SEC-1c-1c Plan-Review — SMTP + Filament Password-Reset

Projekt: Steve-TradingBot · Phase: SEC-1c-1c (parallel zu SEC-1c-1b) · Author: claude-opus-4-7[1m]
Generated: 2026-05-13 20:22 UTC · master HEAD: `2fbc7b5`
Status: **NO CODE** Plan-Review only — Operator-GO erforderlich vor Implementation.
Empfehlung: **webgo SMTP via STARTTLS 587 + Filament-native passwordReset() + Admin-Email-Update + SPF/DKIM-DNS**

SECRET-HYGIENE-PIN: SMTP-Username + SMTP-Passwort wurden im Conversation-Transcript bereitgestellt (Operator-Risk-Acceptance, gleiche Logik wie Telegram/OpenAI/Anthropic-Tokens vom 2026-05-13). Werte landen ausschließlich in `gui/.env` mode 600 in der Implementation-Phase — **niemals** in diesem PDF, in Memory-Pins, in Commit-Messages oder in nachfolgenden Chat-Antworten. Rotation der SMTP-Credentials wird Teil von SEC-1e Final Token-Rotation.

0 — Executive Summary

Ziel: aktivieren der Filament-eingebauten Password-Reset-Flow, sodass Operator kein manuelles tinker-Reset mehr braucht und PWs niemals den Server verlassen (Self-Service via E-Mail-Token).

Voraussetzungs-Chain: (a) funktionierender Laravel-Mailer, (b) echte Admin-E-Mail-Adresse, (c) Filament-Panel mit `->passwordReset()`, (d) DNS-Records (SPF, optional DKIM) für Mail-Deliverability.

Live-Findings:

- Aktuelle `gui/.env` enthält **keine** `MAIL_*`-Settings — Laravel fällt auf defaults zurück (kein Versand möglich).
- Aktueller Admin-User: `admin@example.local` — nicht existierende Adresse, Reset-Mail würde verworfen.
- `password_reset_tokens`-DB-Tabelle existiert bereits (Laravel-Default-Schema seit Setup).
- `AdminPanelProvider` hat **kein** `->passwordReset()`; Filament's RequestPasswordReset-Page existiert in vendor/, muss aber per Config aktiviert werden.
- Kein MX-Record / kein SPF-Record auf `gewerbespeicher-rechner.de` — Mails werden vermutlich im Spam-Folder landen ohne SPF-Setup.

SMTP-Provider: `s137.goserver.host` (webgo-Hosting-Infrastruktur, da Operator-Domain bei webgo liegt). Credentials sind operator-bereitgestellt. Setup ist Standard-SMTP-AUTH.

---

1 — SMTP-Provider Analyse (webgo / goserver.host)

Parameter	Wert	Empfehlung
SMTP-Host	`s137.goserver.host`	direkt verwenden
Port (Optionen)	465 SSL/TLS **oder** 587 STARTTLS	**587 STARTTLS** (Standard für Submission, besser firewall-kompatibel; UFW erlaubt outgoing nach default-allow)
Encryption	TLS (STARTTLS auf 587)	`MAIL_ENCRYPTION=tls` in Laravel
Auth	USER + PASSWORD	`MAIL_USERNAME=<webgo-account>` + `MAIL_PASSWORD=<>` — nur in `gui/.env` mode 600
Inbound (MX)	nicht nötig	wir senden nur outbound; Reset-Token-Mail landet beim Operator (talk@kw-baustoffe.de o.ä.)
Reverse-DNS / EHLO	nginx-Host vs SMTP-EHLO	webgo kümmert sich (relay-server); kein Setup-Bedarf auf unserer Seite
Connection-Test (geplant in Step 1.6)	`php artisan tinker; Mail::raw('SMTP-test',...);`	vor Filament-Aktivierung verifizieren

Warum 587 STARTTLS über 465 SSL/TLS?

- **RFC 8314** empfiehlt 587 als Submission-Port; 465 ist legacy „smtps".
- Laravel/Symfony-Mailer (seit Symfony 6) unterstützt 587+STARTTLS first-class.
- Beide funktionieren prinzipiell — aber 587 ist robuster gegenüber zukünftigen Firewall-/UFW-Änderungen.

