Threat Model: Poisoning + Failure Scenarios

Version: 3.1.0
Philosophy: Assume adversarial environment. Plan for failure.

---

1. Graph Poisoning (Fake Mother Wallets)

Threat: Adversaries create fake wallet clusters that appear profitable to attract copy-traders, then dump on followers.

Detection Methods:
- Monitor new mother wallet discovery rate
- Check for unrealistic win rates (>90%)
- Analyze cluster correlation patterns
- Track sudden wallet activity spikes

System Response:

```
if new_mothers_24h > 10:
    ACTIVATE_GRAPH_KILL_SWITCH()
    log_event("MOTHER_EXPLOSION", count=new_mothers_24h)
```

Mitigation:
- 30-day confidence decay (half-life)
- Require minimum trade history before trusting
- Cross-reference with known poisoning patterns
- V2.0 fallback always available

---

2. CEX Funding Dead-Ends

Threat: Wallet graph terminates at CEX withdrawals, preventing mother wallet identification.

Detection Methods:
- Flag wallets with CEX blacklist matches
- Identify funding patterns typical of CEX (round amounts, timing)

System Response:

```
if is_cex_wallet(source):
    mark_as_cex(source)
    # Do not trace further up
    # Child wallet still tracked on own merit
```

Mitigation:
- Maintain comprehensive CEX address blacklist
- Track child wallets independently

- Use V2.0 scoring for CEX-funded wallets
- No graph boost for CEX-sourced wallets

---

3. Stale Data Scenarios

Threat: Acting on outdated price/liquidity data leads to losses.

Detection Methods:
- Compare signal timestamp to current time
- Monitor API response timestamps
- Track price volatility since signal

System Response:

```
if signal_age > FRESHNESS_LIMIT[asset_class]:
    VETO("STALE_SIGNAL")

if api_data_age > 30:  # seconds
    REFRESH_DATA()
    if still_stale:
        VETO("STALE_DATA")
```

Freshness Limits:

Asset Class	Max Age
meme_coin_low_cap	300s
established_altcoin	900s
major_crypto_cex	1800s

---

4. Time Drift

Threat: Server clock drift causes incorrect signal timing, cooldown miscalculation.

Detection Methods:
- Compare system time to NTP servers
- Log time discrepancies with API responses

System Response:

```
if abs(system_time - ntp_time) > 5:  # seconds
    log_warning("TIME_DRIFT", drift=delta)
    if drift > 30:
        PAUSE_TRADING()
        alert_operator("Critical time drift")
```

Mitigation:
- Use NTP synchronization
- Log timestamps from external APIs for comparison
- All internal times in UTC

5. RPC Rate Limits Hit

Threat: Exceeding rate limits causes missed signals or failed executions.

Detection Methods:
- Track request counts per minute
- Monitor 429 response codes
- Measure response latencies

System Response:

```
if rate_limit_hit:
    apply_exponential_backoff()
    switch_to_backup_rpc()
    if all_rpcs_limited:
        PAUSE_TRADING()
        alert_operator("RPC rate limits exhausted")
```

Mitigation:
- Implement token bucket rate limiter
- Maintain multiple RPC endpoints
- Prioritize execution over monitoring
- Cache repeated queries

---

6. Simulation Bypass Attempts

Threat: Malicious tokens detect simulation and behave differently (pass simulation, fail real trade).

Detection Methods:
- Compare simulation results to actual execution
- Track simulation accuracy over time
- Flag tokens with simulation/execution divergence

System Response:

```
if simulation_result != execution_result:
    log_event("SIMULATION_BYPASS_SUSPECTED", token=token)
    blacklist_token(token)
    update_simulator_accuracy(correct=False)
```

Mitigation:
- Use realistic simulation parameters
- Randomize simulation amounts
- Track tokens that pass sim but fail execution
- Require 95% accuracy before enabling Assassin

---

7. Human Override Failure Modes

Threat: Operator makes emotional decisions that bypass safety systems.

Detection Methods:
- Log all manual overrides
- Track override outcomes
- Compare manual vs automated performance

System Response:

```
if manual_override:
    require_confirmation("Are you sure? This bypasses safety.")
    log_event("MANUAL_OVERRIDE", reason=user_input)
    # V2.0 vetoes CANNOT be overridden
    if veto_reason in NON_OVERRIDABLE:
        REJECT_OVERRIDE()
```

Non-Overridable Vetoes:
- Spread > 3%
- Liquidity below minimum
- Tax > 10%
- Kill switch active

---

8. Wallet Tracking by Adversaries

Threat: Others track our execution wallets and front-run or copy our trades.

Detection Methods:
- Monitor for followers on execution wallets
- Track unusual activity patterns around our trades

System Response:

```
if followers_detected(wallet):
    rotate_wallet(wallet)
    log_event("WALLET_COMPROMISED", wallet=wallet)
```

Mitigation (Ghost Mode):
- Rotate 3-5 execution wallets
- Random delay jitter: 5-30ms
- Skip 10% of valid signals randomly
- Vary position sizes ±5%

---

9. Coordinated Pump-and-Dump Detection

Threat: Groups coordinate pumps to extract money from copy-traders.

Detection Methods:
- Multiple unrelated clusters buying same token simultaneously
- Sudden volume spike with no fundamental news
- Social media sentiment surge correlated with buys

System Response:

```
if coordinated_buy_detected(token):
    VETO("COORDINATED_PUMP_SUSPECTED")
    flag_token(token, "potential_pnd")
    increase_confidence_threshold(token, +0.2)
```

Indicators:
- >3 clusters buying within 5 minutes
- Volume >10x 24h average
- Social mentions >5x average

---

10. Database Corruption

Threat: SQLite database becomes corrupted, losing historical data.

Detection Methods:
- Integrity checks on startup
- Periodic consistency validation

System Response:

```
if db_integrity_check_failed:
    PAUSE_TRADING()
    attempt_recovery()
    if recovery_failed:
        restore_from_backup()
        alert_operator("DB restored from backup")
```

Mitigation:
- Daily automated backups
- WAL mode for crash recovery
- Verify backup integrity

---

11. API Key Compromise

Threat: API keys leaked, allowing attackers to drain rate limits or access data.

Detection Methods:
- Monitor for unusual API usage patterns
- Track API calls from unexpected IPs

System Response:

```
if suspicious_api_activity:
    ROTATE_ALL_KEYS()
    alert_operator("Potential key compromise")
    audit_access_logs()
```

Mitigation:
- Keys in `.env` file, not in code
- `.env` excluded from git
- Regular key rotation (monthly)
- Minimal permissions per key

Threat Summary Matrix

Threat	Likelihood	Impact	Detection	Auto-Response
Graph Poisoning	Medium	High	Kill switch triggers	Kill switch
CEX Dead-Ends	High	Low	Blacklist match	Mark & skip
Stale Data	Medium	Medium	Timestamp check	Veto signal
Time Drift	Low	Medium	NTP comparison	Pause trading
RPC Rate Limits	Medium	Medium	429 tracking	Backoff + switch
Simulation Bypass	Low	High	Accuracy tracking	Blacklist token
Human Override	Medium	Medium	Logging	Restrict overrides
Wallet Tracking	Medium	Medium	Follower detection	Rotate wallet
Coordinated P&D	Medium	High	Multi-cluster analysis	Veto + flag
DB Corruption	Low	High	Integrity checks	Restore backup
Key Compromise	Low	High	Usage monitoring	Rotate keys