



coinspect
You build, we defend.



Smart Contract Audit
FTSOv2 Custom Feeds

February 2025



FTSOv2 Custom Feeds Diff Smart Contract Audit

Version: v250210

Prepared for: Flare

February 2025

Security Assessment

1. Executive Summary
2. Summary of Findings
 - 2.3 Solved issues & recommendations
3. Scope
4. Assessment
 - 4.1 Security assumptions
 - 4.2 Decentralization
 - 4.3 Code quality
5. Detailed Findings
 - CUS-002 - Governance can prevent feed access by renaming to non-existing feed ID

CUS-003 - Unchecked list parameters length



CUS-004 - Governance can break feed ID renames
by removing custom feeds pointed by changes

6. Disclaimer

1. Executive Summary

In **February 2025**, **Flare** engaged **Coinspect** to perform a Smart Contract Audit of updates to the FTS0v2 contract and related changes.

These updates primarily involve the implementation of custom feeds and, most notably, the functionality to manage these feeds through governance. For further details, refer to the *Assessment* section.

 Solved	 Caution Advised	 Resolution Pending
High 0	High 0	High 0
Medium 0	Medium 0	Medium 0
Low 1	Low 0	Low 0
No Risk 2	No Risk 0	No Risk 0
Total 3	Total 0	Total 0

During this differential review, Coinspect identified a low-risk issue where the governance can rename a feed using the feed ID of a non-existent feed.

2. Summary of Findings

This section provides a concise overview of all the findings in the report grouped by remediation status and sorted by estimated total risk.

2.3 Solved issues & recommendations

These issues have been fully fixed or represent recommendations that could improve the long-term security posture of the project.

Id	Title	Risk
CUS-002	Governance can prevent feed access by renaming to non-existing feed ID	Low
CUS-003	Unchecked list parameters length	None
CUS-004	Governance can break feed ID renames by removing custom feeds pointed by changes	None

3. Scope

The scope was set to be the diff changes from the repository at <https://gitlab.com/flarenetwork/flare-smart-contracts-v2> between commits **0cce14ac345cbaa0ad5b9dafb0cf9a179bda4db6** (main branch) and **e07cab2e354abdabcf10532309b6368b48234ca6**.

Coinspect primarily focused on reviewing the differential changes on the smart contracts, excluding unchanged contracts from the analysis. Similarly, the `sFLR` contract, which is invoked by the `SFlrCustomFeed` contract, was not included in the review scope.

4. Assessment

The `FtsoV2` contract now includes a **custom feed** type that merges **fast-update feeds** with predefined values from external contracts. Governance can now add, remove, update, and rename feeds, enabling flexibility in response to token renames (e.g., MATIC → POL). The first implementation of this feature, **SFlrCustomFeed**, was introduced for **Staked FLR (sFLR)** and reviewed accordingly.

The `FtsoV2` contract is implemented as a **UUPSUpgradeable proxy**, primarily to accommodate projects that hardcode the FTSO address, preventing breakage when addresses change. However, since the `FlareContractRegistry` already provides a way to query the latest `FtsoV2` address, introducing a proxy arguably increases the attack surface unnecessarily. Since the upgrade logic resides in the implementation contract, which extends the `GovernedProxyImplementation`, the authorized caller (governance) is set to `ZERO_ADDRESS` in the constructor to prevent malicious parties from upgrading the implementation.

It is worth mentioning that before the renaming takes effect (e.g., MATIC → POL), the intended process is to remove the MATIC feed first, and then add the POL feed—note this process is not enforced by the in-scope contracts. Therefore, the `getSupportedFeedIds` function would only return current, non-remapped feed IDs, as per the initial design decision, while historical mappings remain accessible via `getFeedIdChanges`. If another rename occurs (e.g., POL → NEW_POL), the same process is applied, ensuring that changes only reference the latest feed, while governance remaps old IDs accordingly.

Given this design, Coinspect recommends implementing an orchestrator contract to handle removals, additions, and renames in a single atomic call. This would prevent inconsistencies caused by the unordered execution of administrative tasks, such as attempting to remove a fast updates feed that is still referenced by an existing `FeedIdChangeData`.

4.1 Security assumptions

The following security assumptions were considered:

- Custom feeds are not controlled by malicious actors, as this could interfere with data retrieval when querying multiple feed IDs.
- The governance could be manipulated into approving a malicious request, so special attention was given to identifying missing validations that could

compromise the contract's integrity or data consistency.

