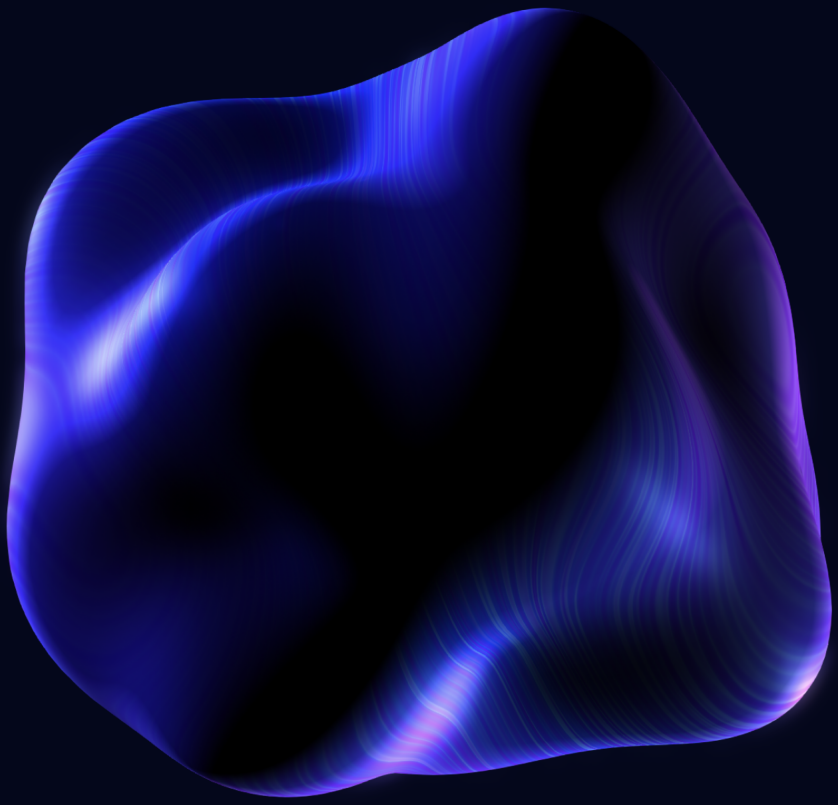
MUNDUS
SECURITY

Security Smart Contract Audit
Segment Finance Oracle Provider



MUNDUS
SECURITY

Segment Finance Oracle Provider security audit

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.
The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed — upon a decision of the Customer.

Reference information

Name	Segment Finance Oracle Provider
Language	Solidity
Network	Superchain Ecosystem
Website	https://www.segment.finance/
Documentation	https://docs.segment.finance/
Repository	https://github.com/Segment-Finance/protocol
Initial audit commit	48725d8ecec760cf83270ac10c592e4e8c059c40

MUNDUS
SECURITY

Findings summary

Findings statistics

Severity	Number	Left acknowledged
High	0	0
Medium	0	0
Low	1	1
Informational	1	1
Gas	1	1
Total	**3**	**3**

Finding Severity breakdown

All vulnerabilities discovered during the source code audit are classified based on their potential severity and have the following classification:

Severity	Description
High	Bugs that can trigger a contract failure or theft of assets. Further recovery is possible only by manual modification of the contract state or replacement of the contract.
Medium	Bugs that can break the intended contract logic or expose it to DoS attacks, but do not cause direct loss of funds.
Low	Bugs that do not pose significant danger to the project or its users but are recommended to be fixed nonetheless.
Informational	All other non-essential recommendations.
Gas	Gas optimization recommendations.

MUNDUS
SECURITY

Project description

Segment Finance Oracle Provider contract is the price feed provider for the Segment Finance lending and borrowing protocol in the Superchain Ecosystem.

Scope of work

Path
packages/oracle/contracts/interfaces/OracleInterface.sol
packages/oracle/contracts/provider/adapters/IOracleFeedAdapter.sol
packages/oracle/contracts/provider/OracleProvider.sol

Findings

ID	Severity	Description	Status
01	Low	owner local variable declaration shadows inherited property	ack.
02	Informational	Public function getPrice should be declared external	ack.
03	Gas	Array length should be cached outside of loop	ack.

Source code audit

ID-01. Low: owner local variable declaration shadows inherited property

Description

The **OracleProvider** contract's constructor (L52) contains owner local variable declaration. This declaration shadows the owner() function inherited from the **Ownable** contract and may cause errors during development.

Recommendation

Rename the local variable owner in the **OracleProvider** contract's constructor.

Alleviation

The Segment Finance team acknowledges this issue.

ID-02. Informational: Public function getPrice should be declared external

Description

The getPrice function (L61) of the **OracleProvider** contract is declared public but is never used from within the contract. Thus it can be declared external for better readability.

Recommendation

Declare the getPrice of the **OracleProvider** contract external instead of public.

Alleviation

The Segment Finance team acknowledges this issue.

ID-03. Gas: Array length should be cached outside of loop

Description

The updateTokens function (L101) of the **OracleProvider** contract contains a for-loop with uncached length. The length should be cached for gas optimization.

Recommendation

Cache the configs array length outside of the loop in the updateTokens function of the **OracleProvider** contract.

Alleviation

The Segment Finance team acknowledges this issue.

Disclaimers

Mundus disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical disclaimers

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.