SEC-1c-3 Plan-Review — DB Least-Privilege (3-Rollen-Modell)

Projekt: Steve-TradingBot · Phase: SEC-1c-3 · Author: claude-opus-4-7[1m]
Generated: 2026-05-14 06:29 UTC · master HEAD: `d351a3a`
Status: **NO CODE** Plan-Review only — Operator-GO erforderlich vor jeder DB-Änderung.
Empfehlung: **3-Rollen-Modell (app / migrator / legacy-superuser), Split in 1c-3a (Laravel, no-Bot-Touch) + 1c-3b (Bot+Worker, mit Restart-Window)**

0 — Executive Summary

Aktuelle DB-Situation (Live verifiziert 2026-05-14 06:29 UTC): `tradingbot_gui` ist die einzige DB-Rolle und hat **SUPERUSER + Create role + Create DB + Replication + Bypass RLS**. Sie wird von **drei** Clients gleichzeitig genutzt:

- **Laravel-GUI** (steve-tradingbot-gui-Container) via `gui/.env` `DB_USERNAME=tradingbot_gui`
- **Bot** (clawbot-Container) via `GUI_DB_DSN=postgres://tradingbot_gui:…@gui-db:5432/tradingbot_gui`
- **Worker** (clawbot-worker-Container) via `GUI_DB_DSN` + dedizierten `GUI_DB_USER=tradingbot_gui`

Das ist **genau die strukturelle Schwachstelle**, die der INCIDENT 2026-05-12 ausnutzbar gemacht hat: jeder kompromittierte App-Prozess (Laravel, Bot, Worker) kann `DROP TABLE` / `DROP DATABASE` / `ALTER SYSTEM` ausführen.

Ziel SEC-1c-3: Privilege-Separation in **drei** Rollen:

- `tradingbot_gui_app` — Runtime-Rolle für alle 3 Clients. NUR `SELECT/INSERT/UPDATE/DELETE` + `USAGE/SELECT` auf Sequenzen + `TEMPORARY` auf DB. **Kein CREATE/ALTER/DROP**.
- `tradingbot_gui_migrator` — Schema-Migrationen via `php artisan migrate`. `CREATE/ALTER/DROP TABLE/INDEX/SEQUENCE` + alle DML. NICHT im Runtime-Container.
- `tradingbot_gui` (legacy) — SUPERUSER bleibt erhalten als DBA-Notfall-Zugriff via SSH+psql (nie in `.env` eines App-Containers).

Split-Empfehlung wegen Restart-Boundary: SEC-1c-3 in 2 Sub-Phasen:

- **SEC-1c-3a Laravel-Cutover (sicher, kein Bot-Touch)**: Rollen anlegen + GRANT + GUI `.env`-Wechsel + GUI-Container-Restart (~10s GUI-Downtime; Bot/Worker unbeeinflusst).
- **SEC-1c-3b Bot+Worker-Cutover (erfordert Bot/Worker-Restart-Window)**: separate Phase, getrennter Operator-GO. Bot/Worker reconnect zu DB mit neuer App-Rolle.

---

1 — Live-State (Recon-Ergebnisse)

Item	Wert
master HEAD	`d351a3a`
Bot in-container PID	363 (`python3 main.py --paper`)
Worker in-container PID	1 (`python3 -m trading.command_worker`)
BINANCE_TESTNET	`true`
DB-Rollen (postgres `\du`)	**1 Rolle**: `tradingbot_gui` mit Attributen **Superuser, Create role, Create DB, Replication, Bypass RLS**
DB `tradingbot_gui` — Tables	24 (alle Owner `tradingbot_gui`)
DB — Sequenzen	18 (alle Owner `tradingbot_gui`)
DB — Functions	0
DB — Views	0
DB — Indexes	104
DB — Foreign Keys	10
Schema-Owner `public`	`pg_database_owner` (Postgres-Default)
Default-Privileges via `\ddp`	keine gesetzt (= GRANTs müssen für alle bestehenden + zukünftigen Tables manuell)
Laravel `gui/.env` DB-Vars	`DB_CONNECTION=pgsql DB_HOST=gui-db DB_PORT=5432 DB_DATABASE=tradingbot_gui DB_USERNAME=tradingbot_gui`
Bot DSN	`postgres://tradingbot_gui:…@steve-tradingbot-gui-db:5432/tradingbot_gui`
Worker DSN	identisch zu Bot — `tradingbot_gui`-User mit SUPERUSER
aktive Connections jetzt	nur 1 psql-Test-Verbindung von Claude

