niepokrywające się z tym, co stoi w `compose.yaml`.

4. Szyfrowanie kopii przed wysyłką

```
openssl enc -aes-256-cbc -pbkdf2 -salt \
  -in /docker/apps/vaultwarden/kopie/dane-2026-09-28.tar.gz \
  -out /docker/apps/vaultwarden/kopie/dane-2026-09-28.tar.gz.enc \
  -pass file:/root/.vaultwarden-kopia-haslo
```

Kopia sejfu haseł to sam w sobie najbardziej wrażliwy plik w firmie: zawiera zaszyfrowane hasła wszystkich osób, ale też klucze `rsa_key*` w postaci jawnej. Szyfrujemy archiwum osobnym hasłem, zanim trafi na `bkp01`, i trzymamy to hasło poza samym `bkp01`, na przykład w innym Vaultwardenie albo w sejfie fizycznym.

5. Wysyłka na bkp01

```
scp /docker/apps/vaultwarden/kopie/db-2026-09-28.sqlite3
nazwa_usera@bkp01:/backup/vaultwarden/
scp /docker/apps/vaultwarden/kopie/dane-2026-09-28.tar.gz.enc
nazwa_usera@bkp01:/backup/vaultwarden/
```

6. Harmonogram: cron albo timer systemd

Wybieramy jeden z dwóch mechanizmów opisanych osobno, cron: zadania cykliczne albo systemd timery: OnCalendar zamiast crona, i spinamy nim kroki 2 do 5 w jeden skrypt uruchamiany co noc.

```
crontab -e
```

Przykład:

```
0 3 * * * /docker/apps/vaultwarden/skrypty/kopia-nocna.sh >> /var/log/vaultwarden-kopia.log
2>&1
```

> W Zielonej nocną kopię Vaultwardena robi osobne zadanie restic do repozytorium `vaultwarden` z poziomem retencji 365 dni (decyzja autora z 11 października 2026), opisane w kroku 7 strony Kopia całego docker01: /docker/apps, /docker/compose i bazy w kontenerach, a poziomy i uzasadnienie są na stronie Strategia kopii zapasowych Zielonej: RPO, RTO, reguła 3-2-1-1-0 i dwie lokalizacje. Zadanie używa tego samego `sqlite3 .backup`, a repozytorium restic jest szyfrowane, więc osobne `openssl enc` nie jest tam potrzebne. Kroki 2 do 5 z tej strony zostają do kopii ręcznej przed aktualizacją, a cron z kroku 6 włączamy tylko tam, gdzie nie ma zadania restic, żeby dwa mechanizmy nie kopiowały tych samych danych.

7. Test odtworzenia

Raz na kwartał odtwarzamy najnowszą kopię na osobnej, testowej instalacji tej samej wersji Vaultwardena: zatrzymanie testowego kontenera, podmiana `db.sqlite3` i katalogów `attachments`/`sends`/`rsa_key*`/`config.json` na te z kopii, ponowne uruchomienie, logowanie testowym kontem. Kopia, której nikt nigdy nie spróbował odtworzyć, jest tylko przypuszczeniem, że działa.

```
docker compose -f /docker/compose/vaultwarden-test/compose.yaml down
cp /docker/apps/vaultwarden/kopie/db-2026-09-28.sqlite3 /docker/apps/vaultwarden-
test/db.sqlite3
tar -xzf /docker/apps/vaultwarden/kopie/dane-2026-09-28.tar.gz -C /docker/apps/vaultwarden-
test
docker compose -f /docker/compose/vaultwarden-test/compose.yaml up -d
```

Aktualizacja

1. Przeczytanie informacji o wydaniu

Przed zmianą tagu sprawdzamy release notes na `github.com/dani-garcia/vaultwarden/releases` pod kątem zmian wymagających ręcznej ingerencji: usuniętych zmiennych środowiskowych, zmian w formacie `ADMIN_TOKEN` albo ostrzeżeń o migracji bazy.

2. Wykonanie kopii sprzed aktualizacji

Powtarzamy kroki 2 do 5 z sekcji wyżej, poza harmonogramem, bezpośrednio przed aktualizacją: to jest kopia, do której wracamy w razie problemu, świeższa niż ostatnia nocna.

