Security Audit Report

Hatom controller (2)

MultiversX smart contract

by ARDA

on November 24, 2023

Table of Content

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/HatomProtocol/hatom-protocol
- **Commit:** `f1388191ae7f17765917f71074fd93ca665f8783`
- **MultiversX smart contract path:** ./controller/

Final scope

- **Repository:** https://github.com/HatomProtocol/hatom-protocol
- **Commit:** `648977c004d99665116fd1971c99fb018fa19cef`
- **MultiversX smart contract path:** ./controller/

3 inherent risks in the final scope

0 issue in the final scope

19 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	2	0	0	0	0	0
Medium	10	0	0	0	0	0
Minor	6	0	0	0	0	0

Inherent Risks

R1: The solvency of a user might be incorrectly assessed, possibly leading to bad debt or to the liquidations of solvent users.

This is because the solvency of a user depends on the value of his collateral relative to the value of his debt, and the prices of these tokens are obtained from Hatom Oracle, thus there is a risk as for any oracle that incorrect prices are returned. Consequently:

- Insolvent users might be deemed solvent: This would prevent the liquidations of these users, and would also allow them to borrow assets or withdraw collateral, possibly creating bad debt and preventing lenders from withdrawing their funds.
- Solvent users might be deemed insolvent: This could result in unexpected liquidations, possibly making borrowers lose funds.

R2: Lenders have no guarantee that liquidations of insolvent borrowers will be timely performed.

This is because liquidations must be triggered by external accounts, therefore it is possible that at a time when some users are insolvent, there are no sufficiently active liquidators to perform liquidations. This could in turn create bad debt and prevent lenders in the affected money markets from fully withdrawing their funds.

R3: Users may not be able to claim rewards as HTM if they claim too late.

This is because the contract has only a limited amount of rewards that can be converted to HTM.

Example: Let’s say that if Alice claims now, she would be able to claim rewards as HTM. However, if she rather decides to claim one week later, it is possible that she may not be able to claim rewards as HTM anymore, for instance in the following cases:

- Other users have claimed rewards as HTM during the week, and there are not enough remaining rewards that can be converted to HTM for Alice.
- No other users claimed during the week, but Alice’s rewards have increased and may have now exceeded the contract’s amount of rewards that can be converted to HTM.

Code Issues & Recommendations

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