---

2 — Risk-Profil: vorher vs. nachher

Szenario	Heute (SUPERUSER)	SEC-1c-3 Soll (Least-Privilege)
Laravel-Process kompromittiert (XSS-RCE, dependency-injection, etc.)	**Kann `DROP TABLE` / `DROP DATABASE` / `ALTER SYSTEM` — INCIDENT-Vektor**	**Nur SELECT/INSERT/UPDATE/DELETE; kein Schema-Touch; kein Permission-Escalation**
Bot-Process kompromittiert (Market-data-RCE, etc.)	**Wie oben — gleiche SUPERUSER-Macht**	**Wie oben — gleiches App-User-Limit**
Worker-Process kompromittiert	**Wie oben**	**Wie oben**
Test-Code (`php artisan test`) ohne Safe-Runner	**Kann live-DB wipen (INCIDENT 2026-05-12 Präzedenz)**	**migrate:fresh failed mit Permission-Error — strukturelle Defense**
Migration laufen	OK (SUPERUSER kann alles)	**Läuft mit separater migrator-Rolle, getrennter Pfad**
DBA-Notfall-Zugriff	via SUPERUSER + psql	**via legacy `tradingbot_gui` + psql (unverändert)**

---

3 — 3-Rollen-Modell (detailliert)

Rolle	Zweck	Privs erlaubt	Privs verboten
`tradingbot_gui_app`	Runtime (Laravel + Bot + Worker)	`CONNECT` auf `tradingbot_gui` `USAGE` auf `public`-Schema `SELECT/INSERT/UPDATE/DELETE` auf ALL TABLES in public `USAGE/SELECT` auf ALL SEQUENCES in public `EXECUTE` auf Functions (zukunftssicher; aktuell 0 Funcs) `TEMPORARY` auf DB	**kein** `CREATE` auf Schema (kann keine neuen Tabellen anlegen kein `ALTER` / `DROP` auf Tabellen kein `TRUNCATE` (zu zerstörerisch) kein `SUPERUSER` / `CREATEROLE` / `CREATEDB` / `REPLICATION` / `BYPAS`

`tradingbot_gui_migrator`	Schema-Migrationen (`php artisan migrate`)	`CONNECT`
		`USAGE/CREATE` auf `public`-Schema
		`SELECT/INSERT/UPDATE/DELETE/TRUNCATE/REFERENCES/TRIGGER` auf ALL TABLES
		ALL on ALL SEQUENCES
		`TEMPORARY`
			kein `SUPERUSER` / `CREATEROLE` / `CREATEDB` / `REPLICATION`
			keine Berechtigung andere DBs zu ändern

`tradingbot_gui` (legacy)	DBA-Notfall — nur via SSH+psql vom Host	SUPERUSER bleibt	(unverändert)

