Security Audit Report

Hatom liquid-staking
smart contract

by ARDA

on July 15, 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

Inherent risk: A risk for users that comes from a behavior inherent to the smart contract design.

Inherent risks only represent the risks inherent to the smart contract 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 smart contract 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 smart contracts deployed as upgradeable 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.

Audit Summary

Scope of initial audit

- **Repository:** https://github.com/HatomProtocol/hatom-liquid-staking
- **Commit:** `843a4d995345e0aed74e67c6b28451e32bb299ca`
- **Path to Smart contract:** `./liquid-staking/`

Scope of final audit

- **Repository:** https://github.com/HatomProtocol/hatom-liquid-staking
- **Commit:** `77cf558862226a4aab0caa8111384d3bc3112afb`
- **Path to Smart contract:** `./liquid-staking/`

Report objectives

1. Reporting all **inherent risks** of the smart contract.
2. Reporting all **issues** in the smart contract **code**.
3. Reporting all **issues** in the smart contract **test**.
4. Reporting all **issues** in the **other** parts of the smart contract.
5. Proposing **recommendations** to address all issues reported.

1 inherent risk in the final commit

0 issue in the final commit

32 issues reported from the initial commit and 0 remaining in the final commit:

Severity	Reported			Remaining		
	Code	Test	Other	Code	Test	Other
Critical	4	0	0	0	0	0
Major	12	0	0	0	0	0
Medium	10	0	0	0	0	0
Minor	6	0	0	0	0	0

Inherent Risks

R1: Users might make smaller profits than if they delegated directly to the provider of their choice.

This is because:

1) A protocol fee is applied on the rewards generated from staked EGLD.

2) Users’ EGLD might not all be delegated in the providers generating the highest profits:

- Hatom admins decide which providers are allowed for delegations.
- The selection of the provider where users’ EGLD are delegated depends on off-chain data about providers provided by an oracle, therefore these data might be inaccurate and/or not favor the providers that would lead to highest profits.
- The algorithm selecting the provider where users’ EGLD are delegated / undelegated might not select the providers which generate the highest / lowest profits.
- The actual delegation of EGLD can only be made by admins. Therefore if admins are inactive, these EGLD would not generate any profits to users.
- The admins are allowed to undelegate EGLD from a provider of their choice, in order to re-delegate these EGLD afterwards in another provider, chosen by the selection algorithm. Until they are re-delegated, which takes at least 10 days to pass the unbonding period, these EGLD would not generate any profits to users.

Code Issues & Recommendations

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