# SEC-1c-1 Plan-Review — HTTPS + Let's Encrypt + Domain-Split

Projekt: Steve-TradingBot · Phase: SEC-1c-1 · Author: claude-opus-4-7[1m]
Generated: 2026-05-13 19:29 UTC · master HEAD: `2fbc7b5` (SEC-1b-6 closed, SEC-1c-0 LIVE)
Status: **NO CODE / NO RELOAD** Pure Plan-Review — Operator-GO erforderlich vor jeder Implementation.
Empfehlung: **3-vhost-Split mit single multi-SAN certbot-Cert + minimal-Pflicht-Companions (TrustedProxies + SECURE_COOKIE)**

## 0 — Executive Summary

**DNS ist vollständig propagiert**: alle drei Subdomains ( `steve.` / `gui.` / `files.gewerbespeicher-rechner.de` ) resolven via 1.1.1.1 und 8.8.8.8 auf `81.169.213.37`. TTL ~3600s (1h Rollback). Registrar: `webgo.de` (NS ns1-4), separates Hosting bei Strato — Standard-Setup.

**certbot ist NICHT installiert** — Pflicht-Install in Sub-Step 1.0 (apt oder snap).

**Bestehende nginx-Topologie**: drei aktive vhosts ( `dashboard-ssl` Port 443+80, `shares-8088` Port 8088, `wp-test-ssl` Port 8443 orphan). Alle binden an `server_name 81.169.213.37` (IP-only). `dashboard-ssl` proxiet `/` zu Bot-Dashboard `127.0.0.1:8050` mit SEC-1c-0-Basic-Auth; `/shares/` zu `/srv/shares/`. Filament-GUI auf `127.0.0.1:8090` ist **aktuell nicht via nginx erreichbar**.

**Empfohlene Aufteilung**:

- `steve.gewerbespeicher-rechner.de` → Bot-Dashboard (Basic-Auth bleibt; Operator-PW `bot-admin` )
- `gui.gewerbespeicher-rechner.de` → Filament Admin (**kein** Basic-Auth; MFA-Login übernimmt; doppelte Auth-Layer wäre UX-feindlich)
- `files.gewerbespeicher-rechner.de` → `/srv/shares/` mit autoindex (PDFs/Reports, kein Auth)

**Kritischer Pflicht-Companion**: bei `APP_URL=https://gui.…` -Wechsel müssen **gleichzeitig** `TrustedProxies` -Middleware und `SESSION_SECURE_COOKIE=true` aktiviert werden — sonst verliert Laravel das Cookie-Secure-Flag bei nginx-proxy (HTTPS-Erkennung schlägt fehl ohne `X-Forwarded-Proto` -Trust). Operator wird sonst nach erstem Login ausgesperrt.

---

## 1 — Live-State (Recon-Ergebnisse)

| Item | Wert | Note |
|---|---|---|
| DNS A `steve.…` | `81.169.213.37` | via 1.1.1.1 + 8.8.8.8 ✓ |
| DNS A `gui.…` | `81.169.213.37` | via 1.1.1.1 + 8.8.8.8 ✓ |
| DNS A `files.…` | `81.169.213.37` | via 1.1.1.1 + 8.8.8.8 ✓ |
| AAAA | leer | kein IPv6 — OK für SEC-1c-1, IPv6-AAAA als BACKLOG-Item |
| NS | `ns1-4.webgo.de` | Registrar webgo (Strato hostet Server) |
| TTL | ~3600s (1h) | moderat — bei DNS-Änderung 1h Rollback-Latenz |
| certbot | **NICHT INSTALLIERT** | `apt install certbot python3-certbot-nginx` in Sub-Step 1.0 |
| Let's Encrypt-Zertifikate | keine | `/etc/letsencrypt/` existiert nicht |
| nginx Version | 1.18.0 (Ubuntu) | modern; supports TLSv1.3, HTTP/2 (nicht aktiv) |
| nginx sites-enabled | 3: dashboard-ssl, shares-8088, wp-test-ssl | wp-test-ssl ist orphan (502); soll weg in SEC-1c-0.5 |
| GUI `APP_URL` | `http://127.0.0.1:8090` | muss zu `https://gui.…` |
| GUI `APP_ENV` | `production` | OK |
| GUI `APP_DEBUG` | `false` | OK (SEC-1b-0 closure) |
| GUI `SESSION_DRIVER` | `database` | OK |
| GUI `SESSION_LIFETIME` | `120` min | 30 min Empfehlung erst in SEC-1c-2 Block A |
| GUI `SESSION_ENCRYPT` | `false` | wird zu true in SEC-1c-2 |
| GUI `SESSION_DOMAIN` | `null` | muss zu `gui.…` in SEC-1c-1 |
| GUI `SESSION_SECURE_COOKIE` | nicht gesetzt (default `false` ) | **Pflicht-Companion**: true setzen GLEICHZEITIG mit APP_URL-Wechsel |
| GUI `TRUSTED_PROXIES` | nicht konfiguriert | **Pflicht-Companion**: trustProxies-Middleware in `bootstrap/app.php` |
| Filament-Login + MFA | aktiv (SEC-1b-6) | App-Authentication TOTP + Recovery-Codes; Admin-User `admin@example.local` noch ohne Setup (forced enrollment on next login) |
| Bot in-container PID | `363` | `python3 main.py --paper` unverändert |
| Worker in-container PID | `1` | `command_worker` unverändert |
| BINANCE_TESTNET | `true` | via whitelist-grep |
| UFW | aktiv: 22/80/443/8088 allow, default deny | SEC-1c-0 LIVE |
| fail2ban | 3 jails: sshd + nginx-http-auth + nginx-limit-req | SEC-1c-0 LIVE |

