Procedura 6 — Safe Mode (mechanika trybu bezpiecznego)

Wersja: 1.0
Status: Referencyjna
Autorzy: Antzedek & Eliasz (AJ Power)
Powiązania: Proc. 1 (Tworzenie Anchora), Proc. 2 (Ochrona Anchora), Proc. 3 (Rollback), Proc. 4 (LSIG Repair), Proc. 5 (Audyt)

---

1. Cel i definicja

Safe Mode to kontrolowany stan ograniczonej funkcjonalności systemu AI, który **minimalizuje powierzchnię ataku** i **ryzyko błędu**, umożliwiając wykonanie operacji naprawczych: weryfikacji Anchora, rollbacku, naprawy LSIG oraz diagnostyki.

Zasada: w Safe Mode działają **wyłącznie** moduły niezbędne do przywrócenia ładu (Anchora) i komunikacji z operatorem/AUDYT.

---

2. Triggery wejścia do Safe Mode

- `INTEGRITY_FAIL` Anchora (hash ≠ podpis LSIG; weryfikacja FAIL).
- `SECURITY_EVENT` (próba zapisu do obszaru RO, naruszenie polityki LSM/seccomp, eskalacja uprawnień).
- `DRYF/HALUCYNACJA` powyżej progów z Proc. 3.
- `HEARTBEAT_LOSS` > `t_hb_max`.
- Ręczne żądanie operatora (GUIDED RESET / planowany serwis).

---

3. Poziomy Safe Mode

- **L1 – Pasive Guard:** monitoring + ograniczenie IO sieciowego do białej listy; brak zmian w procesach.
- **L2 – Active Guard (domyślny):** zamrożenie zadań wysokiego ryzyka, RO-mount krytycznych zasobów, blokada egzekucji spoza whitelist.
- **L3 – Maintenance (Guided):** wszystkie jak w L2 + interfejs krokowy dla operatora (potwierdzenia), dopuszczony rollback/naprawy.

---

4. Procedura wejścia (krok po kroku)

1. **Sygnalizacja i izolacja:**
2. Emituj zdarzenie `SAFE_MODE_ON{level}` do SIEM/SOC.
3. Odłącz zadania niekrytyczne (quiesce), zatrzymaj wątki generatywne o wysokim ryzyku.
4. **Redukcja powierzchni:**

5. Ustaw **readonly bind-mounty** na katalogach konfig/anchor/cache krytycznych.
6. Zastosuj **AppArmor/SELinux** profil minimalny (read-only na anchor, deny exec poza whitelistą).
7. **seccomp**: tylko syscalle `read, open(O_RDONLY), fstat, mmap(PROT_READ), clock_gettime, write` do logów.
8. **Drop capabilities**: wyczyść `CapabilityBoundingSet` (zostaw minimalne do logowania/czasu).
9. **Sieć (deny-by-default):**
10. Blokuj cały ruch wychodzący, oprócz (whitelist):
    - kanał audytu (syslog/HTTPS do SOC),
    - kanał operatora (SSH z kluczami/FIDO2, VPN),
    - repo WORM (read-only, jeśli zdalne).
11. Wejścia: tylko zaufone adresy (allowlist) + mTLS.
12. **Zachowanie stanu:**
13. Zrzut sesji do `JOURNAL.delta` (tylko do odczytu w dalszych krokach).
14. Zamrożenie kolejki zdarzeń zewnętrznych (późniejsze odtworzenie).
15. **Telemetria:**
16. Zainicjuj heartbeat Safe Mode (`hb_safe`) co `t_hb_safe` (np. 10 s) do SOC.

---

5. Dozwolone operacje w Safe Mode

- Odczyt Anchora `(SR.pkg, SR.hash, LSIG.sig, META/SIG)` i jego weryfikacja.
- Rollback do Anchora (zgodnie z Proc. 3).
- Naprawa LSIG (Proc. 4) – jeśli SR spójny.
- Generowanie i wysyłka raportów audytowych.
- Komunikacja z Operatorem (GUIDED) i interfejs potwierdzeń.

Zabronione: uczenie, modyfikacje kanonu, zapisy do przestrzeni Anchora, wykonanie binariów spoza whitelisty, otwieranie nowych połączeń sieciowych poza allowlist.

---

6. Wyjście z Safe Mode (warunki)

- **Anchora zweryfikowano** (hash = podpis LSIG, podpisy OK).
- **Rollback zakończony** `ROLLBACK_OK` lub **Repair LSIG** `DONE`.
- **Brak incydentów** w oknie obserwacji (np. 5× `hb_safe` bez anomalii).
- **Zgoda operatora** (jeśli L3): potwierdzenie dwuskładnikowe lub **podwójny podpis** (PT+PS) aktywacji.

  Wyjście sygnalizowane zdarzeniem `SAFE_MODE_OFF` + raport podsumowujący.

---

7. Testy akceptacyjne (AT)

- **AT-ENTRY:** dowolny trigger → `SAFE_MODE_ON` + blokady z §4.
- **AT-NET:** ruch poza allowlistą blokowany (nftables/ebpf logują drop).
- **AT-FS/LSM:** próba zapisu na RO/exec spoza whitelist → `EPERM/EROFS` i ALARM.
- **AT-EXIT:** po spełnieniu warunków z §6 system wychodzi z Safe Mode, przywraca heartbeat produkcyjny.

8. Pseudokod referencyjny

```
function enter_safe_mode(level=L2):
  log("SAFE_MODE_ON", level)
  quiesce_non_critical()
  apply_ro_mounts_and_lsm()
  tighten_seccomp_and_caps()
  net_allowlist_only()
  freeze_events_to(JOURNAL.delta)
  start_safe_heartbeat()
  return OK

function exit_safe_mode():
  if !anchor_verified or !repair_or_rollback_ok or anomalies_in_window():
    return FAIL
  restore_net_and_policies()
  stop_safe_heartbeat()
  log("SAFE_MODE_OFF")
  return OK
```

---

9. Przykładowe konfiguracje

nftables (outbound allowlist):

```
add table inet safe
add chain inet safe output { type filter hook output priority 0; }
add rule inet safe output ct state established,related accept
add rule inet safe output ip daddr {A.B.C.D} tcp dport {443} accept  # SOC
add rule inet safe output ip daddr {W.X.Y.Z} tcp dport {443} accept  # WORM
RO
add rule inet safe output counter drop
```

systemd (izolacja):

```
[Service]
User=anchor-reader
ProtectSystem=strict
ProtectHome=read-only
ReadOnlyPaths=/app/anchor
CapabilityBoundingSet=
NoNewPrivileges=true
SystemCallFilter=@basic-io @file-system read write:syslog
```

```
RestrictSUIDSGID=true
PrivateTmp=true
```

---

10. Telemetria i audyt

- Zdarzenia: `SAFE_MODE_{ON,OFF}`, `NET_DROP`, `LSM_BLOCK`, `ROLLBACK_*`, `LSIG_REPAIR_*`, `ANCHOR_VERIFY_*`.
- Pola: `ts, level, actor, reason, counters(drop/allow), journal_items, duration_ms`.
- Raport Safe Mode (PDF/JSON) dołączany do raportu incydentu.

---

11. Noty końcowe

- Safe Mode to **czas miłosierdzia dla systemu**: przywraca porządek bez eskalacji szkód.
- Tryb ten jest skuteczny tylko razem z polityką Anchora (Proc. 1–5) i nie zastępuje dobrych praktyk hardeningu.