GRANT-Sequenz (Plan only)

```
-- AUSFÜHRUNG via `tradingbot_gui` (SUPERUSER) am Host:
-- docker exec -i steve-tradingbot-gui-db psql -U tradingbot_gui -d tradingbot_gui << SQL

-- Variant: Passwoerter werden NIEMALS in diesem SQL-Snippet stehen. Generation:
--    PW_APP=$(openssl rand -base64 24)
--    PW_MIG=$(openssl rand -base64 24)
-- via heredoc + WITH PASSWORD '$PW_APP' — Heredoc-Body wird NICHT geechoed.

-- 1. Rollen anlegen
CREATE ROLE tradingbot_gui_app WITH LOGIN PASSWORD <PW_APP> NOSUPERUSER NOCREATEDB NOCREATEROLE NOREPLICATION;
CREATE ROLE tradingbot_gui_migrator WITH LOGIN PASSWORD <PW_MIG> NOSUPERUSER NOCREATEDB NOCREATEROLE NOREPLICATION;

-- 2. Grants fuer App-User (ZUERST, sicher fuer existing tables)
GRANT CONNECT ON DATABASE tradingbot_gui TO tradingbot_gui_app;
GRANT TEMPORARY ON DATABASE tradingbot_gui TO tradingbot_gui_app;
GRANT USAGE ON SCHEMA public TO tradingbot_gui_app;
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO tradingbot_gui_app;
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO tradingbot_gui_app;
GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA public TO tradingbot_gui_app;

-- 3. Default-Privileges fuer ZUKUENFTIGE Tabellen (= nach Migrationen automatisch)
ALTER DEFAULT PRIVILEGES FOR ROLE tradingbot_gui_migrator IN SCHEMA public
    GRANT SELECT, INSERT, UPDATE, DELETE ON TABLES TO tradingbot_gui_app;
ALTER DEFAULT PRIVILEGES FOR ROLE tradingbot_gui_migrator IN SCHEMA public
    GRANT USAGE, SELECT ON SEQUENCES TO tradingbot_gui_app;
ALTER DEFAULT PRIVILEGES FOR ROLE tradingbot_gui_migrator IN SCHEMA public
    GRANT EXECUTE ON FUNCTIONS TO tradingbot_gui_app;

-- 4. Grants fuer Migrator-User
GRANT CONNECT ON DATABASE tradingbot_gui TO tradingbot_gui_migrator;
GRANT TEMPORARY ON DATABASE tradingbot_gui TO tradingbot_gui_migrator;
GRANT USAGE, CREATE ON SCHEMA public TO tradingbot_gui_migrator;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO tradingbot_gui_migrator;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO tradingbot_gui_migrator;
GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN SCHEMA public TO tradingbot_gui_migrator;

-- 5. Migrator MUSS Owner werden fuer existing tables damit er ALTER kann
--    (alternativ: GRANT REASSIGN OWNED BY tradingbot_gui TO tradingbot_gui_migrator)
--    Aber das macht den Migrator zum De-facto-Owner und nimmt SUPERUSER raus.
--    Pragmatischer: SUPERUSER bleibt Owner; Migrator kriegt ALL GRANTS.
--    Migration via Laravel testet sich vor Cutover (mit migrate --pretend).

-- 6. SUPERUSER NICHT vom Owner zurückziehen (sonst wird REASSIGN bei naechster
--    Migration nötig); SUPERUSER bleibt Owner aller 24 Tabellen + 18 Sequenzen.
--    Defense-in-Depth: SUPERUSER nicht mehr in Container.env
```

---

4 — Cutover-Strategy & Sub-Phase-Split

SEC-1c-3a — Laravel-Cutover (sicher, no-Bot-Touch)