---

## 2 — Domain-Aufteilung (Soll)

| Subdomain | Funktion | Internal Backend | Auth-Layer | Notes |
|---|---|---|---|---|
| `steve.gewerbespeicher-rechner.de` | Bot-Dashboard (Trading-State, Positions, Logs) | `127.0.0.1:8050` | nginx Basic-Auth (User `bot-admin` , SEC-1c-0 Step 3) | `/shares/` entfällt hier (wandert nach `files.` ) |
| `gui.gewerbespeicher-rechner.de` | Filament Admin (Trade-Logs, Config, Managed-Proposals) | `127.0.0.1:8090` | Filament-Login + SEC-1b-6 TOTP-MFA (NICHT zusätzlich Basic-Auth) | einziger Pfad für Admin-Aktionen; MFA als sole-2nd-factor |
| `files.gewerbespeicher-rechner.de` | PDF-Reports / Shares | `/srv/shares/` (alias) | kein Auth (autoindex) | Operator-Self-Service; `backups/` bleibt durch dir-mode 0700 verborgen |

### Server-Block-Layout

```
# /etc/nginx/sites-available/steve.gewerbespeicher-rechner.de
server {
  listen 80;
  server_name steve.gewerbespeicher-rechner.de;
  location /.well-known/acme-challenge/ { root /var/www/letsencrypt; }
  location / { return 301 https://$host$request_uri; }
}
server {
  listen 443 ssl http2;
  server_name steve.gewerbespeicher-rechner.de;
```

```
  ssl_certificate     /etc/letsencrypt/live/<cert-name>/fullchain.pem;
  ssl_certificate_key /etc/letsencrypt/live/<cert-name>/privkey.pem;
  ssl_protocols TLSv1.2 TLSv1.3;
  ssl_ciphers HIGH:!aNULL:!MD5;
  ssl_prefer_server_ciphers on;
  # SEC-1c-1: HSTS staged-start 1 day (Phase 2 erhoeht auf 1 week / 1 month / 1 year)
  add_header Strict-Transport-Security "max-age=86400" always;

  location / {
    auth_basic           "Steve Bot Dashboard";
    auth_basic_user_file /etc/nginx/.htpasswd-bot-dashboard;
    proxy_pass http://127.0.0.1:8050;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host              $host;
    proxy_set_header X-Real-IP         $remote_addr;
    proxy_set_header X-Forwarded-For   $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_read_timeout 300s;
  }
}

# /etc/nginx/sites-available/gui.gewerbespeicher-rechner.de
server {
  listen 80;
  server_name gui.gewerbespeicher-rechner.de;
  location /.well-known/acme-challenge/ { root /var/www/letsencrypt; }
  location / { return 301 https://$host$request_uri; }
}
server {
  listen 443 ssl http2;
  server_name gui.gewerbespeicher-rechner.de;
  ssl_certificate     /etc/letsencrypt/live/<cert-name>/fullchain.pem;
  ssl_certificate_key /etc/letsencrypt/live/<cert-name>/privkey.pem;
  ssl_protocols TLSv1.2 TLSv1.3;
  ssl_ciphers HIGH:!aNULL:!MD5;
  add_header Strict-Transport-Security "max-age=86400" always;

  client_max_body_size 5m;

  location / {
    # KEIN Basic-Auth — Filament hat eigene MFA (SEC-1b-6).
    proxy_pass http://127.0.0.1:8090;
    proxy_http_version 1.1;
    proxy_set_header Host              $host;
    proxy_set_header X-Real-IP         $remote_addr;
    proxy_set_header X-Forwarded-For   $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_read_timeout 300s;
  }
}

# /etc/nginx/sites-available/files.gewerbespeicher-rechner.de
server {