---

2 — Vorbedingung: Admin-E-Mail-Update

Aktuell: `users.id=23, email='admin@example.local'` — passt zu keiner echten Inbox.

Wahl der neuen E-Mail-Adresse

Option	Adresse	Pro	Contra
(a) Operator-Inbox	`talk@kw-baustoffe.de` ☆	real existierende Inbox; kontrolliert vom Operator; Reset-Mails landen dort	From-Address & To-Address sind unterschiedliche Domains
(b) eigene Domain-Inbox	`admin@gewerbespeicher-rechner.de`	einheitliche Brand-Domain	setzt Mailbox bei webgo voraus — Operator-Setup-Aufwand
(c) bestehende webgo-Mailbox	unbekannt	nutzt bestehenden Account	Operator müsste Mailbox-Adresse mitteilen

Empfehlung: **(a)** — pragmatisch, Operator-Inbox aus dem System-Prompt bekannt, keine zusätzliche Setup-Arbeit.

SQL-Update

```
BEGIN;
UPDATE users
   SET email = '<new-email>',
       updated_at = NOW()
 WHERE id = 23
   AND email = 'admin@example.local';
-- 1 row updated expected
SELECT id, email, role FROM users WHERE id = 23;
COMMIT;
```

Boundary: kein Schema-Change, nur 1-row data-update. pg_dump vorher (durable rule).

---

3 — Filament Password-Reset Architektur

Vorhandene Bausteine (in vendor/)

Class	Funktion
`Filament\Auth\Pages\PasswordReset\RequestPasswordReset`	Public-Page: Email-Eingabe-Form, sendet Token-Mail
`Filament\Auth\Pages\PasswordReset\ResetPassword`	Public-Page: Token-Validierung + neues PW setzen
`Filament\Auth\Notifications\ResetPassword`	Mail-Notification-Klasse mit Token-Link
`password_reset_tokens` Tabelle	Laravel-Default; existiert bereits — Schema: email PK + token + created_at

Aktivierung im AdminPanelProvider

EIN-Zeilen-Änderung in `app/Providers/Filament/AdminPanelProvider.php`:

```
$panel
    ...
    ->login(StrictLoginPage::class)
    ->passwordReset()         // NEU: aktiviert RequestPasswordReset + ResetPassword
    ->multiFactorAuthentication([...])
    ->requiresMultiFactorAuthentication()
    ...
```

Resultierende Routen

- `/admin/password-reset/request` — Email-Eingabe
- `/admin/password-reset/reset/{token}` — PW-Reset mit Token
- Login-Page bekommt automatisch einen „Forgot password?"-Link

Flow-Diagramm

```
1. Operator klickt "Forgot password?" auf https://<domain>/admin/login
2. RequestPasswordReset-Page erscheint -> Operator gibt Email ein
3. Filament generiert Token, INSERT in password_reset_tokens
4. Filament sendet Mail via Laravel-Mailer (= webgo SMTP)
5. Operator empfängt Mail mit Link https://<domain>/admin/password-reset/reset/<token>
6. Operator klickt Link -> ResetPassword-Page (PW-Eingabe-Form)
7. Operator gibt neues PW ein, Filament bcrypt-Hash + UPDATE users
8. Token wird DELETE aus password_reset_tokens (one-time)
9. Operator wird zu /admin/login redirected
10. Erst-Login -> MFA-Setup-Forced-Page (SEC-1b-6) sofern noch nicht enrolled
```

MFA-Interaktion

Wichtig: Password-Reset macht **kein** MFA-Reset. Der `app_authentication_secret` bleibt in der DB. Falls Operator-MFA-Setup verloren ist, ist das ein **separater** Break-Glass-Pfad (siehe SEC-1b-6 Runbook). Aktuell hat `admin@example.local` keinen MFA-Setup (`app_authentication_secret IS NULL`) — nach erfolgreicher PW-Reset wird also direkt forced-MFA-Enrollment ausgelöst (SEC-1b-6 Verhalten).

