Autoppia Web Agents Subnet (Subnet 36) - Repo Audit Report

Date: 2026-02-12

Repo: `autoppia_web_agents_subnet` (local workspace)

Revision reviewed: `opensource` @ `c975f6b` ("Enable file logging")

Executive Summary

- **Launch readiness**: **NOT READY** (critical security + functionality blockers).
- **Test status (current repo state)**: `pytest -q` (Python 3.13.5) => **105 failed, 99 passed, 3 warnings**.
- **Highest-risk areas**:
- Miner auth/blacklist logic (request filtering).
- Sandboxed miner execution (secrets exposure, gateway auth, network exposure).
- Distributed consensus commitment schema mismatch (consensus convergence likely broken).
- IWAP auth headers design (replayable signature if backend only checks static message signature).

Scope & Method

What was done:

- Static review of validator/miner entrypoints, sandbox execution path, gateway proxy, consensus, and IWAP client.
- Repository-wide search for risky patterns (subprocess, eval/exec, pickle, YAML loaders, HTTP usage).
- Ran unit/integration/property/performance tests via `pytest -q` in the current environment.

What was not done:

- No review of external backend services (IWAP API, IPFS operator) beyond how this repo calls them.
- No review of `autoppia_iwa` and `autoppia_webs_demo` source beyond the integration assumptions from this repo.

Architecture Overview (Open-Source Pipeline)

This repo implements (or claims to implement) the following runtime pipeline:

1 **Validator starts** (`neurons/validator.py`), initializes:

- Bittensor neuron base components (wallet, subtensor, metagraph).
- IWAP client integration (`autoppia_web_agents_subnet/platform/...`).
- Sandbox manager (`autoppia_web_agents_subnet/opensource/sandbox_manager.py`).

1 **Round start + handshake**:

- Validator computes season/round boundaries (`validator/round_start/mixin.py`, `validator/round_manager.py`).
- Validator sends `StartRoundSynapse` to miners and receives `agent_name` + `github_url` (`protocol.py`, `round_start/synapse_handler.py`).

1 **Miner evaluation**:

- For each miner `github_url`, validator clones repo and runs it as a containerized HTTP agent (`SandboxManager.deploy_agent()`).
- Validator evaluates the agent using IWA stateful evaluator (`validator/evaluation/stateful_cua_eval.py`).

1 **IWAP reporting**:

- Validator posts round start, miner registration, task set, and evaluation batches to IWAP (`platform/*`).

1 **Consensus + weights**:

- Validator uploads a score snapshot to IPFS and commits CID on-chain (`validator/settlement/consensus.py`).
- Validator aggregates other validators' commitments and sets WTA weights (`validator/settlement/mixin.py`).

Findings

Severity scale used:

- **Critical**: exploitable security flaw or guaranteed mainnet failure.
- **High**: likely failure in production or major security weakness.
- **Medium**: important hardening, operational risk, or correctness gaps.
- **Low**: cleanup / polish / best practice.

Critical

C-01: Miner request auth bypass hardcoded for UID 60

- **Where**: `autoppia_web_agents_subnet/base/miner.py:152-156`
- **What**: Any caller whose hotkey maps to UID `60` bypasses validator-permit and stake checks.
- **Impact**: A single chain identity can invoke miner endpoints without the intended authorization controls. This is a serious security issue and breaks the intended threat model.
- **Recommendation**: Remove the UID 60 special-case entirely. If you need a dev backdoor, gate it behind an explicit `TESTING=true` / `--mock` mode and never ship it enabled.

C-02: Distributed consensus commitment schema mismatch (commit vs aggregate vs docs)

- **Where**:

- Commit writes: `autoppia_web_agents_subnet/validator/settlement/consensus.py:129-147`
- Aggregation filtering: `autoppia_web_agents_subnet/validator/settlement/consensus.py:253-279`
- Docs show different schema: `docs/advanced/CONSENSUS_SYSTEM.md:165-172`
- **What**:
- Commit payload uses `"v": 5` and fields `r,se,te,c`.
- Aggregator expects `entry["v"] == CONSENSUS_VERSION` (default `1`) and also requires season field `"s"`.
- Documentation shows yet another schema (`v:4, e, pe, c, r`).
- **Impact**: Validators will likely skip each other's commitments => **no consensus convergence**, inconsistent winners, potential vTrust penalties, broken incentives.
- **Recommendation**: Define one canonical commitment schema and enforce it everywhere:
- Code commit writer
- Code aggregator filter
- Documentation
- Tests

C-03: Validator secrets injected into untrusted miner container

- **Where**: `autoppia_web_agents_subnet/opensource/sandbox_manager.py:141-145`
- **What**: The validator's `OPENAI_API_KEY` and `CHUTES_API_KEY` are exported into the miner's sandboxed agent container environment.
- **Impact**: Any miner code can read/exfiltrate these keys (via HTTP responses, logs, side channels). This is a hard launch blocker.
- **Recommendation**:
- Do not pass API keys into the agent container.
- If miners need LLM access, force all traffic through the gateway, and give the gateway the keys (not the agent).
- Apply strict egress policies: agent containers should not have direct internet access.