3. Zmiana tagu obrazu

`/docker/compose/vaultwarden/.env`:

```
VAULTWARDEN_TAG=1.38.0
```

`/docker/compose/vaultwarden/compose.yaml`, fragment:

```
services:
  vaultwarden:
    image: vaultwarden/server:${VAULTWARDEN_TAG}
```

```
docker compose -f /docker/compose/vaultwarden/compose.yaml pull
docker compose -f /docker/compose/vaultwarden/compose.yaml up -d
```

Przykład:

```
[+] Running 1/1
 ✔ Container vaultwarden  Started
```

Migracja schematu bazy wykonuje się sama przy starcie nowego obrazu: nie ma osobnej komendy migracji do wywołania ręcznie.

4. Weryfikacja po aktualizacji

```
docker compose -f /docker/compose/vaultwarden/compose.yaml logs vaultwarden | tail -n 30
```

Sprawdzamy w logu brak błędów przy starcie i w stopce strony logowania w przeglądarce nowy numer wersji.

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: stopka strony logowania Vaultwardena po zalogowaniu do panelu `/admin`, z widocznym numerem wersji zgodnym z nowym tagiem obrazu.

Weryfikacja

```
docker compose -f /docker/compose/vaultwarden/compose.yaml ps
```

Przykład:

```
NAME          IMAGE                          STATUS
vaultwarden   vaultwarden/server:1.38.0      Up 3 minutes
```

```
curl -s -o /dev/null -w "%{http_code}\n" https://hasla.zielona.example/alive
```

Przykład:

```
200
```

Wycofanie zmian

Przed migracją (kopia z kroku 2 aktualizacji jeszcze nie była potrzebna): przerywamy, wracamy do poprzedniego tagu w `.env`, `docker compose up -d` uruchamia starą wersję na niezmienionej jeszcze bazie.

Po migracji, gdy nowa wersja działa niepoprawnie: samo cofnięcie tagu obrazu nie wystarcza, baza ma już nowy schemat, którego stara wersja nie rozumie. Zatrzymujemy kontener, podmieniamy `db.sqlite3` i pozostałe pliki na kopię z kroku 2 aktualizacji, dopiero potem uruchamiamy Vaultwardena ze starym tagiem obrazu wskazanym w `.env`.

```
docker compose -f /docker/compose/vaultwarden/compose.yaml down
cp /docker/apps/vaultwarden/kopie/db-przed-aktualizacja.sqlite3
/docker/apps/vaultwarden/db.sqlite3
```

Gdy coś nie zadziała

Objaw	Przyczyna	Co robimy
`sqlite3 .backup` kończy się błędem `database is locked`	równoczesny punkt kontrolny WAL akurat w toku	ponawiamy komendę, przy stałych błędach sprawdzamy obciążenie dysku pod `/docker/apps/vaultwarden`
Kopia pliku `db.sqlite3` zrobiona zwykłym `cp` nie odtwarza się poprawnie	plik skopiowany bez uwzględnienia `db.sqlite3-wal`, niespójny stan	powtarzamy kopię komendą `sqlite3 .backup` zamiast `cp`, patrz sekcja „Dlaczego nie kopiujemy pliku na żywo"
Po aktualizacji kontener bez końca się restartuje	migracja przerwana w połowie, na przykład restart hosta w trakcie	odtwarzamy z kopii sprzed aktualizacji, nie próbujemy uruchamiać ponownie tego samego obrazu na uszkodzonej w połowie bazie
Po odtworzeniu kopii logowanie działa, ale wszystkie sesje wylogowane	plik `rsa_key.pem` nie został odtworzony razem z bazą	odtwarzamy `rsa_key.pem` i `rsa_key.pub.pem` z tej samej kopii co `db.sqlite3`, nigdy osobno

Powiązane: Instalacja Vaultwardena w Dockerze · Świat przykładów: Zielona · cron: zadania cykliczne · systemd timery: OnCalendar zamiast crona · Kopia całego docker01: /docker/apps, /docker/compose i bazy w kontenerach · Strategia kopii zapasowych Zielonej: RPO, RTO, reguła 3-2-1-1-0 i dwie lokalizacje · Vaultwarden: częste problemy

Źródła: github.com/dani-garcia/vaultwarden, wiki „Backing up your vault" · dokumentacja SQLite, rozdział „Online Backup API" i polecenie `.backup` · `sqlite3 --help`

Vaultwarden: częste problemy

Poziom: operacyjny **Dotyczy: Vaultwarden 1.37, Nginx jako reverse proxy, MailCow**

Czas: ~20 min

Zestawienie objawów, na które zespół IT trafia najczęściej przy pracy z Vaultwardenem: dostęp przez HTTPS, powiadomienia na żywo, pocztę i panel administracyjny. Tabela niżej daje szybką odpowiedź, rozwinięcie pod nią tłumaczy przyczynę do końca.

