Security Audit Report

Hatom account-manager-deployer
MultiversX smart contract

by ARDA

on March 4, 2025

Table of Content

Disclaimer 3

Terminology 3

Objective 4

Audit Summary 5

Inherent Risks 6

Test Issues & Recommendations 7

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-account-manager-deployer
- **Commit:** `3dc9b4de022a7da3c62d5a3d48eff594ce718454`
- **MultiversX smart contract path:** ./account-manager-deployer/

Final scope

- **Repository:** https://github.com/HatomProtocol/hatom-account-manager-deployer
- **Commit:** `410b38e374c0430526f9d340f05e55051d29d5c2`
- **MultiversX smart contract path:** ./account-manager-deployer/

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	2	0	0	0	0
Minor	0	0	0	0	0	0

Inherent Risks

R1: A user can’t change the Account Manager tied to him.

This is because the Deployer authorizes each user to deploy at most one Account Manager, which is forever tied to the user.

Therefore, in particular, if the user transfers the ownership of his Account Manager smart contract to another address, then the user won’t ever be able to deploy a new Account Manager that is tied to him and that he could own.

Test Issues & Recommendations

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