---

4 — DNS-Konfiguration für Mail-Deliverability

Status

- **kein SPF-Record** auf `gewerbespeicher-rechner.de` — Mails können als unauthentisiert klassifiziert werden (Spam-Folder-Risiko)
- kein DKIM-Record (webgo bietet DKIM-Setup im Admin-Panel an — Operator-Aufgabe bei Bedarf)
- kein DMARC-Record (optional, BACKLOG)

Empfohlene SPF-Erweiterung (von Operator im webgo-DNS-Panel zu setzen)

```
; SPF für outbound-Mails via webgo SMTP-Relay
gewerbespeicher-rechner.de.  TXT  "v=spf1 include:goserver.host ~all"
```

Bedeutung:

- `include:goserver.host` : webgo's SMTP-Outbound-IPs sind erlaubte Sender für Mails von `<sender>@gewerbespeicher-rechner.de`
- `~all` : soft-fail für andere Sender (lieber zuerst `~all` , später auf `-all` verschaerfen)

Aber: nur relevant wenn From-Address = `*@gewerbespeicher-rechner.de`

Wenn der MAIL_FROM_ADDRESS eine **fremde Domain** ist (z. B. eine webgo-eigene Mailbox-Domain), greift SPF von *jener* Domain — webgo kümmert sich.

Decision-Point: welche From-Address wird genutzt? Davon hängt SPF-Bedarf ab.

---

5 — `gui/.env` Soll-Werte

Zu setzende Variablen (Klartext nur in `gui/.env` mode 600, nicht hier):

```
MAIL_MAILER=smtp
MAIL_HOST=s137.goserver.host
MAIL_PORT=587
MAIL_USERNAME=<operator-provided>     # webgo SMTP-User
MAIL_PASSWORD=<operator-provided>     # webgo SMTP-Passwort
MAIL_ENCRYPTION=tls                      # STARTTLS auf 587
MAIL_FROM_ADDRESS=<operator-decision>   # siehe Decision-Point unten
MAIL_FROM_NAME="Steve TradingBot"
```

Boundary: nur diese 7 Zeilen werden hinzugefügt; alle bestehenden `.env` -Einträge bleiben unberührt. Backup-Copy vor Änderung (Pflicht).

---

6 — AdminPanelProvider Änderung

Eine Zeile zu addieren:

```
->passwordReset()
```

Position: nach `->login(StrictLoginPage::class)` und vor `->multiFactorAuthentication([…])` (Reihenfolge unkritisch, aber konsistente Convention).

Boundary: AdminPanelProvider hat aktuell folgende panel-Builder-Calls:

- `->default()` · `->id('admin')` · `->path('admin')`
- `->login(StrictLoginPage::class)`
- **NEU**: `->passwordReset()`
- `->multiFactorAuthentication([AppAuthentication::make()->recoverable()])`
- `->requiresMultiFactorAuthentication()`
- `->brandName('Trading Bot Control Center')`
- weitere unverändert

---

7 — Implementation-Sequenz (15 Sub-Steps)

#	Aktion	Risiko	Zeit
1.0	Pre-flight-Snapshot (git, Bot/Worker PIDs, BINANCE_TESTNET, managed_proposals, ufw, nginx-Status)	0	3 min
1.1	Backup: pg_dump GUI-DB + `gui/.env` -Copy nach `/root/sec-1c-1c-backup-<ts>/` mode 600	0	3 min
1.2	SPF-Record bei webgo-DNS-Panel setzen (**Operator-Aufgabe**, kann parallel)	LOW	5 min
1.3	`gui/.env` ergaenzen um 7 `MAIL_*` -Zeilen (Klartext bleibt in .env, nicht im Output)	LOW	5 min
1.4	`php artisan config:clear` in GUI-Container (kein Restart; `.env` wird beim nächsten Request neu gelesen)	LOW	1 min
1.5	SMTP-Connection-Smoke-Test via `php artisan tinker`: `Mail::raw('test',...)->send();` → Operator-Inbox	MED (Auth-Failure möglich)	5 min
1.6	SQL: `UPDATE users SET email = '<real>' WHERE id=23 AND email='admin@example.local'`	LOW	2 min
1.7	`AdminPanelProvider` ergaenzen um `->passwordReset()`	LOW	2 min
1.8	`php artisan config:clear + filament:cache-components` (kein Restart)	LOW	2 min
1.9	`nginx -t` + ggf. reload (sollte nicht nötig sein; aber: prophylaktisch)	0	1 min
1.10	Smoke-Test 1: `curl -k https://81.169.213.37/admin/password-reset/request` → expect 200 (page erscheint)	LOW	2 min
1.11	Smoke-Test 2: Browser-Test der Forgot-Password-Flow — Email eingeben, Mail empfangen, Link klicken, neues PW setzen	HIGH (Operator-Aufgabe)	5 min
1.12	Verifizieren: Login mit neuem PW → forced MFA-Setup-Page (SEC-1b-6) → Authenticator-App scannen → Recovery-Codes sichern → eingeloggt	HIGH (Operator-Aufgabe)	10 min
1.13	Tests via Safe-Runner: `bash gui/scripts/run_tests_safe.sh --filter MfaTotpTest` + Subset von Filament-Resource-Tests — verify keine Regression	LOW	10 min
1.14	Optional: Closure-Pin + Memory-Update + Commit (falls Repo-Files geändert wurden: `app/Providers/Filament/AdminPanelProvider.php`)	LOW	5 min
1.15	STOP-Report	0	2 min