Skróty użyte na tej stronie (6)

Skrót	Co znaczy
HTTPS	HTTP zabezpieczony szyfrowaniem TLS
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
KDF	Key Derivation Function, funkcja wyprowadzenia klucza: algorytm obliczający z hasła głównego klucz szyfrujący sejf, celowo powolny, żeby utrudnić łamanie hasła siłową metodą
SMTP	Simple Mail Transfer Protocol: protokół wysyłania poczty
VPN	Virtual Private Network: szyfrowany tunel do sieci firmowej
WebSocket	stałe, dwukierunkowe połączenie przeglądarki z serwerem, przez które Home Assistant na żywo odświeża panele; serwer pośredniczący musi przekazywać nagłówki Upgrade i Connection

Fiszka

TYLKO ODCZYT

Sprawdzenie logu kontenera Vaultwardena: docker compose logs

```
docker compose -f /docker/compose/vaultwarden/compose.yaml logs --tail 100 vaultwarden
```

Pierwsze miejsce diagnozy niemal każdego z objawów niżej.

□□ **docker** = narzędzie do kontenerów

□□ TYLKO ODCZYT

Sprawdzenie aktualnej wartości DOMAIN: docker compose exec env

```
docker compose -f /docker/compose/vaultwarden/compose.yaml exec vaultwarden printenv DOMAIN
```

Przykład:

```
https://hasla.zielona.example
```

□□ **docker** = narzędzie do kontenerów

□□ TYLKO ODCZYT

Sprawdzenie, czy serwer w ogóle odpowiada: curl /alive

```
curl -s https://hasla.zielona.example/alive
```

Endpoint `/alive` zwraca samą datę serwera, bez logowania: szybki test, że Nginx i kontener w ogóle rozmawiają, zanim zaczniemy szukać przyczyny głębiej.

