Security Audit Report

Autoscale vault
MultiversX smart contract

by ARDA

on April 24, 2025

Table of Contents

Disclaimer

The report makes no statements or warranties, either expressed or implied, regarding the security of the code, the information herein or its usage. It also cannot be considered as a sufficient assessment regarding the utility, safety and bugfree status of the code, or any other statements.

This report does not constitute legal or investment advice. It is for informational purposes only and is provided on an "as-is" basis. You acknowledge that any use of this report and the information contained herein is at your own risk. The authors of this report shall not be liable to you or any third parties for any acts or omissions undertaken by you or any third parties based on the information contained herein.

Terminology

Code: The code with which users interact.

Inherent risk: A risk for users that comes from a behavior inherent to the code's design.

Inherent risks only represent the risks inherent to the code's design, which are a subset of all the possible risks. **No inherent risk doesn't mean no risk.** It only means that no risk inherent to the code's design has been identified. Other kind of risks could still be present. For example, the issues not fixed incur risks for the users, or the upgradability of the code might also incur risks for the users.

Issue: A behavior unexpected by the users or by the project, or a practice that increases the chances of unexpected behaviors to appear.

Critical issue: An issue intolerable for the users or the project, that must be addressed.

Major issue: An issue undesirable for the users or the project, that we strongly recommend to address.

Medium issue: An issue uncomfortable for the users or the project, that we recommend to address.

Minor issue: An issue imperceptible for the users or the project, that we advise to address for the overall project security.

Objective

Our objective is to share everything we have found that would help assessing and improving the safety of the code:

1. The **inherent risks** of the code, labelled R1, R2, etc.
2. The **issues** in the **code**, labelled C1, C2, etc.
3. The **issues** in the **testing** of the code, labelled T1, T2, etc.
4. The **issues** in the **other** parts related to the code, labelled O1, O2, etc.
5. The **recommendations** to address each issue.

Audit Summary

Initial scope

- **Repository:** https://github.com/autoscale-defi/sc-app-rs
- **Commit:** `1555405073efcb58cb6b3da6acade0661cab4a41`
- **MultiversX smart contract path:** ./vault/

Final scope

- **Repository:** https://github.com/autoscale-defi/sc-app-rs
- **Commit:** `6b99f42ad0a365f321795d2d2c0ad9b71638521d`
- **MultiversX smart contract path:** ./vault/

2 inherent risks in the final scope

0 issue in the final scope

24 issues reported in the initial scope and 0 remaining in the final scope:

Severity	Reported			Remaining		
	Code	Test	Other	Code	Test	Other
Critical	1	0	0	0	0	0
Major	4	0	0	0	0	0
Medium	8	0	0	0	0	0
Minor	11	0	0	0	0	0

Inherent Risks

R1: Users are not guaranteed to earn more than if they followed the optimal strategy as a standalone user.

This is because the yields from the strategy would be reduced in the following situations:

- Deposit and withdraw fees: When a user deposits and withdraws in a strategy, a fee can be taken on his assets,
- Delays: If there are few active compounders and rewards are compounded with delays, it would lead to smaller yields,
- Increase of compound fees: The Autoscale team can increase the fees taken on rewards, which would reduce the users' yields,
- Rewards gaming: If the fees for depositing and withdrawing funds are insufficiently deterrent, e.g. they have values 0%, then some malicious users could perform quick enter-and-exit tactics to earn rewards from strategies which they do not deserve, thus reducing other users' yields,
- Unfavorable market conditions: If the strategy compounds rewards, the amount of reinvested rewards depend on the price and slippage in the liquidity pools used for the swaps, which are unpredictable and might be manipulated,
- Reduction of HTM staked in Hatom Booster: At any time the Autoscale team can withdraw some staked HTM from Hatom Booster, which would reduce the Booster's rewards.

R2: Users might not be protected against enter-and-exit tactics which would make them lose some rewards.

Users who quickly deposit assets before a compound and withdraw just after, would steal a portion of the compounded rewards from other, honest users who actively participate in the protocol.

To protect against such enter-and-exit tactics, the Autoscale team can activate a penalty mechanism: if users withdraw before a certain duration, they would pay a penalty on the withdrawn assets.

However, there is no guarantee that the chosen values for the duration and penalty are big enough to dissuade against enter-and-exit tactics. Thus, if such tactics remain profitable, attackers might exploit them and steal a portion of the Vault’s rewards.

Code Issues & Recommendations

Since the code is not open-source, only the remaining issues are published.