  listen 80;
  server_name files.gewerbespeicher-rechner.de;
  location /.well-known/acme-challenge/ { root /var/www/letsencrypt; }
  location / { return 301 https://$host$request_uri; }
}
server {
  listen 443 ssl http2;
  server_name files.gewerbespeicher-rechner.de;
  ssl_certificate     /etc/letsencrypt/live/<cert-name>/fullchain.pem;
  ssl_certificate_key /etc/letsencrypt/live/<cert-name>/privkey.pem;
  ssl_protocols TLSv1.2 TLSv1.3;
  ssl_ciphers HIGH:!aNULL:!MD5;
  add_header Strict-Transport-Security "max-age=86400" always;

  location / {
    alias /srv/shares/;
    autoindex on;
    autoindex_exact_size off;
    autoindex_localtime on;
    types {
      application/pdf  pdf;
      text/plain       txt log md;
```

```
      text/html        html htm;
      application/json json;
    }
    default_type application/octet-stream;
    add_header X-Content-Type-Options nosniff always;
  }
}
```

---

## 3 — Cert-Strategie

| Option | Vor | Nach | Empfehlung |
|---|---|---|---|
| **(a) Single multi-SAN-Cert** `certbot certonly --webroot -w /var/www/letsencrypt -d steve.… -d gui.… -d files.… --cert-name gewerbespeicher` | 1 Renewal-Job; alle Domains atomar; weniger Files in `/etc/letsencrypt/live/` | Bei einer Domain-Renewal-Failure wird alle 3 Domains betroffen | **EMPFOHLEN** |
| **(b) 3 separate Certs** (operator-pin in GO: "Let's Encrypt Zertifikate pro Domain") | Failure-Isolation pro Domain; cleaner conceptual separation | 3 Renewal-Jobs; 3 Cert-Dirs; mehr Wartung | Falls operator-pin strikt |
| (c) Wildcard `*.gewerbespeicher-rechner.de` | 1 Cert für alles, auch für zukünftige Subdomains | Braucht DNS-01 Challenge (kein HTTP-01); braucht DNS-API-Zugang bei webgo | OVERKILL für 3 Domains |

### Operator-Decision

Operator-GO sagte „Let's Encrypt Zertifikate pro Domain“ (Plural). Lesart ambig:

- Lesart 1: **(b) drei separate Certs** — wörtlich
- Lesart 2: **(a) ein Cert mit drei Domains** — „Zertifikate“ meint die Tatsache, dass alle Domains zertifiziert sind

Meine Empfehlung: **(a)** — weniger Operational-Overhead, atomic renewal, Standardpattern. Operator-Anpassung auf (b) jederzeit moeglich.

### certbot-Plugin

- **(i)** `--webroot` : nginx behauptet, `/.well-known/acme-challenge/` wird nach `/var/www/letsencrypt` served. KEIN nginx-restart noetig. Renewals laufen ohne Eingriff. **EMPFOHLEN**
- (ii) `--nginx` (python3-certbot-nginx): certbot patcht die nginx-Config automatisch. Schneller initial setup, aber Drift-Gefahr (nicht-deklarative Änderungen)
- (iii) `--standalone` : certbot oeffnet eigenen :80-Listener — würde nginx-stop erfordern (NO-GO da nginx live serviert)

---

## 4 — Konflikte mit aktuellem `dashboard-ssl`

### 4.1 Port-Clash

Aktueller `dashboard-ssl` bindet `listen 443` ohne `server_name`-Filter (matched alles, weil `server_name 81.169.213.37` nicht Host-Header-basiert greift; nginx liefert das als `default_server` bei IP-Zugriff). Die drei neuen vhosts mit `server_name <subdomain>` binden ebenfalls 443.