□□ **curl** = client URL: pobieranie i wysyłanie przez HTTP z terminala

Częste błędy

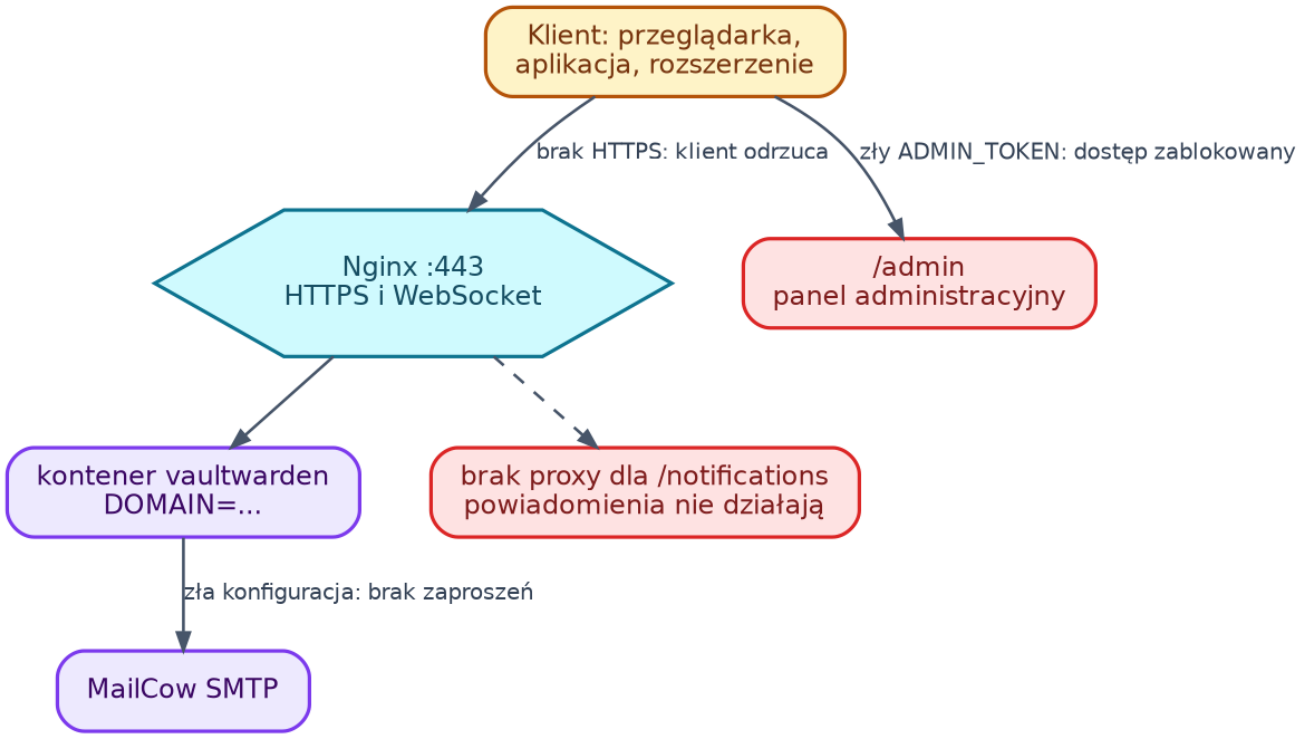
Objaw	Przyczyna	Co robimy
Klient nie odblokowuje sejfu, mimo że strona się otwiera	dostęp po `http://` zamiast `https://`, przeglądarka blokuje Web Crypto API poza bezpiecznym kontekstem	otwieramy wyłącznie adres z `https://`, sprawdzamy certyfikat w Nginksie
Powiadomienia w interfejsie i synchronizacja między urządzeniami nie działają na żywo	Nginx nie przekazuje nagłówków `Upgrade` / `Connection` dla ścieżki `/notifications/hub`	uzupełniamy blok `location /notifications/hub` w konfiguracji Nginksa
Linki w mailach z zaproszeniami wskazują zły adres	`DOMAIN` w `compose.yaml` niezgodny z rzeczywistym adresem instancji	poprawiamy `DOMAIN`, `docker compose up -d`

Objaw	Przyczyna	Co robimy
Zaproszenia i powiadomienia mailowe nie wychodzą	zła konfiguracja SMTP albo MailCow nieosiągalny z `docker01`	`docker compose logs vaultwarden \

Objaw	Przyczyna	Co robimy
Panel `/admin` odrzuca poprawne hasło	`ADMIN_TOKEN` w `.env` niezgodny z hasłem użytym przy `vaultwarden hash`, albo literówka przy wklejaniu hasza w cudzysłowie	generujemy nowy hasz komendą `vaultwarden hash`, wklejamy cały ciąg `$argon2id$...` w pojedynczym cudzysłowie
Panel `/admin` w ogóle nie otwiera się w przeglądarce	`ADMIN_TOKEN` nieustawiony albo pusty, co w niektórych wersjach wyłącza panel całkowicie	sprawdzamy `docker compose exec vaultwarden printenv ADMIN_TOKEN`, uzupełniamy zmienną i restartujemy kontener
Ktoś zapomniał hasła głównego i prosi o odzyskanie	hasło główne nigdy nie trafia na serwer, sam Vaultwarden nie ma jak go sprawdzić ani odtworzyć	patrz rozwinięcie „Zapomniane hasło główne" niżej

HTTPS, WebSocket i droga żądania

Zanim zaczniemy szukać przyczyny konkretnego objawu, przypominamy sobie, którędy w ogóle idzie żądanie od klienta do kontenera, i gdzie w tej drodze mogą leżeć najczęstsze błędy konfiguracji.



Klient, Nginx i kontener vaultwarden: gdzie leżą najczęstsze przyczyny błędów HTTPS, WebSocket, SMTP i panelu admina (kliknięcie otwiera pełny rozmiar)

Brak HTTPS blokuje klienta jeszcze przed dotarciem do samego Vaultwardena, po stronie przeglądarki. Brak przekazywania `Upgrade`/`Connection` dla `/notifications/hub` nie blokuje logowania ani zapisu haseł, tylko powiadomienia na żywo: objaw łatwy do przeoczenia, bo reszta interfejsu działa normalnie, tylko zmiany trzeba odświeżać ręcznie.

