Security Audit Report

Hatom ush-money-market
MultiversX smart contract

by ARDA

on March 4, 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:** `c7dfa00ea09700243fe9b5163427ddcb52cd0955`
- **MultiversX smart contract path:** ./ush-money-market/

Final scope

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

4 inherent risks in the final scope

0 issue in the final scope

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

Severity	Reported			Remaining		
	Code	Test	Other	Code	Test	Other
Critical	3	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	7	1	0	0	0	0
Minor	4	0	0	0	0	0

Inherent Risks

R1: 1 USH minted by the USH money market might not be backed by collateral with value superior to 1 dollar.

This is because there is a trust that the oracles providing prices and that the liquidation bots work properly:

- If for any reason the prices returned by the oracles are erroneous, then the collateral of USH borrowers might have a smaller dollar value than the amount of borrowed USH.
- If for any reason liquidations are not executed or fail to be executed, then the amount of borrowed USH might continue to increase and exceed the dollar value of the collateral of USH borrowers.

R2: Users might not be able to acquire the USH needed for repaying their debt.

This is because, in the USH money market, users are forced to repay their whole debt in USH, including the interests, and in order to acquire USH they have two options:

1. Minting new USH: At the time of this audit, users can only mint new USH by making new borrows in the USH money market and USH isolated lending modules. However, by making a new borrow, users would increase their total debt, so this approach does not let them repay their current debt.
2. Finding USH in the circulating supply: However, the circulating supply is the total amount of USH that has been borrowed as well as the interests already minted by Hatom admins. Therefore, this supply can be smaller than the total debt, and the part of this supply which can be acquired can be even smaller. In turn, it might be impossible for all users to repay their debts unless other users borrow USH to increase the circulating supply.

R3: The less a user interacts with the USH money market, the bigger the interest he has to repay.

This is because the discount mechanism applies a discount on the user's interest each time he interacts with the USH money market. Thus, at each interaction, the discount is applied on the interest, making the total debt of the user smaller than if he did not interact.

Example: Two users Alice and Bob who borrowed 100 USH enjoy a 50% discount on their interest, and in the money market the monthly interest rate is of 60%.

1) On the one hand, Alice interacts with the USH money market only once, after 2 months. The interest she has to repay is 78 USH (`50% * 100 * (1.6^2 - 1)`), so her new total debt is 178 USH.

2) On the other hand, Bob interacts with the USH money market more frequently. He interacts after 1 month, and then again after another month. After the 1st month, the interest he has to repay is 30 USH (`50% * 100 * (1.6 - 1)`), hence his total debt becomes 130 USH. After the 2nd month, the interest he has to repay is 39 USH (`50% * 130 * (1.6 - 1)`), hence his total debt is 169 USH, which is smaller than than Alice's total debt.

R4: The less a user interacts with the USH money market, the more borrow rewards he loses.

This is because the user's share of the borrow rewards is his total debt computed at his last interaction. Therefore, users interacting less frequently with the USH money market will have a smaller share and lose borrow rewards that go to users interacting more frequently.

Example: There are two borrowers in the USH money market, Alice and Bob, both borrowing 1000 USH at a monthly interest rate of 5%. Besides, there are 1000$ of borrow rewards distributed monthly. After 1 month, Alice interacts with the USH money market. Since her total debt is 1200 USH, her share of the borrow rewards becomes 60%. The 2nd month passes, during which Bob's share of the borrow rewards is 40%. Therefore, when they both claim at the end of the 2nd month, Bob earns 400$ for the 2nd month, while Alice earns 600$, i.e. Bob has lost 100$ of borrow rewards which have gone to Alice.

Code Issues & Recommendations

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

Test Issues & Recommendations

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