Bloch-SIS-PoW — Consolidated Security Report

A single-document roll-up of every security review performed on the Bloch-SIS-PoW codebase to date. 2026-07-22.

Read this first. Bloch-SIS-PoW is an **unaudited mainnet beta**. It has **not** been audited by a third party. The reviews below are an **internal adversarial audit** plus an **automated open-source scanner pipeline** — they raise the floor, they do not certify safety. Consensus logic and cryptographic parameter choices require human + external review. The native-contract features (bloch-euvm / Ustav / Kirpich) **go live before an external audit exists** and are **not safe until one does — use is entirely at the user's own risk.** BLCH is not a security and carries no value claim. The network is 51%-attackable at current hashrate. Designed ≠ built ≠ booted.

1. Executive summary

REVIEW	SCOPE	VERDICT	CRITICALS	HIGHS	STATUS
Internal adversarial audit	bloch-euvm eUTXO VM (consensus-critical)	Proceed to engineering; hard-block on activation until fixed	0	2	**All fixed** ✓
Automated scanner pipeline	L1 Rust workspace (node + crates)	Green with tracked exceptions	0 (live-exploitable)	2 (transitive DoS, upstream-blocked)	Fixed / tracked
EVM / L2	Rust revm scaffold (no Solidity deployed)	Scanned as Rust; Solidity tooling pre-configured	—	—	Design-only

Bottom line. No live-exploitable critical exists. The two internal-audit HIGHs (state-commitment binding, gas metering) are **remediated with regression tests**. The two scanner HIGHs are **transitive DNS-library DoS advisories with no upstream fix yet** (blocked on libp2p), tracked and mitigated. A **third-party audit remains mandatory** before any consensus activation of native contracts.

2. Internal adversarial audit — bloch-euvm

The native eUTXO contract VM (the execution substrate for the Ustav token standard) received a dedicated adversarial audit across 8 dimensions: determinism, panics/overflow, gas/DoS, value/supply conservation, state-proof soundness, module-compiler bypass, batcher/AMM safety, activation/feature-off. Every finding carries a passing repro test (`crates/bloch-euvm/tests/audit_*.rs`).

#	SEV	FINDING	FIX	STATUS
F1	HIGH	Block commitment bound a 36-byte scalar summary, not the eUTXO state — two blocks with different token movements produced byte-identical commitments	Commit `SparseMerkleTree::root()` over the resulting UTXO set; gas/fee carried as bound side-data	**FIXED** ✓
F2	HIGH	Flat gas schedule — a 1-byte and an 8-MB hash both cost 60 gas; `Dup` amplified ~50 MiB per ~1000 gas → machine-dependent block acceptance = consensus split when wired	Gas ∝ operand byte length (`GAS_PER_BYTE` / `HASH_GAS_PER_BYTE`) + hard ceilings (per-operand 1 MiB, per-program 1 MiB, total-alloc 64 MiB, tx-resource limits), fail-closed	**FIXED** ✓

#	SEV	FINDING	FIX	STATUS
F3	MED + X	`[profile.release]` lacked `overflow-checks` → release wraps while debug/test panics = mixed-profile validator divergence; plus unchecked arithmetic at flagged sites	`checked_add` / `checked_mul` at the sites + `overflow-checks = true` **mandated** for consensus builds	**FIXED** ✓
F4–F6	LOW	Mint ordering, batcher reserve saturation, governance-compile edge cases (Pick-depth truncation, threshold==0)	Canonical ordering, `checked_add` reserves, fail-closed governance compile	**FIXED** ✓

Post-remediation: 330 tests pass (debug **and** release parity). The crate stays isolated / feature-off / not consensus-wired. Activation remains gated on a **third-party audit + a coordinated height-gated hard fork**. On top of the VM, **Kirpich** adds 23 deterministic, fail-closed KRP charter-audit rules that refuse to compile a token contract with any blocking defect.

3. Automated scanner pipeline — L1 Rust

TOOL	RESULT
cargo-audit (RustSec)	Green with the documented `--ignore` set; the only real vulns are the two transitive hickory-proto DoS advisories (below)
cargo-deny (advisories · licenses · bans)	**advisories ok · licenses ok · bans ok.** License policy fixed to allow the six first-party AGPL crates while still denying third-party copyleft
cargo-geiger	`unsafe` surface inventoried across the tree
Clippy (hardened)	pedantic + `arithmetic_side_effects` + no `unwrap` / `expect` in consensus paths
Miri	consensus/serialization suites under UB detection (nightly)
cargo-fuzz	10 targets: block/netmsg/handshake/pow/merkle/ghostdag/mempool/sig parse + SHA-256d PoW verify; **OSS-Fuzz** application prepared
proptest	deterministic ordering, blue-score monotonicity, emission conservation

⚠️ Open finding — tracked, not dismissed

RUSTSEC-2026-0118 / 0119 — hickory-proto 0.25.2 (DNS library): a DoS unbounded loop (NSEC3) and O(n²) name-compression CPU exhaustion. **No upstream fix reachable:** the fix is in hickory 0.26.x, but **libp2p 0.56.0 pins 0.25.2** via `libp2p-dns` + `libp2p-mdns`. **Reachability:** `/dns4/` multiaddr resolution + LAN-only mDNS — the node's canonical peering uses `/ip4/` addresses (no DNS) and DNSSEC is not enabled, so practical exposure is a malicious resolver / hostile LAN. **Action:** remove the two ignores the moment libp2p repins hickory ≥ 0.26. Accepted-and-tracked, not hidden.

Other advisories are **unmaintained-notices** (vendored PQClean `pqcrypto-*`, frozen by design; SP1/zk host-side toolchain crates that never touch the consensus/P2P runtime) — full disposition in `deny.toml` / `audit.toml`.

4. EVM / L2

The "EVM parallel chains" are **not deployed Solidity contracts** — the L2 is a Rust **revm-based scaffold** (`bloch-protocol/l2/*`, design-only, testnet/zero-value). It is scanned with the L1 Rust suite. The Solidity toolchain (Slither, Aderyn, Mythril, Echidna/Medusa, Foundry, Halmos, Semgrep, solhint) is **pre-configured for when real contracts exist** — no contracts were fabricated to scan. The **Rust↔EVM bridge** is the designated highest-risk component; its threat model (message replay, cross-side PQ-signature verification, finality inheritance from a 51%-attackable base, total-supply conservation) is documented for the future L2.

5. What tooling cannot certify — the human/third-party gate

Automated scanning catches dependency vulnerabilities, unsafe usage, panics, and memory/DoS classes. It does **not** verify: consensus logic (GhostDAG ordering, reorg-depth safety, fork mechanics), cryptographic parameter choices (the hybrid PQ construction, SHAKE-256 domain separation, the PoW regime), the bridge, or low-hashrate economic security. **A third-party audit is required before any consensus activation.**

6. Continuous posture

`.gitlab-ci.yml` (+ `.github/workflows/security.yml` for the GitHub mirror) run the scanners on every change, blocking on high-severity vulns / license failures and reporting pedantic lints. `SECURITY_TOOLING.md` documents each tool + local-run commands. OSV-Scanner / Dependabot-style monitoring watches for the hickory fix so the tracked exception can be lifted.
