



SMART CONTRACT SECURITY AUDIT

RedStone

Fuel SDK

CLIENT

RedStone

PLATFORM

Fuel

LANGUAGE

Sway

REVIEW

September 2024

DOCUMENT

Document information

Project	RedStone, Fuel SDK
Client	RedStone
Platform	Fuel
Language	Sway
Repository	Redstone Fuel SDK
Commit	afc708b91f46afabcaa6428b0836e2231f6eb985
Methods	Not recorded in the original report
Review	24 August to 5 September 2024

Document history

VERSION	DATE	CHANGE
1.0	September 2024	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
	H-01 Lack of Validation for Empty feed_ids Allows Oracle Timestamp Manipulation	8
	H-02 Missing SRC-5 Ownership Standard Implementation	8
	L-01 SRC-2 Documentation Not Followed	8
	L-02 Redundant join() Function on Bytes	9
	L-03 Commented Out Code Left in Production	9
	L-04 Custom FromBytes Trait Instead of From<Bytes>	9
	L-05 Unnecessary Payload Struct	9
	L-06 Use of Magic Numbers in Timestamp Logic	10
	L-07 Near Duplicate Functions Across Multiple Files	10
07	Conclusion	11
	Disclaimer	12

Executive summary

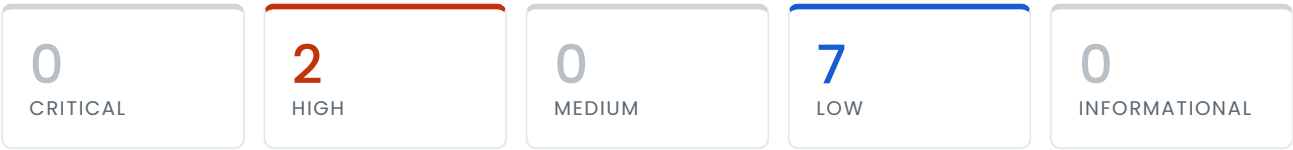
About RedStone

An oracle protocol that enables ultra-low latency connections between data providers and both on-chain and off-chain applications. This audit focuses on the integrity of the price feed mechanism and the timestamp validation system implemented via Fuel smart contracts.

Summary of the review

Safe Edges reviewed RedStone's Fuel SDK between 24 August and 5 September 2024, focusing on the integrity of the price feed mechanism and the timestamp validation implemented in Sway. Nine issues were identified: two High and seven Low. The most significant, H-01, allows an updater to submit an empty `feed_ids` vector and advance the oracle timestamp without updating prices, so stale prices can be reported as fresh.

Findings by severity



Findings by status

STATUS	ISSUES
● Not recorded	9

Scope

Repository	Redstone Fuel SDK
Commit	afc708b91f46afabcaa6428b0836e2231f6eb985
Platform	Fuel
Language	Sway

IN-SCOPE FILES

```
src/contract/src/prices.sw
src/core/config_validation.sw
src/core/processor.sw
src/protocol/data_point.sw
src/utils/bytes.sw
```

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: Not recorded in the original report.

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
H-01	Lack of Validation for Empty <code>feed_ids</code> Allows Oracle Timestamp Manipulation	High	● Not recorded
H-02	Missing SRC-5 Ownership Standard Implementation	High	● Not recorded
L-01	SRC-2 Documentation Not Followed	Low	● Not recorded
L-02	Redundant <code>join()</code> Function on Bytes	Low	● Not recorded
L-03	Commented Out Code Left in Production	Low	● Not recorded
L-04	Custom <code>FromBytes</code> Trait Instead of <code>From<Bytes></code>	Low	● Not recorded
L-05	Unnecessary <code>Payload</code> Struct	Low	● Not recorded
L-06	Use of Magic Numbers in Timestamp Logic	Low	● Not recorded
L-07	Near Duplicate Functions Across Multiple Files	Low	● Not recorded

Detailed findings

H-01 Lack of Validation for Empty `feed_ids` Allows Oracle Timestamp Manipulation

Severity **High** Status ● Not recorded Location `prices.sw — write_prices()`

DESCRIPTION

An updater can submit an empty `feed_ids` vector with forged payloads to maliciously update only the timestamp, effectively freezing future legitimate updates.

IMPACT

Oracle reports stale prices as fresh.

RECOMMENDATION

Add `assert(feed_ids.len() > 0);` and enforce signer verification early.

H-02 Missing SRC-5 Ownership Standard Implementation

Severity **High** Status ● Not recorded Location `prices.sw`

DESCRIPTION

Ownership logic is manually implemented. This increases the risk of inconsistencies and errors.

RECOMMENDATION

Use standardized SRC-5 ownership module for clarity and maintainability.

L-01 SRC-2 Documentation Not Followed

Severity **Low** Status ● Not recorded Location `processor.sw`

DESCRIPTION

Parameters in function doc-comments are missing or unordered.

RECOMMENDATION

Follow SRC-2 formatting.

L-02 Redundant `join()` Function on Bytes

Severity Low Status ● Not recorded Location `bytes.sw`

DESCRIPTION

Custom `join()` duplicates native `append()` behavior.

RECOMMENDATION

Replace `join()` with `append()` to simplify logic.

L-03 Commented Out Code Left in Production

Severity Low Status ● Not recorded Location `config_validation.sw`, line 45

DESCRIPTION

Signer validation is commented out, allowing unregistered payloads to pass.

RECOMMENDATION

Uncomment `revert(SIGNER_NOT_RECOGNIZED + index);`

L-04 Custom `FromBytes` Trait Instead of `From<Bytes>`

Severity Low Status ● Not recorded Location `data_point.sw`

DESCRIPTION

Custom parsing logic adds unnecessary complexity.

RECOMMENDATION

Use the standard `From<Bytes>` trait for better readability and compatibility.

L-05 Unnecessary `Payload Struct`

Severity Low Status ● Not recorded Location `payload.sw`

DESCRIPTION

Struct wraps `Vec<DataPackage>` but adds no extra utility.

RECOMMENDATION

Replace with `type Payload = Vec<DataPackage>;`

L-06 Use of Magic Numbers in Timestamp Logic

Severity Low Status ● Not recorded Location `config.sw`

DESCRIPTION

Code includes unexplained expressions like `10 + (1 << 62)`.

RECOMMENDATION

Replace with named constants and add explanatory comments.

L-07 Near Duplicate Functions Across Multiple Files

Severity Low Status ● Not recorded Location `sample.sw, config.sw`

DESCRIPTION

Identical logic (`get_u64(...)`) is copied across files.

RECOMMENDATION

Abstract and reuse common utility logic to reduce redundancy.

Conclusion

This review was conducted by Safe Edges with a focus on logical correctness, adherence to Fuel standards (SRC), and prevention of critical oracle manipulation. While no critical vulnerabilities were found, timestamp drift logic and signer handling should be corrected before production.

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