



SMART CONTRACT SECURITY AUDIT

Bako Safe

Multisig Contracts

CLIENT

Bako Safe

PLATFORM

Fuel

LANGUAGE

Sway

REVIEW

September 2024

DOCUMENT

Document information

Project	Bako Safe, Multisig Contracts
Client	Bako Safe
Platform	Fuel
Language	Sway
Repository	Not recorded in the original report
Commit	Not recorded in the original report
Methods	Static and dynamic analysis, manual code review
Audit period	September 2024

Document history

VERSION	DATE	CHANGE
1.0	Not recorded	Original report issued View original report
1.1	October 2026	Reformatted into the Safe Edges standard template. Findings unchanged

This report was reformatted into the Safe Edges standard template in October 2026. Findings, severities, recommendations and code references are unchanged from the original report, which is preserved unchanged in the [Safe Edges Audit Portfolio on GitHub](#).

CONTENTS

Contents

01	Executive summary	3
02	Scope	4
03	Methodology	5
04	Severity and status	6
05	Findings summary	7
06	Detailed findings	8
	I-01 Deprecated Usage of ZERO_B256 and Address::from(ZERO_B256)	8
07	Functional test scenarios	9
08	Automated testing	9
09	Final summary	9
	Disclaimer	10

Executive summary

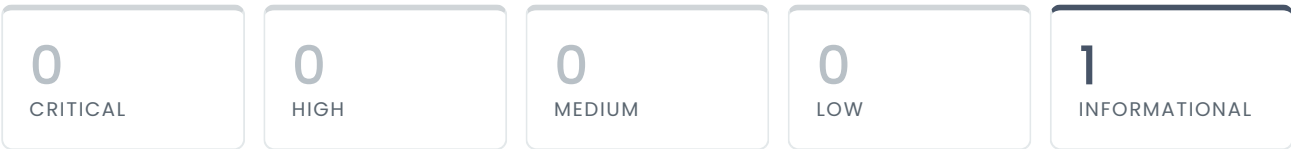
About Bako Safe

Bako Safe is a multisig wallet on the Fuel network. This audit aimed to assess the security and functionality of the Bako Safe contracts.

Summary of the review

Safe Edges reviewed the Bako Safe contracts in September 2024 using static and dynamic analysis along with manual code review. One Informational issue was identified, covering deprecated Sway constants, and the development team resolved it. Functional test scenarios for multisig signing passed, and automated testing found no major issues.

Findings by severity



Findings by status

STATUS	ISSUES
Resolved	1

Scope

Repository	Not recorded in the original report
Commit	Not recorded in the original report
Platform	Fuel
Language	Sway

Only the code at the commit above was reviewed. Changes made after this commit are not covered by this report.

Methodology

Methods recorded for this engagement: Static and dynamic analysis, manual code review.

Manual review

Auditors read the in-scope code line by line, tracing control flow, state changes and trust boundaries against the intended behaviour described by the client.

Threat modelling

We map privileged roles, external dependencies and value flows, then test the assumptions the code makes about each of them.

Functional and automated testing

Where tooling exists for the language and chain, we run static analysis and targeted tests to confirm suspected issues and check edge cases.

Fix review

After the client addresses the findings, we review the updated code and record the final status of each issue.

Severity and status

Severity levels

SEVERITY	DEFINITION
Critical	Direct loss or permanent lock of funds, or full compromise of the protocol, with few or no preconditions. Must be fixed before deployment.
High	Significant loss of funds, broken core functionality or manipulation of critical state under realistic conditions. Should be fixed before deployment.
Medium	Limited loss or degraded functionality, or issues that need specific conditions or privileged access to exploit. Should be fixed.
Low	Minor impact, low likelihood, or deviations from best practice. Fix when practical.
Informational	Code quality, documentation, gas and style observations with no direct security impact.

Status definitions

STATUS	DEFINITION
Resolved	The issue was fixed and the fix was verified.
Partially resolved	The risk was reduced but not fully removed.
Acknowledged	The client accepted the issue without changing the code.
Open	The issue was not addressed at the time of the report.
Not recorded	The original report did not record a resolution status for this issue.

Findings summary

ID	TITLE	SEVERITY	STATUS
I-01	Deprecated Usage of ZERO_B256 and Address::from(ZERO_B256)	Informational	● Resolved

Detailed findings

I-01 Deprecated Usage of ZERO_B256 and Address::from(ZERO_B256)

Severity Informational

Status ● Resolved

DESCRIPTION

The use of `ZERO_B256` and `Address::from(ZERO_B256)` is deprecated as per the updated Sway language standards.

RECOMMENDATION

- Replace `ZERO_B256` with `b256::zero()`.
- Replace `Address::from(ZERO_B256)` with `Address::zero()`.

LOCATION

- `configurables`
- `validations.sw`

07

Functional test scenarios

The following scenarios were tested and passed:

- Multisig wallet using the same address (e.g., 3/4 signatures)
- Validations for zero address usage in signing
- Signature validity for other addresses

08

Automated testing

No major issues were found during automated testing. Some false positives were reported but deemed non-impactful. All relevant issues were categorized appropriately.

09

Final summary

The audit for Bako Safe revealed no critical or high severity issues. Informational-level improvements were suggested and have been resolved by the development team.

Disclaimer

This report is based on the code at the commit listed in the Scope section and on information the client provided during the review. It reflects Safe Edges' assessment at the time of the review and is not a guarantee that the code is free of vulnerabilities.

Changes made after the reviewed commit, deployment configuration, off-chain components and third-party dependencies are outside the scope of this report unless stated otherwise.

Security is ongoing. We recommend continued testing, monitoring and bug bounty coverage alongside any audit.

This report is not investment advice and is not an endorsement of the project, its team or any token.



Safe Edges is a Web3 security firm and official security partner for the Fuel, ENI and XI EcoChain ecosystems. We work across smart contract audits, Rust and WASM reviews, ZK circuit review, AI and MCP security, and penetration testing.

200+

PROJECTS AUDITED

\$4B+

TVL SECURED

0

POST-AUDIT HACKS