# CANTINA

# OpenVM
## Security Review

Cantina Managed review by:

**Zigtur**, Lead Security Researcher
**Om Parikh**, Security Researcher

August 16, 2025

# Contents

# 1 Introduction

## 1.1 About Cantina

Cantina is a security services marketplace that connects top security researchers and solutions with clients. Learn more at cantina.xyz

## 1.2 Disclaimer

Cantina Managed provides a detailed evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While Cantina Managed endeavors to identify and disclose all potential security issues, it cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities that were absent during the initial review. Therefore, any changes made to the code require a new security review to ensure that the code remains secure. Please be advised that the Cantina Managed security review is not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

## 1.3 Risk assessment

| Severity | Description |
| --- | --- |
| **Critical** | *Must* fix as soon as possible (if already deployed). |
| **High** | Leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users. |
| **Medium** | Global losses <10% or losses to only a subset of users, but still unacceptable. |
| **Low** | Losses will be annoying but bearable. Applies to things like griefing attacks that can be easily repaired or even gas inefficiencies. |
| **Gas Optimization** | Suggestions around gas saving practices. |
| **Informational** | Suggestions around best practices or readability. |

### 1.3.1 Severity Classification

The severity of security issues found during the security review is categorized based on the above table. Critical findings have a high likelihood of being exploited and must be addressed immediately. High findings are almost certain to occur, easy to perform, or not easy but highly incentivized thus must be fixed as soon as possible.

Medium findings are conditionally possible or incentivized but are still relatively likely to occur and should be addressed. Low findings a rare combination of circumstances to exploit, or offer little to no incentive to exploit but are recommended to be addressed.

Lastly, some findings might represent objective improvements that should be addressed but do not impact the project's overall security (Gas and Informational findings).

# 2   Security Review Summary

OpenVM is a performant and modular zkVM framework built for customization and extensibility.

From Aug 10th to Aug 13th the Cantina team conducted a review of openvm on commit hash dbc5005c. The team identified a total of **2** issues:

**Issues Found**

| Severity | Count | Fixed | Acknowledged |
|---|---|---|---|
| Critical Risk | 0 | 0 | 0 |
| High Risk | 0 | 0 | 0 |
| Medium Risk | 0 | 0 | 0 |
| Low Risk | 0 | 0 | 0 |
| Gas Optimizations | 0 | 0 | 0 |
| Informational | 2 | 2 | 0 |
| **Total** | **2** | **2** | **0** |

# 3 Findings

## 3.1 Informational

### 3.1.1 `ISA.md` variables name are different than the codebase

**Severity:** Informational

**Context:** ISA.md#L34-L35

**Description:** In commit 20c494e, two variable names have been modified in the `ISA.md` file: `ADDR_SPACE_-OFFSET` and `addr_space_height`. However, it is confusing as these new names do not correspond to the variable used in the codebase.

**Recommendation:** Consider reverting to the previous naming that matches with the codebase.

**OpenVM:** Fixed in `feat/new-execution` branch linked to PR 1567.

**Cantina Managed:** Fixed. The variables in the code have been renamed.

### 3.1.2 `alloc` will be deprecated in the future

**Severity:** Informational

**Context:** alloc.rs#L24

**Description:** The `alloc::alloc` function is called to allocate memory. This function's documentation indicates that this function is expected to be deprecated when the `allocate` function and the `Allocate trait` are stable.

> This function is expected to be deprecated in favor of the allocate method of the Global type when it and the Allocator trait become stable.

**Recommendation:** Consider monitoring future stable release to upgrade to `allocate` function when it is stable.

**OpenVM:** Fixed in PR 1981.

**Cantina Managed:** Fixed. A `TODO` comment has been added to keep track of this potential deprecation.