#	Step	Risiko	Zeit
3a.0	Pre-flight + git/PID/TESTNET/managed_proposals-Snapshot	0	2 min
3a.1	pg_dump-Backup (Pflicht durable rule)	0	3 min
3a.2	Generate 2 starke Random-PWs in `/root/.secrets/sec-1c-3a-{app,migrator}.SECRET-READ-ONCE` mode 600	0	1 min
3a.3	Rollen + GRANTs anlegen via Heredoc-SQL (PWs werden NICHT geechoed)	LOW	3 min
3a.4	Verify-Tests: Connect mit `tradingbot_gui_app` — `SELECT` erlaubt, `CREATE TABLE x()` muß fehlschlagen, `DROP TABLE users` muß fehlschlagen, `ALTER SYSTEM` muß fehlschlagen	LOW	5 min
3a.5	Verify-Tests: Connect mit `tradingbot_gui_migrator` — `CREATE TABLE foo()` + `DROP TABLE foo` + `ALTER TABLE users ADD COLUMN ...` alle erlaubt; `CREATE ROLE` muß fehlschlagen	LOW	5 min
3a.6	Backup `gui/.env`	0	1 min
3a.7	`gui/.env` ändern: `DB_USERNAME=tradingbot_gui_app` + `DB_PASSWORD=<app-PW>`	MEDIUM (Wert nur in .env, niemals im Output)	2 min
3a.8	Cache invalidation: `config:clear`	LOW	1 min
3a.9	**GUI-Container restart** — ~10s Downtime; alle aktiven Filament-Sessions invalidiert (akzeptabel, Operator wartet)	HIGH (Operator-Recovery beobachten)	2 min
3a.10	Smoke-Test: `https://gui.*/admin/login` → 200; Filament-Login (steve@gewerbespeicher-rechner.de) klappt; eine simple Read-Page (DecisionLogs) lädt	HIGH (Operator)	5 min
3a.11	Migration-Script-Test: `DB_USERNAME=tradingbot_gui_migrator php artisan migrate --pretend` — bestaetigt Pfad funktioniert ohne wirklich migrieren	LOW	2 min
3a.12	Defense-in-Depth: `ALTER USER tradingbot_gui WITH NOSUPERUSER NOCREATEDB NOCREATEROLE NOREPLICATION` ? → DEFERRED zu 1c-3b/1c-3c (Operator-Pin nötig)	OPTIONAL	1 min
3a.13	Closure-Pin + STOP-Report	0	3 min

Σ SEC-1c-3a: ~35 min, davon ~10s GUI-Container-Restart.

SEC-1c-3b — Bot+Worker-Cutover (eigene Phase, Operator-GO erforderlich)

#	Step	Risiko	Zeit
3b.0	Pre-flight + Operator-Notification: Bot/Worker-Restart-Window nötig	0	2 min
3b.1	Bot-Container .env: `GUI_DB_USER=tradingbot_gui_app` + `GUI_DB_PASSWORD=<app-PW>` + DSN entsprechend	MEDIUM	5 min
3b.2	Worker-Container .env: gleiche Änderungen	MEDIUM	5 min
3b.3	**Bot-Container Restart** — Bot PID 363 wird neu — **Mainnet-Block & Bot-Verhalten verifizieren**	HIGH (Operator-Approve für Restart-Window)	10 min
3b.4	**Worker-Container Restart**	HIGH	5 min
3b.5	Connection-Tests: Bot können Read+Write zu `commands` / `managed_proposals` usw.	HIGH	5 min
3b.6	Worker-Heartbeat-Verifikation	HIGH	5 min
3b.7	Closure-Pin	0	3 min

Σ SEC-1c-3b: ~40 min, mit explizitem Bot/Worker-Restart-Window.

SEC-1c-3c (BACKLOG) — SUPERUSER-Lock-Down

Nach erfolgreichem 1c-3a + 1c-3b: `ALTER USER tradingbot_gui WITH NOSUPERUSER NOCREATEDB NOCREATEROLE NOREPLICATION`. SUPERUSER bleibt nur via psql-Direkt-Reconnect-mit-PW-Reset wieder aktivierbar. Mehr Defense, weniger DBA-Convenience — Operator-Decision.

---

5 — Test-Plan (vor Cutover)

