OpenVM
Security Review

Solo review by:
Zigtur, Lead Security Researcher

October 4, 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

A security review is 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 the review 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 a 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 level	Impact: High	Impact: Medium	Impact: Low
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

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 are 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 Sep 28th to Sep 29th the security researchers conducted a review of openvm on commit hash 13362dc6. The review focused on the following scope:

- PR 2150 to fix constant alignment.
- PR 2152 to change the nightly version string to a newer rust version.

A total of **1** issues were identified:

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	1	0	1
Total	**1**	**0**	**1**

3 Findings

3.1 Informational

3.1.1 `max_block_size` will never be used

Severity: Informational

Context: lib.rs#L976-L993

Description: The `max_block_size` value is initialized to `4`. Then, depending on the number of items, it is set to either `16` or `32`. The only case for which the initial `4` value is kept is when no item is found. This is never met in practice:

```
let mut max_block_size = 4; // @audit default value will never be used

for (mod_idx, item) in items.into_iter().enumerate() {
    let modulus = item.value();
    let modulus_bytes = string_to_bytes(&modulus);
    let mut limbs = modulus_bytes.len();
    let mut block_size = 32; // @audit 32 is set when a least 1 item is found

    if limbs <= 32 {
        limbs = 32;
    } else if limbs <= 48 {
        limbs = 48;
        block_size = 16;
    } else {
        panic!("limbs must be at most 48");
    }

    max_block_size = max_block_size.max(block_size); // @audit will be either 32 or 16
```

Recommendation: The `max_block_size` value could default to either `16` or `32`.

OpenVM: Acknowledged and intentional. The default of 4 is set for defensive coding since that's the minimum that the prover requires. But indeed right now it will always be either 16 or 32.

Cantina Managed: Acknowledged.