Yuriy Khomenkov

SMART CONTRACT SECURITY RESEARCHER

yuriykhomenkov@gmail.com · khomdev.org · x.com/YKhomenkov · github.com/KhomDev · hackenproof.com/hackers/KhomDev

PROFILE

Independent smart contract security researcher auditing Solidity/EVM and Rust/Move protocols. 20 paid findings across 50 programs on HackenProof — 3 High, 17 Low, leaderboard rank #208, 607 reputation — every one accepted by the team that owned the code, plus a public credit from Ledgity Yield for a private disclosure. I pin scope to a commit, model the invariants a protocol depends on, and report only what I have proven with a test against the real contracts. Available for private audits.

20	3	#208	50
PAID FINDINGS	HIGH SEVERITY	LEADERBOARD RANK	PROGRAMS HUNTED

EXPERIENCE

Independent Smart Contract Security Researcher

May 2026 – Present

KhomDev · Independent practice · Remote

- Hunt logic, access-control and accounting flaws in live DeFi code; 20 paid findings across 50 programs on HackenProof (3 High, 17 Low), rank #208 of the platform leaderboard.
- Run audits end to end: scope pinned to a commit, deployed bytecode checked against it, value flows and invariants mapped, then attacked by hand review plus a fuzz harness written for that codebase.
- Report to one standard: root cause at the exact line and commit, preconditions, attack path, broken invariant, a PoC asserting the full balance change, and the smallest safe fix with a regression test — followed by a fix review.
- Keep an open rejection log: 113 rejected reports classified by the earliest check that would have caught them, turned into two open-source tools that check my own submissions before they are sent.
- Publish the method: 25 technical articles on khomdev.org/research, each number backed by a public repository.

Protocol Engineer, Vyper (self-directed)

Dec 2025 – Aug 2026 (public repositories from May 2026)

FLYWHEEL · four composed DeFi protocols · gitlab.com/khomdev-group

- Built in Vyper 0.4 and deployed on Ethereum Sepolia: a CDP stablecoin with a peg keeper, a tranched multi-strategy ERC-4626 vault, ve-governance with gauges and bribes, and a batch-auction intent DEX (EIP-712, ERC-1271). About 9,200 lines and 700+ tests; Hypothesis stateful fuzzing caught a rounding bug in my own vault. Testnet only, unaudited, labelled so.
- Published vyper-scales: three offline security tools (seam coverage, compiler-upgrade diff, signature matrix) and three long-form articles.

SELECTED FINDINGS

Fixed findings that are public. Severity as rated by the program; a private disclosure carries no rating.

CREDITED · YIELD ACCOUNTING — Nov 2026

Ledgity Yield, a live RWA yield protocol on six chains (about $2.1M TVL). Its migration path skipped the one-day maturity fee deposits pay, so entering a second before midnight took a day of yield from other holders. Reported privately with a Foundry PoC; Ledgity made its legacy tokens withdrawal-only on every chain and credited the report in its AUDIT.md. Case study.

HIGH · BROKEN ACCESS CONTROL — Aug 2026

At the contract level any account could assign itself the admin permission. An off-chain check blocked it in the deployed system, so the program rated it High rather than Critical. Fixed. Shared with 60 other reporters.

HIGH · COLLATERAL ACCOUNTING — Oct 2026

A self-backed draw in a stablecoin vault system permanently blocked surplus distribution. Proven with a PoC on the real entry points; fixed by the team. Another researcher had reported it first; the analysis and PoC are mine. Case study.

OPEN-SOURCE SECURITY TOOLING

Author and maintainer · MIT licensed, offline, no account or API key required.

realism-gate · Rust CLI, v1.4.0

Does the proof prove the claim? Reads Foundry, Anchor, Solana Rust, Sui and Aptos Move PoCs (20 rules) and the report that cites them, and marks each claim shown, not shown, contradicted or not checkable. Claim binding was right 91% of the time on its first run over 120 public PoC + report pairs. JSON, SARIF and a GitHub Action.

priorart · Rust CLI, v0.2.2

Is it new, in scope, and payable? Builds an offline dossier from a program's known issues, prior audits, 14 classes of exclusion clause and payout tiers, and lists what to read before the PoC, with quotes and page numbers. Ships retrieval-only until a fresh held-out split clears my 30% false-alarm bar. Includes my 113-rejection dataset (CC BY 4.0).

OPEN-SOURCE GUIDES

Rugproof · five-part series, v1.0.0 · MIT / CC BY 4.0
From Rust fundamentals to Solana security, with exercises, vulnerable versus fixed code, and case studies of Wormhole, Cashio, Crema and other public exploits.

TECHNICAL SKILLS

LANGUAGES	Solidity, Vyper, Rust, Move (Sui, Aptos), Cairo, Python, TypeScript
ECOSYSTEMS	EVM L1 and L2, Solana, Sui, Aptos, Starknet
TOOLING	Foundry, Echidna, Medusa, Halmos, Slither, Semgrep, solana-program-test, LiteSVM, Mollusk
PROTOCOL DOMAINS	Lending, vaults and yield, stablecoins, AMMs and hooks, staking, bridges, governance, oracles
REPORTING	HackenProof, Immunefi, Cantina, Code4rena, Sherlock; severity argumentation, PoC, fix review

AUDIT METHOD

Five steps, each ending in something the client can check.

01 **Pin the scope** — repo, commit and file list agreed in writing; live contracts read on chain and flagged where they differ from the commit.

02 **Read what is known** — prior audits, open issues, tests and docs; forks diffed against upstream so hours go to changed code.

03 **Model the protocol** — where value sits, who can move it, and the invariants that must hold — every documentation claim becomes a target.

04 **Attack two ways** — a fuzz harness written for the codebase runs while every function's assumptions are checked by hand.

05 **Prove, challenge, report** — each finding gets a PoC on real entry points, passes realism-gate, survives the strongest counter-argument, then fixes are reviewed.

RESEARCH AND WRITING

25 articles on bug hunting, proof quality and why reports fail, with the public repositories behind the numbers — full list at khomdev.org/research. Selected:

— *I measured my own audit tools against real exploit proofs, fixed what broke, and measured again* (25 Nov 2026) — judged PoCs read 1/12 → 25/28; fresh precision 10/31

— *A passing PoC is not a proof* (30 Oct 2026) — 69% of 886 exploit PoCs assert nothing

— *The duplicate ceiling* (28 Oct 2026) — 41 of 113 rejections died at the duplicate check

PRIVATE AUDIT ENGAGEMENTS

Available for private audits across Solidity, Vyper, Rust, Move and Cairo. Standard scope and deliverables per engagement:

- **Coverage map** — what was fuzzed, what was read, and what the review did not reach.
- **Ruled-out leads** — every serious lead that died, with the reason it died.
- **The fuzz harness** — invariant tests written for your codebase, ready to run in CI.
- **Findings to one standard** — root cause at the exact line and commit, preconditions, attack path, broken invariant, a PoC asserting the full balance change, and the smallest safe fix with a regression test.

Scope is quoted per engagement once I have read the code. If nothing is found I say so plainly — the coverage map and ruled-out leads are still delivered. Request an audit →