C-04: Sandbox gateway control-plane endpoint is unauthenticated and reachable by miner containers

- **Where**: `autoppia_web_agents_subnet/opensource/gateway/main.py:122-133`
- **What**: `/set-allowed-task-ids` has no auth. Miner containers can reach the gateway over the shared Docker network.
- **Impact**: A malicious miner can reset allowed task ids, disable usage tracking, or cause denial of service (break others, break cost limiting).
- **Recommendation**:
- Add an admin token (env var) and require it for this endpoint.

- Consider binding this endpoint to localhost only or a separate internal network not reachable by miners.

C-05: Sandboxed miner agent HTTP servers may be exposed publicly (random host port bind)

- **Where**: `autoppia_web_agents_subnet/opensource/sandbox_manager.py:147-158`
- **What**: `ports={f"{SANDBOX_AGENT_PORT}/tcp": None}` often publishes a random host port on `0.0.0.0`.
- **Impact**: Untrusted miner code becomes remotely reachable from the internet (depending on host firewall). This expands attack surface significantly.
- **Recommendation**: Bind explicitly to loopback only, e.g. publish to `("127.0.0.1", <random_or_allocated_port>)`, or avoid host publishing entirely and communicate over an isolated network.

High

H-01: Evaluation phase does not persist scores/evaluated state; settlement gating likely prevents weight updates

- **Where**:
- Evaluation loop: `autoppia_web_agents_subnet/validator/evaluation/mixin.py`
- Settlement gating: `autoppia_web_agents_subnet/validator/settlement/mixin.py:37`
- **What**: The evaluation phase does not:
- increment `agents_evaluated`,
- set `AgentInfo.evaluated = True`,
- set/store `AgentInfo.score` (final aggregate).
- **Impact**: Settlement may conclude “not all agents evaluated” and keep original weights; network incentives break.
- **Recommendation**: Track per-agent completion and compute a final score/avg reward, then set `evaluated=True` and store `score`.

H-02: Evaluation uses stale `current_block` for settlement cutoff

- **Where**: `autoppia_web_agents_subnet/validator/evaluation/mixin.py:32-48`
- **What**: `current_block` is read once at phase start and never refreshed, but it is used to decide whether to stop for settlement.
- **Impact**: The validator can run past settlement deadlines or stop too early, depending on drift.
- **Recommendation**: Refresh `current_block = self.block` inside the evaluation loop (or use `round_manager.get_status(self.block)`).

H-03: `dendrite_with_retries()` can crash or return None instead of a list

- **Where**: `autoppia_web_agents_subnet/utils/dendrite.py:59-63`
- **What**:
- Asserts all elements are non-None (`assert all(el is not None for el in res)`).

- On exception, it logs and exits without returning a list.
- **Impact**: Handshake can fail catastrophically (assertion error or `NoneType` usage), causing round failure.
- **Recommendation**:
- Remove the assert; return partial results and let caller handle missing miners.
- Ensure function always returns `List[T|None]`.

H-04: Gateway cost limiting and provider support is inconsistent / likely broken

- **Where**:
- Env name mismatch: validator exports `COST_LIMIT_VALUE`, gateway expects `COST_LIMIT_PER_TASK` (`sandbox_manager.py:93-97`, `gateway/config.py:5-6`)
- Chutes unsupported in provider config: `gateway/models.py:35-45`
- Usage parsing defaults to 10k tokens if missing: `gateway/main.py:62-66`
- **Impact**: Cost limiting may not work as intended; chutes path likely fails; usage will be wildly inaccurate.
- **Recommendation**:
- Standardize env variable names.
- Implement provider configs for all supported providers.
- Parse actual OpenAI usage fields (typically `prompt_tokens`, `completion_tokens`, `total_tokens`).

H-05: IWAP auth headers appear replayable (static message signature)

- **Where**:
- Header signing: `autoppia_web_agents_subnet/platform/utils/iwa_core.py:153-167`
- Message source: `autoppia_web_agents_subnet/validator/config.py:66` and `autoppia_web_agents_subnet/platform/mixin.py:40`
- **What**: The signature is over a static message, and there is no nonce/timestamp/request-body binding in this repo.
- **Impact**: If the backend accepts this as auth, then captured headers can be replayed to impersonate a validator.
- **Recommendation**: Use per-request signing (include timestamp + request hash), and enforce freshness on the backend.

H-06: Validator config can `sys.exit(1)` at import-time

- **Where**: `autoppia_web_agents_subnet/validator/config.py:113-123`
- **What**: `validate_config()` runs at import-time and exits if env vars are missing.
- **Impact**: Breaks test tooling and makes library imports unsafe; encourages “works only in one runtime mode”.
- **Recommendation**: Validate on actual runtime entry (e.g. validator __`main`__), not on module import. Provide explicit `validate_config()` calls.

