

# SafeEdges Audit Report

**Reviewed by:** SafeEdges  
**Prepared for:** RedStone  
**Review Dates:** 24/08/2024 – 05/09/2024

## Protocol Summary

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.

## Scope

**Repository:** [Redstone Fuel SDK](#)  
**Review Commit Hash:** [afc708b91f46afabcaa6428b0836e2231f6eb985](#)





**In-Scope Contracts:**






- [src/contract/src/prices.sw](#)
- [src/core/config\\_validation.sw](#)
- [src/core/processor.sw](#)
- [src/protocol/data\\_point.sw](#)
- [src/utils/bytes.sw](#)

**Deployment Chain:**

- The **Fuel Blockchain**

## Vulnerability Summary

Identifier	Title	Severity	Details
H1	Lack of Validation for Empty <a href="#">feed_ids</a> Allows Oracle Timestamp Manipulation	 High	<a href="#">View Details</a>
H2	Missing SRC-5 Ownership Standard Implementation	 High	<a href="#">View Details</a>
L1	SRC-2 Documentation Not Followed	 Low	<a href="#">View Details</a>
L2	Redundant <a href="#">join()</a> Function on <a href="#">Bytes</a>	 Low	<a href="#">View Details</a>

Identifier	Title	Severity	Details
L3	Commented Out Code Left in Production	 Low	<a href="#">View Details</a>
L4	Custom <code>FromBytes</code> Trait Instead of <code>From&lt;Bytes&gt;</code>	 Low	<a href="#">View Details</a>
L5	Unnecessary <code>Payload</code> Struct	 Low	<a href="#">View Details</a>
L6	Use of Magic Numbers in Timestamp Logic	 Low	<a href="#">View Details</a>
L7	Near Duplicate Functions ( <code>get_u64</code> ) Across Multiple Files	 Low	<a href="#">View Details</a>

## Detailed Findings

### H1. Lack of Validation for Empty `feed_ids` Allows Oracle Timestamp Manipulation

- **Severity:** High
- **Location:** `prices.sw` — `write_prices()`
- **Issue:** 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.

### M1. Missing SRC-5 Ownership Standard Implementation

- **Severity:** High
- **Location:** `prices.sw`
- **Issue:** Ownership logic is manually implemented. This increases the risk of inconsistencies and errors.
- **Recommendation:** Use standardized SRC-5 ownership module for clarity and maintainability.

### L1. SRC-2 Documentation Not Followed

- **Severity:** Low
- **Location:** `processor.sw`
- **Issue:** Parameters in function doc-comments are missing or unordered.
- **Recommendation:** Follow [SRC-2](#) formatting.

### L2. Redundant `join()` Function on `Bytes`

- **Severity:** Low
- **Location:** `bytes.sw`
- **Issue:** Custom `join()` duplicates native `append()` behavior.
- **Recommendation:** Replace `join()` with `append()` to simplify logic.

---

### L3. Commented Out Code Left in Production

- **Severity:** Low
  - **Location:** `config_validation.sw`, line 45
  - **Issue:** Signer validation is commented out, allowing unregistered payloads to pass.
  - **Recommendation:** Uncomment `revert(SIGNER_NOT_RECOGNIZED + index);`
- 

### L4. Custom `FromBytes` Trait Instead of `From<Bytes>`

- **Severity:** Low
  - **Location:** `data_point.sw`
  - **Issue:** Custom parsing logic adds unnecessary complexity.
  - **Recommendation:** Use the standard `From<Bytes>` trait for better readability and compatibility.
- 

### L5. Unnecessary `Payload` Struct

- **Severity:** Low
  - **Location:** `payload.sw`
  - **Issue:** Struct wraps `Vec<DataPackage>` but adds no extra utility.
  - **Recommendation:** Replace with `type Payload = Vec<DataPackage>;`
- 

### L6. Use of Magic Numbers in Timestamp Logic

- **Severity:** Low
  - **Location:** `config.sw`
  - **Issue:** Code includes unexplained expressions like `10 + (1 << 62)`.
  - **Recommendation:** Replace with named constants and add explanatory comments.
- 

### L7. Near Duplicate Functions Across Multiple Files

- **Severity:** Low
  - **Location:** `sample.sw`, `config.sw`
  - **Issue:** Identical logic (`get_u64(...)`) is copied across files.
  - **Recommendation:** Abstract and reuse common utility logic to reduce redundancy.
- 

## ☒ Final Notes

This review was conducted by SafeEdges 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.

#### Contact:

For further auditing support or verification: <https://safeedges.in>