**Lösungs-Optionen**:

- **(A) Neue vhosts ZUSÄTZLICH** zu `dashboard-ssl` ; nginx routet per Host-Header. `dashboard-ssl` bleibt als IP-only-Fallback erreichbar bis Schritt N (Cleanup-Cut).
  - Pro: atomar reversibel; alte IP-URLs gehen weiter
  - Contra: Cert-IP-only-Self-Signed bleibt im Umlauf
- **(B)** `dashboard-ssl` **umbauen**: `server_name 81.169.213.37` → `steve.gewerbespeicher-rechner.de` ; `shares/` -Location entfernen (wandert nach `files.` ); SSL-Cert auf Let's-Encrypt umschwenken.
  - Pro: 1 file weniger; sauber
  - Contra: 1 file edit + reload statt 3 neue files

**Empfehlung**: **(A) zusätzlich**; `dashboard-ssl` bleibt vorerst aktiv als Sicherheitsnetz (IP-only Zugriff funktioniert weiter). Nach 1–7 Tage stabiler 3-vhost-Operation: Cleanup-Cut macht `dashboard-ssl` obsolet (steht in BACKLOG SEC-1c-1-FU-1).

### 4.2 default_server-Frage

Mit Subdomain-vhosts: was passiert bei IP-only-Zugriff ( `https://81.169.213.37/` )? Drei Optionen:

1. **(i)** **`dashboard-ssl`** **bleibt default** — IP-Zugriff geht zu Bot-Dashboard (Basic-Auth). Empfohlen für Übergangsphase.
2. (ii) Neuer `default_server` -vhost gibt 444 (Connection-Close) zurück — signalisiert „IP-only nicht erlaubt“.
3. (iii) Redirect IP → `steve.…`

Empfehlung: **(i)** für SEC-1c-1; (ii) später im Cleanup-Cut.

---

## 5 — `APP_URL` / `SESSION_DOMAIN` Auswirkungen + Pflicht-Companions

### Sollwerte (SEC-1c-1)

```
APP_URL=https://gui.gewerbespeicher-rechner.de
SESSION_DOMAIN=gui.gewerbespeicher-rechner.de
SESSION_SECURE_COOKIE=true         # PFLICHT-COMPANION zu APP_URL=https
TRUSTED_PROXIES=*                  # in unserem Setup (localhost-nginx) sicher
```

### Pflicht-Companion 1: TrustedProxies-Middleware

In `gui/bootstrap/app.php` im `->withMiddleware()` -Block **HINZUFÜGEN**:

```
$middleware->trustProxies(
    at: '*',
    headers: \Illuminate\Http\Request::HEADER_X_FORWARDED_FOR
           | \Illuminate\Http\Request::HEADER_X_FORWARDED_HOST
           | \Illuminate\Http\Request::HEADER_X_FORWARDED_PORT
           | \Illuminate\Http\Request::HEADER_X_FORWARDED_PROTO,
);
```

**Begründung**: ohne diesen Block erkennt Laravel den Request als HTTP (nicht HTTPS), weil das eingehende Paket auf `http://127.0.0.1:8090` ankommt. Folge: `SESSION_SECURE_COOKIE=true` wird ignoriert (Cookie wird nicht mit `Secure` -Flag gesetzt). Filament-Redirects gehen auf `http://…` statt `https://…` . **Operator wird sonst aus dem Admin ausgesperrt** nach erstem Redirect.

### Pflicht-Companion 2: `SESSION_SECURE_COOKIE=true`

Sobald HTTPS aktiv ist, MUSS dieser Wert auf `true` , sonst:

- Cookie wird auch über HTTP gesetzt (Downgrade-Vektor)
- HTTP-Endpoints koennten Session-Cookies abgreifen

### Konsequenz: alle bestehenden Sessions invalidiert

`SESSION_DOMAIN` -Wechsel `null` → `gui.…` macht alle bestehenden Cookies ungültig. Operator (und alle Filament-User) müssen neu einloggen. **Akzeptabel im 1-Admin-Setup**.

