Security Audit Report

Hatom money-market (2)
MultiversX smart contract

by ARDA

on January 17, 2025

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:** `ca6d658d58beabcb2351f987c7b5ab02690b96fc`
- **MultiversX smart contract path:** ./money-market/

Final scope

- **Repository:** https://github.com/HatomProtocol/hatom-protocol
- **Commit:** `e636e6e6ddba00c090c6cd324063c8d5a48bb952`
- **MultiversX smart contract path:** ./money-market/

1 inherent risk in the final scope

0 issue in the final scope

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

Severity	Reported			Remaining		
	Code	Test	Other	Code	Test	Other
Critical	0	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	0	0	0	0	0	0
Minor	2	0	0	0	0	0

Inherent Risks

R1: Lenders might not be able to withdraw the tokens they supplied.

This is because lenders can’t withdraw their supplied tokens in either of the following situations:

1. Borrowers are non-liquidable and never entirely reimburse their debt.
2. Borrowers are liquidable but liquidations never entirely repay their debt.

In the 2nd situation, there are two possible reasons why liquidations might never entirely repay the debt of liquidable borrowers:

- Liquidations are not performed, e.g. because there are no active liquidators or if prices fail to be obtained from the Oracle,
- Liquidations are performed too late, i.e. once the borrower is insolvent, as then even after fully liquidating his collateral, the borrower’s debt will not be entirely repaid.

Code Issues & Recommendations

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