Security Research Labs

Astar Collective proxy pallet Audit

Hacking assessment report

V1.0, November 4th, 2024

Haroon Basheer haroon@srlabs.de

Regina Biro regina@srlabs.de

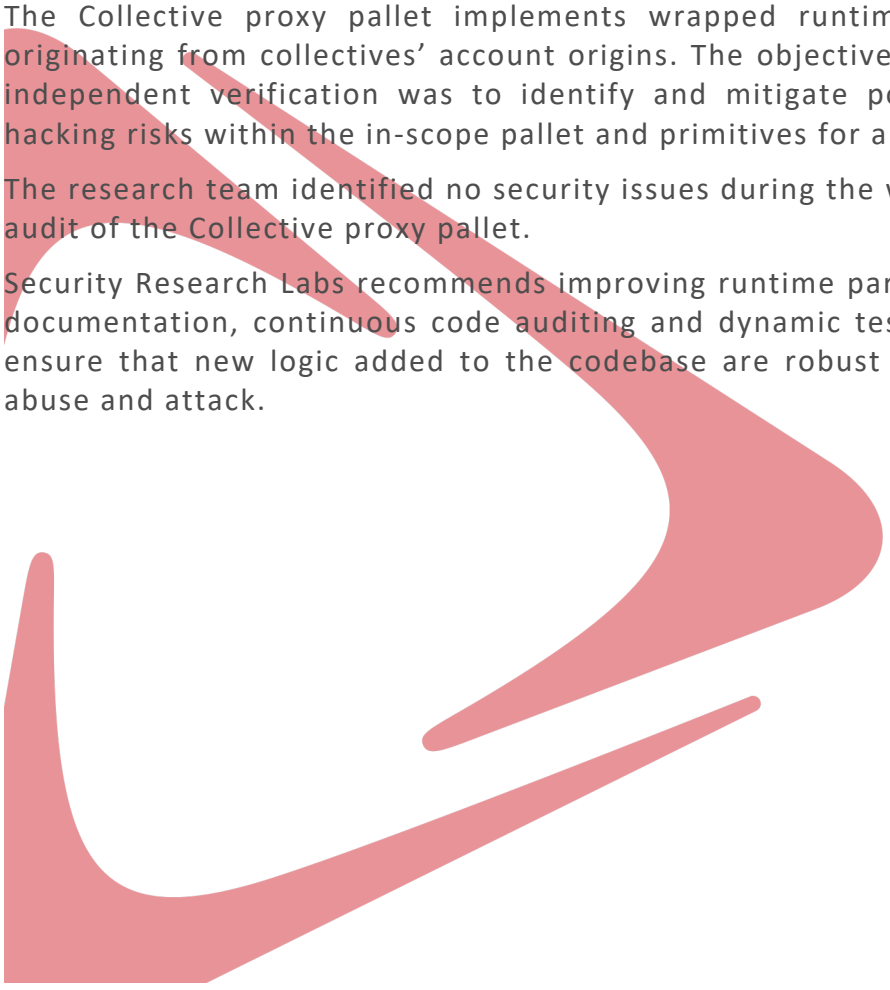
Abstract. This work describes the result of a thorough and independent security assurance audit of the Astar Collective-proxy pallet by Security Research Labs. Security Research Labs is a consulting firm that has been providing specialized audit services for Substrate-based blockchains since 2019, including in the Polkadot ecosystem.

Between October 28, 2024, and November 4, 2024, Security Research Labs conducted a comprehensive security assurance audit of Collective proxy pallet. During the audit, Astar provided sufficient support and access to relevant documentation.

The Collective proxy pallet implements wrapped runtime calls originating from collectives' account origins. The objective of this independent verification was to identify and mitigate potential hacking risks within the in-scope pallet and primitives for abuse.

The research team identified no security issues during the week of audit of the Collective proxy pallet.

Security Research Labs recommends improving runtime parameter documentation, continuous code auditing and dynamic testing to ensure that new logic added to the codebase are robust against abuse and attack.



Content

1 Disclaimer

This report describes the findings and core conclusions derived from the audit carried out by Security Research Labs within the agreed-on timeframe and scope as detailed in Chapter 2. Please note that this report does not guarantee that all existing security vulnerabilities were discovered in the codebase exhaustively and that following all evolution suggestions described in Chapter 4 may not ensure all future code to be bug free.

2 Motivation and scope

This security assurance audit evaluated the Astar's Collective-proxy [1] pallet from an attacker's viewpoint to identify and understand potential business logic misconfigurations and technical vulnerabilities, with the objective of strengthening its security posture.

The Collectives pallet which is part of the on-chain governance, allows to vote on proposals and execute extrinsics calls. The Collective-proxy pallet provides proxy functionality to the accounts of collective origin and wraps runtime extrinsics on behalf of the collective account origins. The pallet at the time of the audit was enabled in Astar's testnet runtime (Shibuya) and on its local runtime.

Security Research Labs (SRLabs), supported by the Astar team, investigated each aspect, strongly focusing on issues that may affect the business logic, reputation, or security model of pallet. The employed audit methodology included a comprehensive threat-model driven manual code audit. This in-depth approach ensures that potential vulnerabilities and weaknesses are thoroughly identified and addressed.

3 Baseline Assurance

3.1 Findings summary

During the analysis of the collective proxy pallet and related runtime configurations, Security Research Labs identified no security issues.

4 Evolution suggestions

To ensure that all future pallets, primitives, and runtimes are secure against known and yet undiscovered threats alike, the auditors recommend the following best practices:

Document the runtime parameter for approval threshold. Both Shibuya and the local runtime have different approval thresholds for internal testing purposes. We recommend updating the final 2/3 approval threshold for the Astar production runtime in the technical design [2] documentation. This will create clarity for users and potential future security audits on the runtime parameters related to the Collective proxy pallet configuration.

Regular code review and continuous fuzz testing. Regular code reviews are recommended to avoid introducing new logic or arithmetic bugs, while continuous fuzz testing can identify potential vulnerabilities early in the development process. Ideally, Astar should continuously fuzz their code on each commit made to the codebase. The substrate-runtime-fuzzer [3] (which uses Ziggy [4], a fuzzer management tool) can be a good starting point.

5 Bibliography

[1] [Online]. Available: https://github.com/AstarNetwork/Astar/tree/master/pallets/collective-proxy.

[2] [Online]. Available: https://github.com/AstarNetwork/astar-docs/blob/main/docs/learn/governance/technical_guide.md.

[3] [Online]. Available: https://github.com/srlabs/substrate-runtime-fuzzer.

[4] [Online]. Available: https://github.com/srlabs/ziggy.