Σ SEC-1c-1c: ~60 min + Operator-Browser-Test-Zeit.

---

8 — Rollback-Pfade

Failure-Zeitpunkt	Rollback
nach Step 1.3 (`.env` -Edit)	`cp /root/sec-1c-1c-backup-<ts>/gui.env /projekte/Steve-TradingBot/gui/.env && config:clear`
nach Step 1.5 (SMTP-Test failed)	diagnostisch: AUTH-Fehler? Port-Block? Encryption-Mismatch? Operator-Decisions reviewen. .env zurück oder fix-iterate.
nach Step 1.6 (Email-Update kollidiert?)	`UPDATE users SET email = 'admin@example.local' WHERE id=23` (aus pg_dump-Backup wiederherstellbar)
nach Step 1.7 (passwordReset() bricht panel)	Zeile aus AdminPanelProvider entfernen + config:clear
nach Step 1.11 (Reset-Link 404 oder Token invalid)	Diagnostik: `password_reset_tokens` -Tabelle? Token-Lifetime? Logs. Fix-iterate.
komplettes Rollback	pg_dump-restore + `.env` -restore + AdminPanelProvider-Edit rückgängig + config:clear — < 5 min

---

9 — Stop-Regeln

- **STOP** bei Auth-Failure im SMTP-Test (Step 1.5) → Diagnose vor weiterer Änderung
- **STOP** bei `nginx -t` -Fehler — nicht reloaden
- **STOP** wenn `users.id=23` -UPDATE mehr als 1 Row betrifft
- **STOP** wenn Test-Mail nach >5 min nicht ankommt (Spam-Folder prüfen; falls dort: SPF-Setup nachholen, sonst Provider-Block)
- **STOP** wenn Bot/Worker-PID sich während des Cutovers ändert
- **STOP** wenn ein Secret-Wert (SMTP-PW oder neues User-PW) im Output / Log sichtbar wird — Hygiene-Regel
- **STOP** bei Forgot-Password-Flow-Failure auf Browser-Test — Diagnose-Logs + Rollback statt Workaround

---

10 — NO-GO-Bedingungen

- **NO-GO** ohne pg_dump-Backup vor Step 1.6 (UPDATE users)
- **NO-GO** ohne `gui/.env` -Backup vor Step 1.3
- **NO-GO** wenn SMTP-Test-Mail mit Klartext-Secret in einem Logfile landet
- **NO-GO** wenn Bot/Worker-Restart implizit nötig wäre
- **NO-GO** Mainnet-Toggle
- **NO-GO** Push ohne Operator-GO
- **NO-GO** SMTP-Pw oder neues User-PW in Memory-Pin / PDF / Commit / Chat