### Filament `asset_url`

Aktuell `asset_url = NULL` . Filament generiert Asset-Pfade aus `APP_URL` . Da `APP_URL` nach SEC-1c-1 HTTPS-canonical wird, sollten Assets dann mit HTTPS ausgeliefert werden. Mixed-Content-Pruefung im Browser nach Cutover.

---

## 6 — MFA / Passkey-Kompatibilität

| | |
|---|---|
| **SEC-1b-6 TOTP-MFA** | kompatibel: Authenticator-Code ist Domain-unabhängig; nach SEC-1c-1 funktioniert weiterhin. **Operator-Hinweis**: beim ersten Login auf `https://gui.…` wird das forced-enrollment-Setup-Page (SEC-1b-6) gestartet, da Admin `admin@example.local` noch `app_authentication_secret IS NULL` . |
| **SEC-1d Passkey/WebAuthn** | BENOEHTIGT diese Phase — WebAuthn benötigt secure-context (HTTPS mit gültigem TLS auf einer stabilen Domain). SEC-1c-1 ist die **Vorbedingung**. Nach SEC-1c-1 ist SEC-1d entblockt. |
| **SESSION_DOMAIN-Konflikt** | Passkey-RP-ID muss exakt die Domain matchen. SESSION_DOMAIN= `gui.gewerbespeicher-rechner.de` ist konsistent mit RP-ID= `gui.gewerbespeicher-rechner.de` . |
| **Cross-subdomain-MFA** | nicht relevant — MFA gilt nur für Filament-Admin ( `gui.` ); Bot-Dashboard ( `steve.` ) hat eigenen Basic-Auth. |

---

## 7 — CSP / HSTS-Vorbereitung (was schon jetzt, was in SEC-1c-2)

### SEC-1c-1 MIN (Pflicht zusammen mit HTTPS)

- `add_header Strict-Transport-Security "max-age=86400" always;` — staged start (1 Tag), Operator-Pin
- `add_header X-Content-Type-Options "nosniff" always;` — bereits auf `/shares/`, jetzt auf alle vhosts
- `add_header X-Frame-Options "DENY" always;` — auf `gui.` und `steve.`; **nicht** auf `files.` (PDF-Embedding via iframe optional erlaubt)
- `add_header Referrer-Policy "strict-origin-when-cross-origin" always;`

### SEC-1c-2 FOLLOW-UP (Phase 2, nicht jetzt)

- HSTS `max-age=31536000; includeSubDomains` nach 1 Woche fehlerfrei; preload erst nach 1 Monat
- CSP `Report-Only` mit Reporter-URI; nach 1 Woche enforce
- `Permissions-Policy`
- SESSION_ENCRYPT=true, SESSION_HTTP_ONLY=true, SESSION_SAME_SITE=strict
- Idle-Timeout SESSION_LIFETIME=30

---

## 8 — Operator-Decision-Points (Pflicht vor Code-Start)

1. **Cert-Strategie**: (a) single multi-SAN-Cert (Empfehlung) ODER (b) 3 separate Certs?
2. **certbot-Plugin**: (i) `--webroot` mit `/var/www/letsencrypt` (Empfehlung) ODER (ii) `--nginx`?
3. **Cleanup-Cut von `dashboard-ssl`**: Beibehalten (Empfehlung — alte IP-URLs funktionieren weiter während Test-Phase) ODER sofort entfernen nach Cert-Setup?
4. **HTTP/2**: aktivieren in den neuen vhosts (Empfehlung — Performance) ODER nur HTTP/1.1?
5. **HTTP/1.1 80-Port-Fallback**: nur ACME-Challenge + Redirect (Empfehlung) ODER zusätzlich `/shares/` über HTTP wie heute?
6. **files.... backup-shielding**: aktuelle dir-mode 0700 reicht, oder zusätzlich nginx `location ~ /backups` mit `internal;` oder `deny all;`?
7. **SSH-IP-Whitelist gleichzeitig**: hast du deine feste IP, sodass wir `ufw allow from <IP> to any port 22` + `delete allow 22/tcp` beim Cutover machen?
8. **E-Mail für Let's Encrypt**: certbot speichert eine Admin-E-Mail-Adresse für Renewal-Failure-Notifications. Welche E-Mail-Adresse verwenden? (Operator-Default: `talk@kw-baustoffe.de`?)