Zapomniane hasło główne

> **Pojęcie: KDF.** Funkcja wyprowadzenia klucza: algorytm, który z hasła głównego i dodatkowej soli oblicza klucz szyfrujący sejf, w sposób celowo powolny i trudny do przyspieszenia sprzętowo. Bitwarden i Vaultwarden liczą go zawsze w kliencie, w przeglądarce albo w aplikacji, nigdy na serwerze.

Serwer Vaultwarden przechowuje tylko zaszyfrowane dane i skrót pochodny hasła użyty do samej weryfikacji logowania, nigdy hasła głównego ani klucza wystarczającego do odszyfrowania sejfu. To zamierzona cecha bezpieczeństwa, ta sama co w oficjalnym Bitwardenie: nawet ktoś z pełnym dostępem administracyjnym do serwera i do bazy danych nie odczyta zawartości niczyjego sejfu bez hasła głównego tej osoby.

Jedyna droga odzyskania dostępu bez znajomości hasła to wcześniej włączona polityka organizacji `Account recovery administration`, opisana na stronie Pierwsza konfiguracja: panel admina, organizacja i zaproszenia. Włączenie jej działa tylko na przyszłość: osoba musi zaakceptować udział w mechanizmie odzyskiwania, zanim zapomni hasła, bo dopiero wtedy jej klucz szyfrujący zostaje dodatkowo zaszyfrowany kluczem organizacji. Konto bez wcześniej włączonej i zaakceptowanej polityki, które straci hasło główne, traci dostęp do swojego sejfu bezpowrotnie: jedyne wyjście to założenie konta od nowa i odtworzenie haseł ręcznie albo z ich innych źródeł.

> W Zielonej włączamy `Account recovery administration` dla całej organizacji od razu przy zaproszeniu nowej osoby, zamiast czekać, aż ktoś zapomni hasła i będzie już za późno na przygotowanie odzyskiwania.

□□ Ćwiczenia

1. Pani Halina zgłasza, że nie może zalogować się do rozszerzenia Vaultwarden w domu, mimo że w biurze działa bez problemu. Wskazujemy najbardziej prawdopodobną przyczynę.

Odpowiedź

Najpierw sprawdzamy, czy w domu w ogóle otwiera się adres `https://hasla.zielona.example` po HTTPS: jeśli firma wystawia go tylko wewnątrz sieci firmowej, bez VPN dom nie ma jak się połączyć. Jeśli adres się otwiera, ale logowanie mimo to nie działa, sprawdzamy `Self-hosted → Server URL` w ustawieniach rozszerzenia na tym domowym komputerze.

1. Zespół zgłasza, że zmiany w kolekcji widoczne są dopiero po ręcznym odświeżeniu strony, nigdy same. Wskazujemy, co sprawdzamy w pierwszej kolejności.

Odpowiedź

Konfigurację Nginksa dla ścieżki `/notifications/hub`: brak przekazywania nagłówków `Upgrade` i `Connection` wyłącza powiadomienia na żywo, ale nie blokuje reszty interfejsu, więc łatwo pomylić ten objaw z czymś poważniejszym.

1. Handlowiec zgubił kartkę z hasłem głównym i prosi administratora o zresetowanie go tak, jak resetuje się hasło domenowe w Active Directory. Wyjaśniamy, dlaczego to nie zawsze jest możliwe.

Odpowiedź

Hasło główne nigdy nie trafia na serwer: klucz szyfrujący sejf powstaje z niego lokalnie w kliencie. Bez wcześniej włączonej i zaakceptowanej przez tę osobę polityki `Account recovery administration` administrator nie ma żadnego technicznego sposobu na odczytanie ani zresetowanie dostępu do istniejącego sejfu, w odróżnieniu od hasła domenowego w Active Directory.

Powiązane: Instalacja Vaultwardena w Dockerze · Pierwsza konfiguracja: panel admina, organizacja i zaproszenia · Konta z Active Directory: Directory Connector i SSO · Kopia zapasowa i aktualizacja Vaultwardena

Źródła: github.com/dani-garcia/vaultwarden, wiki „Enabling admin page" i „Enabling WebSocket notifications" · bitwarden.com/help, artykuły o hasle głównym i KDF · dokumentacja Nginx, dyrektywy `proxy_set_header Upgrade` / `Connection`