Connect-Test (NACH GRANT, VOR Cutover)

```
# positive case (sollte alle gehen):
PGPASSWORD=<app-PW> psql -U tradingbot_gui_app -h gui-db -d tradingbot_gui -c "SELECT COUNT(*) FROM users;"
PGPASSWORD=<app-PW> psql -U tradingbot_gui_app -h gui-db -d tradingbot_gui -c "INSERT INTO audit_events (event,
occurred_at) VALUES ('sec-1c-3a-test', NOW()); SELECT LASTVAL();"
PGPASSWORD=<app-PW> psql -U tradingbot_gui_app -h gui-db -d tradingbot_gui -c "DELETE FROM audit_events WHERE event='sec-
1c-3a-test';"

# negative case (MUSS alle fehlschlagen mit permission denied):
PGPASSWORD=<app-PW> psql -U tradingbot_gui_app -h gui-db -d tradingbot_gui -c "CREATE TABLE leak_test (id int);"
PGPASSWORD=<app-PW> psql -U tradingbot_gui_app -h gui-db -d tradingbot_gui -c "DROP TABLE users;"
PGPASSWORD=<app-PW> psql -U tradingbot_gui_app -h gui-db -d tradingbot_gui -c "ALTER TABLE users ADD COLUMN leak text;"
PGPASSWORD=<app-PW> psql -U tradingbot_gui_app -h gui-db -d tradingbot_gui -c "TRUNCATE TABLE users;"
PGPASSWORD=<app-PW> psql -U tradingbot_gui_app -h gui-db -d tradingbot_gui -c "CREATE ROLE evil;"

# Migrator-Test (sollte alle gehen):
PGPASSWORD=<mig-PW> psql -U tradingbot_gui_migrator -h gui-db -d tradingbot_gui -c "CREATE TABLE _sec_1c_3_canary (id
int); DROP TABLE _sec_1c_3_canary;"
```

Laravel-Tests via Safe-Runner (NACH 1c-3a cutover)

```
bash gui/scripts/run_tests_safe.sh
# Erwartet: 856/856 ✓ wie zuvor.
# Safe-Runner nutzt SQLite in-memory — DB-Wechsel beeinflusst Tests nicht.
```

---

6 — Operator-Decision-Points (vor GO 1c-3a)

1. **Sub-Phase-Split**: 1c-3a (Laravel-only, kein Bot/Worker-Touch) **jetzt**, 1c-3b später? Empfehlung: **JA**, weil Bot/Worker-Restart-Window bewusst getrennt.

2. **Migrator-Owner-Strategie**: SUPERUSER bleibt Owner; Migrator hat `ALL PRIVILEGES` — OK? (Empfehlung: ja, vermeidet REASSIGN-Komplexität)
3. **SEC-1c-3c (SUPERUSER lock-down)**: jetzt mit-planen oder als BACKLOG für später? Empfehlung: **BACKLOG** bis 1c-3a/1c-3b live + 1 Woche stabil.
4. **PW-Speicherort**: `/root/.secrets/<phase>.SECRET-READ-ONCE` mode 600 + Operator-Shred nach PW-Manager-Import. Standard-Pattern, OK?
5. **Migration-CLI-Vereinfachung**: ein Wrapper-Script `/projekte/Steve-TradingBot/gui/scripts/run_migrate_safe.sh` der automatisch zwischen App-User (Runtime) und Migrator-User (Migrations) wechselt? Empfehlung: ja, BACKLOG nach 1c-3a.
6. **Filament-Tinker-Zugriff**: nutzt `php artisan tinker` die App-User-Connection oder soll es eine separate Read-Only-Rolle dafuer geben? Empfehlung: bleibt App-User für jetzt; Read-Only-Rolle Optional BACKLOG.

---

7 — Stop-Regeln

- **STOP**: positive-case-Test mit App-User schlägt fehl (z. B. SELECT auf users) → GRANT inkomplett, Rollback
- **STOP**: negative-case-Test mit App-User **klappt** (z. B. CREATE TABLE) → Defense kaputt, sofortiger Rollback
- **STOP**: nach `.env`-Wechsel + Container-Restart kann GUI sich nicht zu DB connecten → sofortiger Revert auf alten `tradingbot_gui`-User
- **STOP**: Filament-Login schlägt fehl nach Cutover → Rollback
- **STOP**: Bot/Worker-PID ändert sich während 1c-3a (Bot ist explizit out-of-scope)
- **STOP**: irgendwelche Secret-Werte (App-PW, Migrator-PW) erscheinen im Output / Memory / Commit / PDF
- **STOP**: Mainnet-Toggle, Bot-Code-Touch, runtime_config-touch, managed_state-touch