---

## 9 — Atomicity & Commit-Block-Plan

| # | Step | Aktion | Risiko | Zeit |
|---|---|---|---|---|
| 1.0 | Pre-flight | Live-State-Snapshot (git, Bot/Worker PIDs, BINANCE_TESTNET, managed_proposals, ss -tlnp, ufw status, nginx -t) | 0 | 5 min |
| 1.1 | Backup | nginx-Config gesamt-Tarball + GUI `.env` -Copy + pg_dump GUI-DB (Pflicht, durable-rule) | 0 | 10 min |
| 1.2 | certbot Install | `apt install -y certbot` (kein nginx-plugin wenn webroot-Variante) | LOW (Paket-Install, kein Service-Restart) | 5 min |
| 1.3 | ACME-webroot | `mkdir -p /var/www/letsencrypt/.well-known/acme-challenge && chown -R www-data:www-data` | LOW | 2 min |
| 1.4 | nginx-Skeleton | 3 neue vhost-Configs als Skelett anlegen (NUR Port 80 + ACME-Challenge-Location + 301-Redirect; KEIN HTTPS-Block bis Cert da ist) | LOW (validate via `nginx -t`) | 15 min |
| 1.5 | nginx reload | `nginx -t && systemctl reload nginx` — alte vhosts (dashboard-ssl, shares-8088) bleiben aktiv | LOW | 2 min |
| 1.6 | ACME-Test | `curl http://steve.…/.well-known/acme-challenge/test` → muss 404 (location existiert) statt 403 | LOW | 5 min |
| 1.7 | Cert-Issue | `certbot certonly --webroot -w /var/www/letsencrypt -d steve.… -d gui.… -d files.… --email <op> --agree-tos --no-eff-email` | MEDIUM (Rate-Limits Let's Encrypt; STAGING-Test zuerst empfohlen) | 15 min |
| 1.8 | HTTPS-Block hinzu | 3 vhost-Configs: HTTPS-Block (443) hinzufügen mit Cert-Pfaden; **HSTS max-age=86400**, no preload | MEDIUM (nginx-validate, atomic) | 30 min |
| 1.9 | nginx reload | `nginx -t && systemctl reload nginx` | LOW | 2 min |
| 1.10 | Smoke-Test | curl all 3 https-domains; verify 301 von HTTP, 200/401 auf HTTPS | LOW | 10 min |
| 1.11 | **GUI APP_URL + Session-Companion** | gui/.env ändern: `APP_URL`, `SESSION_DOMAIN`, `SESSION_SECURE_COOKIE`; `gui/bootstrap/app.php` mit TrustedProxies erweitern | HIGH (Login-Cutover; alte Sessions ungueltig) | 20 min |
| 1.12 | GUI-Container restart | `docker exec gui php artisan config:clear && restart` ODER GUI-Container restart (sauber). | HIGH (Service-Downtime ~10s) | 2 min |
| 1.13 | Verifikation | Browser-Test `https://gui.…` → Filament-Login → MFA-Setup-Forced-Page → Setup → Login | HIGH | 30 min |
| 1.14 | Optional Cleanup | orphan `wp-test-ssl` entfernen (separate Sub-Step, BACKLOG SEC-1c-0.5) | LOW | 5 min |
| 1.15 | Closure-Pin | Memory-Pin + PDF-Report | 0 | 20 min |

**Σ SEC-1c-1: ~3h** über 15 atomic steps.

---

## 10 — Rollback-Pfade

| Failure-Zeitpunkt | Rollback-Step |
|---|---|
| nach Step 1.5 (nginx skelett-reload) | `rm /etc/nginx/sites-enabled/{steve,gui,files}.gewerbespeicher-rechner.de && nginx -s reload` — 10 sec, alte IP-only-URLs funktionieren weiter |
| nach Step 1.7 (Cert-Issue failed) | Cert wurde nicht ausgestellt; Skelett-vhosts bleiben (kein Schaden); Re-try mit `--staging` dann production |
| nach Step 1.9 (HTTPS-Block-reload bricht nginx) | `nginx -t` verhindert reload bei Syntax-Fehler; bei semantic-Fehler: backup-tarball restore |
| nach Step 1.12 (Filament-Login broken) | gui/.env aus backup; `docker restart steve-tradingbot-gui`; alte `APP_URL=http://127.0.0.1:8090` funktioniert via SSH-tunnel weiter (Operator-Notfall-Access) |
| komplettes Rollback | backup-tarball restore + `certbot delete --cert-name <name>` + GUI-Container restart — < 5 min |

