

Smart contract audit

Tokos - Fallback Oracle

Content

Project description	3
Executive summary	3
Reviews	3
Scope	3
Technical analysis and findings	4
Security findings	5
**** Critical	5
*** High	5
H01 - Insufficient Input Validation in setAssetSources	5
** Medium	5
M01 - Lack of Stale Price Check Allows Use of Outdated, Inaccurate Prices	5
* Low	6
Informational	6
Risk section	7
Approach and methodology	8



Project description

This security audit focuses exclusively on the FallbackOracle smart contract. The contract is designed to function as a secondary price feed for the Aave V3 protocol using DIA oracles on the Somnia network. It implements the IPriceOracleGetter interface and provides asset prices by querying specified Chainlink-compatible aggregators. Its purpose is to enhance the resilience of the Aave V3 price mechanism by acting as a reliable fallback in case the primary oracle source fails. Contract administration, including the mapping of assets to their price feed sources, is restricted to an owner role.

Executive summary

Type	Utility contracts
Languages	Solidity
Methods	Architecture Review, Manual Review, Unit Testing, Functional Testing, Automated Review
Documentation	README.md
Repository	https://github.com/arenas-fi/arenas-contracts

Reviews

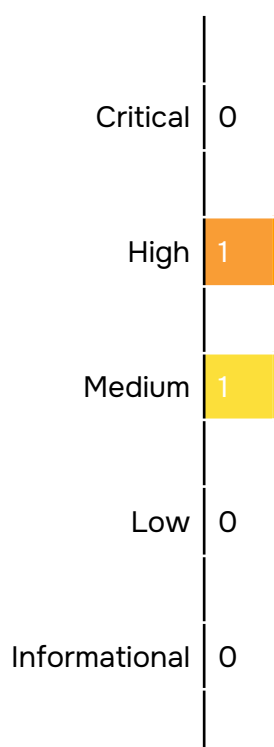
Date	Commit
02/10/2025	066b159ef3701a8177c2a6bad77cb8bbe61c6109
06/10/2025	ff604e18836c8cd4786634514a690bd12f85d846

Scope

File Name
contracts/FallbackOracle.sol



Technical analysis and findings



Security findings

**** Critical

No critical severity issue found.

*** High

H01 - Insufficient Input Validation in setAssetSources

Impact	High
Likelihood	Medium

The setAssetSources function in the FallbackOracle contract is responsible for mapping asset addresses to their corresponding price feed (aggregator) contracts. The function iterates through the assets_ and sources_ arrays but does not include a check to prevent a source address from being the zero address (address(0)). This allows the contract owner to inadvertently or intentionally set a null price feed for an asset.

Path: contracts/FallbackOracle.sol: setAssetSources()

Recommendation: Add a require statement inside the for-loop within the setAssetSources function to ensure that no address in the sources_ array is the zero address.

Found in: 066b159

Status: Fixed in ff604e1

** Medium

M01 - Lack of Stale Price Check Allows Use of Outdated, Inaccurate Prices

Impact	High
Likelihood	Medium



The `getAssetPrice` function retrieves price data by calling `latestAnswer()` on the configured DIA aggregator. While the function correctly checks if the returned price is greater than zero, it fails to check when that price was last updated.

A price feed could become "stale" if the off-chain nodes that power it stop reporting updates. If the Aave protocol's primary oracle fails and calls this contract as a fallback, the `FallbackOracle` would blindly trust and return this stale price. Operating on outdated price data during volatile market conditions is extremely dangerous and could lead to unfair liquidations or the issuance of severely under-collateralized loans, exposing the entire protocol to significant financial risk.

Path: `contracts/FallbackOracle.sol: getAssetPrice()`

Recommendation: Modify the `getAssetPrice` function to use DIA's `latestRoundData()` function, which returns both the price and an `updatedAt` timestamp. Add a `require` statement to ensure that `block.timestamp - updatedAt` does not exceed a configurable maximum threshold (e.g., 24 hour). The transaction should revert if the price is found to be stale.

Found in: 066b159

Status: Fixed in ff604e1

*** Low**

No low severity issue found.

Informational

No informational severity issue found.



Risk section

Centralization of Price Feed Administration: As detailed in finding H01, the system's security model is centralized around a single owner account. This creates a single point of failure. Beyond malicious action resulting from a compromised key, this also introduces operational risks, such as the loss of the private key, which would render the oracle's configuration immutable and unable to adapt to future market or protocol needs.




Implicit Trust in External Oracles: The security of the FallbackOracle is entirely dependent on the integrity and availability of the upstream DIA price feeds it is configured to consume. The contract itself is merely a wrapper; it implicitly trusts that the data from the configured aggregators is accurate. Any vulnerability or manipulation of a designated DIA feed would be passed directly through this oracle and into the Aave protocol.



Approach and methodology




To establish a uniform evaluation, we define the following terminology in accordance with the OWASP Risk Rating

Methodology:

	Likelihood Indicates the probability of a specific vulnerability being discovered and exploited in real-world scenarios
	Impact Measures the technical loss and business repercussions resulting from a successful attack
	Severity Reflects the comprehensive magnitude of the risk, combining both the probability of occurrence (likelihood) and the extent of potential consequences (impact)

Likelihood and impact are divided into three levels: High H, Medium M, and Low L. The severity of a risk is a blend of these two factors, leading to its classification into one of four tiers: Critical, High, Medium, or Low.

When we identify an issue, our approach may include deploying contracts on our private testnet for validation through testing. Where necessary, we might also create a Proof of Concept PoC to demonstrate potential exploitability. In particular, we perform the audit according to the following procedure:

	Advanced DeFi Scrutiny We further review business logics, examine system operations, and place DeFi-related aspects under scrutiny to uncover possible pitfalls and/or bugs
	Semantic Consistency Checks We then manually check the logic of implemented smart contracts and compare with the description in the white paper.
	Security Analysis The process begins with a comprehensive examination of the system to gain a deep understanding of its internal mechanisms, identifying any irregularities and potential weak spots.