H-07: Version check on startup has no timeout

- **Where**: `autoppia_web_agents_subnet/base/neuron.py:270-273`
- **What**: `requests.get(version_url)` can hang.
- **Impact**: Startup stalls; operational instability.
- **Recommendation**: Add a small timeout (e.g. 2-5s) and handle failures gracefully.

H-08: IPFS HTTP API default is plain HTTP and payloads are not verified

- **Where**:
- Default URL is HTTP: `autoppia_web_agents_subnet/validator/config.py:69`
- Aggregation fetches payloads without verification: `validator/settlement/consensus.py:298-334`
- **Impact**: MITM on IPFS API traffic can inject fake payloads for a CID fetch (unless CID verification is performed). This can corrupt consensus.
- **Recommendation**:
- Prefer HTTPS gateways.
- Verify fetched bytes against CID (or use a trusted local IPFS node with authenticated transport).

Medium

M-01: Dependency supply-chain risk (unpinned requirements)

- **Where**: `requirements.txt`
- **What**: Dependencies are unpinned.
- **Impact**: Non-reproducible installs; unexpected breakage; harder security posture (no lock/hashes).
- **Recommendation**: Pin versions and generate a lock file (pip-tools / Poetry / uv), optionally with hash-checking.

M-02: Docker hardening gaps (root user, no digest pins, no read-only FS)

- **Where**:
- Gateway Dockerfile: `autoppia_web_agents_subnet/opensource/gateway/Dockerfile`
- Sandbox Dockerfile: `autoppia_web_agents_subnet/opensource/sandbox/Dockerfile`
- **Impact**: If miner code escapes app constraints, it runs as root inside container. Hardening is weak.
- **Recommendation**: Use non-root user, read-only filesystem, drop capabilities, memory/CPU limits, seccomp/apparmor profiles.

M-03: Repo size limits are checked after cloning

- **Where**: `autoppia_web_agents_subnet/opensource/utils_git.py:154-169`
- **Impact**: A large repo can still fill disk during clone (even shallow).
- **Recommendation**: Add pre-clone constraints where possible (e.g. GitHub API size check) or run clone into a quota-limited filesystem.

M-04: Auto-update script uses destructive reset

- **Where**: `scripts/validator/update/auto_update_deploy.sh:93`
- **Impact**: Can destroy local changes and break running systems if misused.
- **Recommendation**: Make destructive steps opt-in, or warn loudly. Consider using tags/releases instead of tracking `origin/main`.

M-05: Operator doc references scripts that are missing in this repo

- **Where**: `Agents.md:129-185`
- **Impact**: Operators will fail following documented procedures; confusion pre-launch.
- **Recommendation**: Either add the scripts or update docs to match current reality.

M-06: Reporting resend script imports a module that does not exist

- **Where**: `scripts/validator/reporting/resend_report.py:30`
- **Impact**: Script doesn’t work.
- **Recommendation**: Fix import path or add the missing reporting module.

Low

L-01: `setup.py` classifiers list Python 3.8-3.10 but `python_requires >=3.11`

- **Where**: `setup.py:54-77`
- **Impact**: Packaging metadata inconsistency.
- **Recommendation**: Align classifiers with supported versions.

Test Suite Notes (Why It Matters)

The current test suite is a strong signal for launch readiness because it encodes expected behavior of the validator pipeline. As of 2026-02-12:

- `pytest -q` => **105 failed**
- Early failures include:
- ImportError from `validator/evaluation/stateful_cua_eval.py` due to missing stubs in `tests/conftest.py`.
- RoundManager boundary logic now requires `season_start_block` to be set; several tests call `sync_boundaries()` directly without setting it.
- Settlement tests show gating issues when `agents_on_first_handshake` is unset/invalid.

Recommendation: treat “tests green” as a hard gate before launch.

Minimum Launch Checklist (Suggested)

Security:

- Remove UID 60 bypass (`C-01`).
- Stop injecting validator API keys into miner containers (`C-03`).
- Authenticate gateway control endpoints and lock down gateway reachability (`C-04`).

- Ensure miner agent containers are not publicly reachable (`C-05`).
- Fix IWAP auth to be non-replayable (`H-05`).
- Ensure IPFS fetch is integrity-checked and transport is secure (`H-08`).

Correctness:

- Fix consensus commitment schema and aggregation (`C-02`).
- Make evaluation produce persistent scores and set `evaluated=True` (`H-01`).
- Fix evaluation settlement cutoff timing (`H-02`).
- Fix dendrite retry helper to return consistent results (`H-03`).

Release/ops:

- Fix broken docs and missing scripts (`M-05`, `M-06`).
- Add CI (`.github/workflows`) to run tests and static checks on PRs.