---

## 11 — Stop-Regeln (Pflicht)

- **STOP**: `nginx -t` fails → KEIN reload
- **STOP**: `certbot certonly` failed → Re-Run mit `--staging`; bei nochmal Failure: Plan-Review-Cut
- **STOP**: SSH-Verbindung gerät in Gefahr (UFW-Regeln ändern während HTTPS-Block-Aufbau)
- **STOP**: Bot/Worker-PID ändert sich während des Cutovers

- **STOP**: GUI-DB-Connection-Errors nach `.env`-Wechsel
- **STOP**: Secret-Wert wird sichtbar (durable Hygiene-Regel)
- **STOP**: Operator-Login auf `https://gui.…` nach Step 1.13 schlägt fehl → sofortiger Rollback-Pfad
- **STOP**: Mixed-Content-Errors im Browser-Test → `asset_url`-Fix oder URL::forceScheme

---

## 12 — NO-GO-Bedingungen

- **NO-GO** ohne pg_dump-Backup vorab (durable rule)
- **NO-GO** ohne nginx-Config-Backup-Tarball vorab
- **NO-GO** wenn Bot/Worker-Restart implizit notwendig würde
- **NO-GO** ohne Cert-Staging-Test vor Production-Cert (Rate-Limit-Schutz)
- **NO-GO** wenn TrustedProxies-Middleware NICHT gleichzeitig mit `APP_URL`-Wechsel landet
- **NO-GO** wenn `SESSION_SECURE_COOKIE=true` NICHT gleichzeitig mit `APP_URL=https://`-Wechsel landet
- **NO-GO** bei Push (alle Änderungen host-side / GUI-Container-side; kein Repo-Push notwendig)
- **NO-GO** Mainnet-Toggle
- **NO-GO** docker cp auf Bot-Files

---

## 13 — GO/NO-GO Empfehlung

**GO-Empfehlung**: SEC-1c-1 ist **vertretbar als 1 atomarer Block** mit den 15 Sub-Steps, sofern:

1. Operator-Antworten auf Decision-Points 1, 2, 8 vorliegen (Cert-Strategie / Plugin / E-Mail)
2. Operator akzeptiert Re-Enrollment-Flow auf erste `https://gui.…`-Login (Forced MFA-Setup für `admin@example.local`)
3. Cert-Staging-Test vor Production-Cert

**Alternativ: Split in 2 Bloecke**:

- SEC-1c-1a: Cert-Setup + nginx-3-vhost (ohne Filament-APP_URL-Wechsel) — sicherer Test-Run; alte URLs funktionieren weiter, neue URLs sind verfügbar
- SEC-1c-1b: APP_URL + SESSION_DOMAIN + TrustedProxies + SECURE_COOKIE (Filament-Cutover) — nach 1–2 Tagen stabiler 1c-1a-Operation

**Meine Empfehlung**: **Split (1c-1a + 1c-1b)**. Weniger HIGH-Risk-Steps in einem Cutover; klarer Rollback-Pfad pro Schritt.

---

## 14 — 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 | 0 |
| UFW | active |
| nginx | active |
| fail2ban | 3 jails active |
| docker / restart / reload | 0 |
| certbot install | 0 (geplant in Step 1.2) |
| cert-issue | 0 (geplant in Step 1.7) |
| nginx config-edit | 0 (geplant in Step 1.4 + 1.8) |
| gui/.env edit | 0 (geplant in Step 1.11) |
| Mainnet | 0 |
| Push | 0 |
| Secret-Werte in Output | 0 (durable Hygiene-Regel) |

---

**Status**: PLAN-REVIEW fertig. Warte auf:

- Antworten auf Decision-Points 1, 2, 8 (mind. Cert-Strategie / Plugin / E-Mail)
- GO für **Split (SEC-1c-1a + SEC-1c-1b)** ODER GO für **atomic 15-step block**
- (optional) GO für begleitende SSH-IP-Whitelist Decision-Point 7

Kein Code geschrieben. Kein git/docker/cert-Issue/nginx-Reload durchgeführt — pure analysis only.