4.2 Decentralization

Two entities can now manage the FTS0v2 contract:

- **Address Updater** (existing role), which now has the additional ability to update the `FeeCalculator` contract address.
- **Governance Contract**, which can add, remove, and replace custom feeds by updating the registry while keeping the feed ID unchanged. It can also rename feeds by storing a mapping between the old and new feed IDs. Additionally, it has the authority to upgrade the contract implementation when needed.



4.3 Code quality

The changes were straightforward and well-documented, supported by a comprehensive set of unit tests. However, the code contains multiple loops iterating over lists. While this is not an issue at present, Flare should monitor this as the number of feeds grows to prevent potential performance bottlenecks.

5. Detailed Findings

CUS-002

Governance can prevent feed access by renaming to non-existing feed ID

Status Solved	Risk Low
	
Resolution Fixed	Impact Medium Likelihood Low

Location

`./contracts/protocol/implementation/FtsoV2.sol`

Description

The governance can rename a feed using the ID of a non-existent feed, causing a Denial-of-Service (DoS) for users attempting to access the renamed feed. This issue arises due to the lack of validation ensuring that `newFeedId` corresponds to an existing feed.

The FtsoV2 contract allows feed renames by storing a `FeedIdChangeData` object, which is referenced through the original feed ID (`oldFeedId`) in the `changeFeedIds` function. The struct is defined as follows:

```
struct FeedIdChangeData {
    bytes21 newFeedId;
    uint88 index; // index is 1-based, 0 means non-existent
}
```

If `newFeedId` does not correspond to an existing feed, users attempting to access `oldFeedId` will be unable to retrieve feed information.

Additionally, Coinspect identified that the `changeFeedIds` function lacks validation to ensure that custom feed IDs include a valid feed ID. However, the Flare team stated this is an intended behavior as it is possible that some custom feed could be added to fast updates at some point.

Recommendation



Validate that `newFeedId` belongs to an existing feed before allowing the renaming process.

Status

Fixed in commit [e597ab1c8ddd5c8d9b2c21321c7212f33e3a04b1](#). The FtsoV2 contract now includes a validation check to ensure that the feed referenced by `newFeedId` exists.

CUS-003

Unchecked list parameters length

Status Solved	Risk None
	
Resolution Acknowledged	Impact Recommendation
	Likelihood -

Location

ntracts/protocol/implementation/FtsoV2.sol:633"

Description

The `_updateContractAddresses` function assumes that the provided hashes and addresses maintain the same index alignment but does not verify whether the number of hashes matches the number of addresses. While the function will revert if it cannot resolve an address, a discrepancy in the number of elements between these parameters suggests a potential issue, as it could result in incorrect address associations.

```
function _updateContractAddresses(  
    bytes32[] memory _contractNameHashes,  
    address[] memory _contractAddresses  
)
```

Recommendation


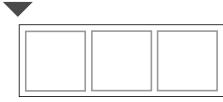
Validate that the number of hashes matches the number of addresses.

Status

Acknowledged. Flare mentioned that the `updateContractAddresses` function is only meant to be called by an instance of the `AddressUpdater` contract, though the `AddressUpdatable` contract functions can be called by any address configured as the address updater.

CUS-004

Governance can break feed ID renames by removing custom feeds pointed by changes

Status Solved	Risk None
	
Resolution Acknowledged	Impact Recommendation
	Likelihood -

Location

`./contracts/protocol/implementation/FtsoV2.sol`

Description

Governance can nullify a feed ID change (rename) by removing the target feed. This is possible due to the `removeCustomFeeds` function, which does not verify whether the feed is referenced in a `FeedIdChangeData` object.

This issue was classified as informational, as removing the feed ultimately has the same effect as this issue.

Recommendation

Verify or remove feed ID changes linked to the feed to be removed before proceeding.

Status

Acknowledged.

6. Disclaimer

The contents of this report are provided "as is" without warranty of any kind. Coinspect is not responsible for any consequences of using the information contained herein.

This report represents a point-in-time and time-boxed evaluation conducted within a specific timeframe and scope agreed upon with the client. The assessment's findings and recommendations are based on the information, source code, and systems access provided by the client during the review period.

The assessment's findings should not be considered an exhaustive list of all potential security issues. This report does not cover out-of-scope components that may interact with the analyzed system, nor does it assess the operational security of the organization that developed and deployed the system.

This report does not imply ongoing security monitoring or guaranteeing the current security status of the assessed system. Due to the dynamic nature of information security threats, new vulnerabilities may emerge after the assessment period.

This report should not be considered an endorsement or disapproval of any project or team. It does not provide investment advice and should not be used to make investment decisions.