---

11 — Operator-Decision-Points

1. **MAIL_FROM_ADDRESS**:
   - (a) `steve@gewerbespeicher-rechner.de` (eigene Domain — braucht SPF-Setup bei webgo)
   - (b) `noreply@gewerbespeicher-rechner.de`
   - (c) bestehende webgo-Mailbox (welche Adresse?)
   - (d) Default-Adresse die mit dem webgo-Account verknüpft ist (Operator prüft webgo-Panel; konkrete Adresse niemals im PDF)
2. **MAIL_FROM_NAME**: `Steve TradingBot` (Empfehlung) oder andere?
3. **Neue Admin-Email für users.id=23**: `talk@kw-baustoffe.de` (Empfehlung) oder andere?
4. **SPF-Record-Setup im webgo-DNS-Panel**: Operator-Aufgabe; will Operator das vorher selbst setzen, parallel, oder nach SMTP-Test?
5. **Port-Wahl**: **587 STARTTLS** ☆ oder 465 SSL/TLS?
6. **Email-Verification**: Filament hat eingebaute `->emailVerification()` -Feature — **jetzt aktivieren** (alle neuen User müssen Email bestätigen) oder als BACKLOG?
7. **Erst-Login-Pfad**: nach SEC-1c-1c ist Forgot-Password als Erst-Login möglich. Soll der tinker-Reset-Pfad weiterhin verfügbar bleiben (BACKLOG: Operator-Runbook) oder ausgeschlossen werden?
8. **Reihenfolge zu SEC-1c-1b**: SMTP-Setup (SEC-1c-1c) **vor** Filament-APP_URL-Cutover (SEC-1c-1b)? **Empfehlung**: ja — Forgot-Password funktioniert dann sowohl auf https://81.169.213.37/admin/login (IP-Fallback) als auch später auf https://gui…./admin/login.

12 — Phasen-Einordnung

Phase	Status / Vorbedingung
SEC-1c-1a (HTTPS+Domain+nginx-vhosts)	✅ CLOSED 2026-05-13
SEC-1c-1c (SMTP+Password-Reset)	🆕 dieses Plan-Review
SEC-1c-1b (APP_URL+SESSION_DOMAIN+TrustedProxies+SECURE_COOKIE Filament-Cutover)	geplant nach SEC-1c-1c
SEC-1c-2 (SESSION_ENCRYPT+CSP+HSTS-staged)	nach SEC-1c-1b
SEC-1c-3 (DB-Least-Privilege)	parallel möglich
SEC-1c-4 (Admin-Rotation)	nach SEC-1c-1b/1c; nutzt jetzt PW-Reset-Flow statt manueller Setup
SEC-1c-5 (Audit-Alerts)	parallel
SEC-1c-6 (SSH-Finetuning)	parallel
SEC-1d (Passkey/WebAuthn)	nach SEC-1c-1b (stable APP_URL)
SEC-1e (Final Secret-Rotation)	Pre-Mainnet-Pflicht-Gate. Erweitert um SMTP-User+Passwort (siehe Secret-Hygiene-Pin ganz oben).

13 — Boundaries (Plan-Review-End)

master HEAD	`2fbc7b5` unverändert
git status	unverändert
Bot in-container PID	363 unverändert
Worker in-container PID	1 unverändert
BINANCE_TESTNET	true (whitelist-grep)
managed_proposals / history	0 / 0
UFW	active
nginx	active
fail2ban	active (3 jails)
SMTP-Setup	0 (geplant)
email-UPDATE	0 (geplant)
AdminPanelProvider-Edit	0 (geplant)
SMTP-Werte im PDF / Memory / Output	0 (durable Hygiene-Regel)
docker / restart / reload	0
Mainnet	0
Push	0

14 — Empfehlung & nächste Schritte

Empfehlung: SEC-1c-1c als **1 atomarer Block** mit den 15 Sub-Steps, vor SEC-1c-1b durchführen (sodass Forgot-Password-Flow auf der noch-aktiven IP-only-Filament-URL `https://81.169.213.37/admin/login` testbar ist).

Begründung: das Risiko in SEC-1c-1c ist deutlich geringer als in SEC-1c-1b (kein Restart, kein Session-Domain-Wechsel, nur additive Änderungen). Wenn SEC-1c-1c funktioniert, ist der Filament-Cutover in SEC-1c-1b auch über Forgot-Password recovery-bar — doppelte Safety.

Status

PLAN-REVIEW fertig. Warte auf:

- Antworten auf **Decision-Points 1 (MAIL_FROM_ADDRESS)**, **3 (neue Admin-Email)** — mindestens diese zwei nötig
- Antwort auf Decision-Points 2, 4, 5, 6, 7, 8 (jeweils mit sinnvollen Defaults verfügbar)
- GO für 15-Step-Block

Kein Code geschrieben. Kein docker/git/test-Touch. Pure analysis only.