

# PLAN\_TRACK\_SELL\_FAILURE\_ALERT

**Status:** planned (P0) **Priority:** P0 **Erstellt:** 2026-05-18 (Nightly-Audit-Finding) **Quelle:** reports/nightly/2026-05-18/ALERTS.md P0-B **Roadmap-ID:** TRACK-SELL-FAILURE-ALERT **Bezug:** Sibling von PHANTOM-SELL-LOOP-FIX. Gemeinsamer Cutover empfohlen.

## 1. Befund

B-OUTAGE-RESILIENCE-1 introduced `_track_sell_failure(symbol)` in `trading/execution/live_trade.py` als Stuck-Position-Alert-Mechanismus.

Während der 22h SHIB-Loop am 2026-05-17 feuerte dieser Alert **nicht sichtbar** — weder Telegram-Notification noch operator-relevanter Logeintrag. 29× Failure ohne Eskalation.

## 2. Root cause (Hypothesen, zu verifizieren)

Hypothese	Wahrscheinlich?
Threshold zu hoch (z. B. > 30 Versuche)	mittel
Counter wird beim Bot-Restart resettet und kommt nie hoch	hoch (5 cutovers heute)
Reporter- <code>_send</code> -Pfad silent fehl bei HTTP-Fail	mittel
<code>_track_sell_failure</code> ruft nur <code>logger.warning</code> , keinen reporter	hoch

## 3. Methodik (read-only erster Schritt)

- `grep _track_sell_failure trading/execution/live_trade.py` → Definition + alle Callsites
- Threshold ablesen
- Path zum reporter prüfen
- Test: bei 5× simulated fail → fired Telegram? In test ja, in prod nein → diff

## 4. Fix-Strategien

### Strategie A — Threshold senken + Telegram-Eskalation hart wired

```
def _track_sell_failure(self, symbol):
    self._sell_failures.setdefault(symbol, 0)
    self._sell_failures[symbol] += 1
    n = self._sell_failures[symbol]
    # NEW: hardwired Telegram-Alert bei N=3
    if n >= 3:
        self.notify_operator_sell_stuck(symbol, count=n)
    # NEW: nach N=5 als Phantom-Kandidat markieren
    if n >= 5:
        self._mark_phantom_candidate(symbol)
```

### Strategie B — Counter-Persistenz über Bot-Restart

State-File `_sell_failures.json` neben `live_portfolio.json`. Beim Boot eingelesen. Verhindert Reset-by-Restart der Eskalation.

### Strategie C — Cross-Reference mit `fetch_balance`

Bei Trigger: vergleiche `pos['quantity']` mit `fetch_balance(base_asset)`. Drift > 5 % → automatisch eskalieren, nicht warten bis Threshold.

## 5. Empfohlene Kombination

- Strategie A für sichtbaren Telegram-Alert
- Strategie C als zusätzlicher Trigger
- Strategie B optional (Bot wird selten mehrmals täglich neugestartet)

## 6. Boundaries

- Bot-Touch: ja (live\_trade.py)
- DB: keine Migration nötig
- 0x Trading-Logik-Change
- 0x Mainnet
- Gemeinsamer Cutover mit PHANTOM-SELL-LOOP-FIX

## 7. Cutover-Plan

---

Bundle mit PHANTOM-SELL-LOOP-FIX: 1. Pre-Cutover Snapshot 2. Watchdog freeze 3. `docker compose build clawbot` 4. Container-Tests (eine Suite für beide Fixes) 5. Recreate clawbot 6. 3-Way MD5 7. Live-Verify: künstlicher Mock-Fail → Telegram-Alert kommt

## 8. Tests

---

```
trading/tests/test_track_sell_failure_alert.py: - test_threshold_3_fires_telegram -  
test_persistent_counter_across_save_state - test_balance_drift_triggers_immediate_escalation -  
test_no_alert_for_first_2_failures - test_phantom_candidate_marker_after_5
```

## 9. Acceptance Criteria

---

- Operator sieht in Telegram: „SHIB sell-failed 3× consecutive (insufficient\_funds)“
- Counter wird nicht beim Bot-Restart auf 0 zurückgesetzt
- Bei drift > 5 % sofort eskalieren (kein Threshold-Wait)
- Test-Suite grün

## 10. Risk

---

Risiko	Severity	Mitigation
Telegram-Spam wenn Loop nicht gebrochen	mittel	combined mit PHANTOM-SELL-LOOP-FIX
Counter-File Race	niedrig	atomare write+rename
False-positive bei API-outage	mittel	category-check (skip transient errors)

## 11. STOP

---

Bundle-Cutover mit PHANTOM-SELL-LOOP-FIX. Operator-GO erforderlich.