---

8 — Rollback-Pfade

Failure-Zeitpunkt	Rollback
nach 3a.3 (Rollen + GRANTs angelegt, App-Test failed)	`DROP ROLE tradingbot_gui_app; DROP ROLE tradingbot_gui_migrator;` — sauber, kein .env-Touch
nach 3a.7 (.env geändert, Container noch nicht restartet)	backup-.env zurück + `config:clear`
nach 3a.9 (Container restarted, GUI broken)	backup-.env zurück + `docker restart steve-tradingbot-gui`
komplettes Rollback	`DROP ROLE`s + .env-Restore + Container-Restart — < 5 min

---

9 — NO-GO-Bedingungen

- **NO-GO** ohne pg_dump-Backup vor 3a.3
- **NO-GO** ohne .env-Backup vor 3a.7
- **NO-GO** ohne erfolgreichen positive+negative Connect-Tests (3a.4 + 3a.5)
- **NO-GO** Bot/Worker-Touch innerhalb 1c-3a
- **NO-GO** Bot/Worker-Restart innerhalb 1c-3a
- **NO-GO** Mainnet-Toggle
- **NO-GO** docker cp auf clawbot
- **NO-GO** Secret-Werte im Output / Memory / PDF / Commit-msg / Logs
- **NO-GO** Push (Closure rein host-side: nur DB + .env, kein Repo-File)

---

10 — Boundaries (Plan-Review-End)

master HEAD	`d351a3a` unverändert
Bot in-container PID	363 unverändert
Worker in-container PID	1 unverändert
BINANCE_TESTNET	true
managed_proposals	0
DB-Rollen-Change	0 (geplant in 1c-3a)
.env-Change	0
Container-Restart	0
Mainnet	0
Push	0
Secret-Werte im PDF / Output	0 (durable Hygiene-Regel)

---

11 — Phasen-Einordnung & Empfehlung

Phase	Status
SEC-1c-0	✅ closed (ufw + nginx-Basic-Auth + netdata/cups + fail2ban-nginx)
SEC-1c-1	✅ closed (HTTPS + multi-SAN-Cert + Session-Hardening + Permissions-Policy + CSP-Report-Only)
SEC-1c-2a	✅ closed (HSTS 1d→1w + SESSION_LIFETIME=30)
SEC-1c-2b	⏸ DEFERRED bis 2026-05-21 (CSP enforce + HSTS staged-bump weiter)
SEC-1c-3a	🆕 dieses Plan-Review (Laravel-Cutover, kein Bot/Worker-Touch)
SEC-1c-3b	geplant nach SEC-1c-3a (Bot+Worker-Cutover, Restart-Window)
SEC-1c-3c (BACKLOG)	SUPERUSER lock-down nach 1c-3a/1c-3b stabil
SEC-1c-4	Admin-Rotation (Email schon zu steve@..., Cleanup pending)
SEC-1c-5	Audit-Alerts
SEC-1c-6	SSH-Finetuning
SEC-1d	Passkey/WebAuthn
SEC-1e	Final Secret-Rotation (Tokens + APP_KEY + DB-PWs)

Empfehlung

GO SEC-1c-3a atomic Block (35 min, kein Bot/Worker-Touch). Danach **STOP** und Operator-Verifikation auf gui.gewerbespeicher-rechner.de bevor Operator-GO für SEC-1c-3b ausgesprochen wird.

Status

PLAN-REVIEW Plan fertig. Warte auf:

- Antworten auf Decision-Points 1–6 (mind. 1, 2 nötig; 3-6 haben sinnvolle Defaults)
- GO für **SEC-1c-3a** (13-Step-Block, ~35 min)

Kein Code geschrieben. Keine DB-Rollen-Änderung. Keine .env-Änderung. Pure analysis only.