

Inverter Issuance Token

Final Audit Report

July 29, 2024



Team Omega

Teamomega.eth.limo

Summary	2
Scope of the Audit	2
Resolution	3
Methods Used	3
Disclaimer	3
Severity definitions	4
Findings	5
General	5
G1. Remove unused imports [resolved]	5
G2. Mark _decimals as immutable [resolved]	5

Summary

Inverter has asked Team Omega to audit the contracts that define the behavior of their contracts.

We found **no high severity issues** (issues that can lead to a loss of funds, and are essential to fix) and we classified **no** issues as “medium” or low. **2** issues were classified as “info” - we believe the code would improve if these issues were addressed as well.

Severity	Number of issues	Number of resolved issues
High	0	0
Medium	0	0
Low	0	0
Info	2	2

Scope of the Audit

The scope of the audit concerns a single ERC20 contract. The code in scope developed here:

`https://github.com/InverterNetwork/contracts/blob/dev/src/external/token/ERC20Issuance_v1.sol`

Resolution

The issues were addressed in commit `8245bc10aa6026ac7f1993e1d609c08683ab16a0`

We checked the changes and updated the report.

Methods Used

Code Review

We manually inspected the source code to identify potential security flaws.

The contracts were compiled, deployed, and tested in a test environment.

Automatic analysis

We have used static analysis tools to detect common potential vulnerabilities. The tools have detected a number of low severity issues, concerning mostly the variables naming and external calls, were found.

We have included any relevant issues below in the appropriate parts of the report.

Disclaimer

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only.

Severity definitions

High	Vulnerabilities that can lead to loss of assets, or data manipulations.
Medium	Vulnerabilities that are essential to fix, but that do not lead to assets loss or data manipulations
Low	Issues that do not represent direct exploit, such as poor implementations, deviations from best practice, high gas costs, etc
Info	Matters of opinion

Findings

General

This contract is a straightforward extension of OpenZeppelin's `ERC20Capped` and `Ownable` contracts and allows for the definition of minter roles and allowances.

G1. Remove unused imports [resolved]

The imports of `Context` and `SafeERC20` imports are unused and can be removed.

Severity: Info

Resolution: The issue was resolved, the unused imports were removed from the code.

G2. Mark `_decimals` as immutable [resolved]

The `_decimals` state variable is set in the constructor and cannot be changed afterwards, and so can be marked `immutable`.

Severity: Info

Resolution: The issue was resolved, the `_decimals` state variable is now marked as `immutable`.