



SMART CONTRACTS REVIEW



April 30th 2024 | v. 1.0

Security Audit Score

PASS

Zokyo Security has concluded that
this smart contract passed a security
audit.



SCORE
100

ZOKYO AUDIT SCORING HEIR/ES

1. Severity of Issues:

- Critical: Direct, immediate risks to funds or the integrity of the contract. Typically, these would have a very high weight.
- High: Important issues that can compromise the contract in certain scenarios.
- Medium: Issues that might not pose immediate threats but represent significant deviations from best practices.
- Low: Smaller issues that might not pose security risks but are still noteworthy.
- Informational: Generally, observations or suggestions that don't point to vulnerabilities but can be improvements or best practices.

2. Test Coverage: The percentage of the codebase that's covered by tests. High test coverage often suggests thorough testing practices and can increase the score.

3. Code Quality: This is more subjective, but contracts that follow best practices, are well-commented, and show good organization might receive higher scores.

4. Documentation: Comprehensive and clear documentation might improve the score, as it shows thoroughness.

5. Consistency: Consistency in coding patterns, naming, etc., can also factor into the score.

6. Response to Identified Issues: Some audits might consider how quickly and effectively the team responds to identified issues.

HYPOTHETICAL SCORING CALCULATION:

Let's assume each issue has a weight:

- Critical: -30 points
- High: -20 points
- Medium: -10 points
- Low: -5 points
- Informational: -1 point

Starting with a perfect score of 100:

- 0 Critical issues: 0 points deducted
- 0 High issues: 0 points deducted
- 0 Medium issues: 0 points deducted
- 1 Low issue: 1 resolved = 0 points deducted
- 3 Informational issues: 3 resolved = 0 points deducted

Hence, the score is 100.

TECHNICAL SUMMARY

This document outlines the overall security of the Heir/es smart contract/s evaluated by the Zokyo Security team.

The scope of this audit was to analyze and document the Heir/es smart contract/s codebase for quality, security, and correctness.

Contract Status



There were 0 critical issues found during the review. (See Complete Analysis)

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract/s but rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that can withstand the Ethereum network's fast-paced and rapidly changing environment, we recommend that the Heir/es team put in place a bug bounty program to encourage further active analysis of the smart contract/s.

Table of Contents

Auditing Strategy and Techniques Applied	5
Executive Summary	7
Structure and Organization of the Document	8
Complete Analysis	9

AUDITING STRATEGY AND TECHNIQUES APPLIED

The source code of the smart contract was taken from the Heir/es repository:

Repo: <https://github.com/SentienceQuest/heirloom>

Initial commit - bb4e57db5007f77cf7d75637adaa7a4244e3e78a

Fixes - <https://github.com/SentienceQuest/heirloom/pull/5>

Within the scope of this audit, the team of auditors reviewed the following contract(s):

- heiress.sol

During the audit, Zokyo Security ensured that the contract:

- Implements and adheres to the existing standards appropriately and effectively;
- The documentation and code comments match the logic and behavior;
- Distributes tokens in a manner that matches calculations;
- Follows best practices, efficiently using resources without unnecessary waste;
- Uses methods safe from reentrance attacks;
- Is not affected by the most recent vulnerabilities;
- Meets best practices in code readability, etc.

Zokyo Security has followed best practices and industry-standard techniques to verify the implementation of Heir/es smart contract/s. To do so, the code was reviewed line by line by our smart contract developers, who documented even minor issues as they were discovered. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

01

Due diligence in assessing the overall code quality of the codebase.

03

Thorough manual review of the codebase line by line.

02

Cross-comparison with other, similar smart contract/s by industry leaders.

Executive Summary



The Zokyo team has performed a security audit of the provided codebase. The contract submitted for auditing is well-crafted and organized. Detailed findings from the audit process are outlined in the "Complete Analysis" section.

STRUCTURE AND ORGANIZATION OF THE DOCUMENT

For the ease of navigation, the following sections are arranged from the most to the least critical ones. Issues are tagged as “Resolved” or “Unresolved” or “Acknowledged” depending on whether they have been fixed or addressed. Acknowledged means that the issue was sent to the Heir/es team and the Heir/es team is aware of it, but they have chosen to not solve it. The issues that are tagged as “Verified” contain unclear or suspicious functionality that either needs explanation from the Client or remains disregarded by the Client. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:



Critical

The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.



High

The issue affects the ability of the contract to compile or operate in a significant way.



Medium

The issue affects the ability of the contract to operate in a way that doesn't significantly hinder its behavior.



Low

The issue has minimal impact on the contract's ability to operate.



Informational

The issue has no impact on the contract's ability to operate.

COMPLETE ANALYSIS

FINDINGS SUMMARY

#	Title	Risk	Status
1	The ModuleCanceled event emits the wrong value for the timer parameter	Low	Resolved
2	densityModule0, densityModule1, densityModule2, densityModule3, densityModule4 could be set immutable	Informational	Resolved
3	Unnecessary if condition for densityModule_timer[_densityModule][_module]	Informational	Resolved
4	The wrong spell in the ModuleBeneficiaryReplaced event name	Informational	Resolved

LOW-1 | RESOLVED

The ModuleCanceled event emits the wrong value for the timer parameter.

ModuleCanceled event emits the timer value of the module to be canceled for the third parameter.

densityModule_timer[_densityModule][_moduleId] is emitted in the event but it is set to 0 before emitting.

So the ModuleCanceled event will always emit 0 for the timer parameter.

Recommendation:

Add a local variable for densityModule_timer[_densityModule][_moduleId] and emit the local variable instead of densityModule_timer[_densityModule][_moduleId].

INFORMATIONAL-1 | RESOLVED

densityModule0, densityModule1, densityModule2, densityModule3, densityModule4 could be set immutable.

densityModule0, densityModule1, densityModule2, densityModule3, densityModule4 variables are only set in the constructor.

Recommendation:

Set the variables immutable.

Unnecessary if condition for densityModule_timer[_densityModule][_module].

<https://github.com/SentienceQuest/heirloom/blob/main/src/Heirloom.sol#L55>

In the moduleSupportedRequirements() modifier, if ownerOfWill is not zero, there is no need to check if densityModule_timer[_densityModule][_module] is zero because densityModule_owner and densityModule_timer variables are set to non-zero or zero at the same time.

Recommendation:

Remove if condition for densityModule_timer.

The wrong spell in the ModuleBeneficiaryReplaced event name.

<https://github.com/SentienceQuest/heirloom/blob/main/src/Heirloom.sol#L25>

ModuleBeneficiaryReplaced should be replaced with ModuleBeneficiaryReplaced.

Recommendation:

Change the event name to ModuleBeneficiaryReplaced.

heiress.sol	
Reentrance	Pass
Access Management Hierarchy	Pass
Arithmetic Over/Under Flows	Pass
Unexpected Ether	Pass
Delegatecall	Pass
Default Public Visibility	Pass
Hidden Malicious Code	Pass
Entropy Illusion (Lack of Randomness)	Pass
External Contract Referencing	Pass
Short Address/ Parameter Attack	Pass
Unchecked CALL	Pass
Return Values	
Race Conditions / Front Running	Pass
General Denial Of Service (DOS)	Pass
Uninitialized Storage Pointers	Pass
Floating Points and Precision	Pass
Tx.Origin Authentication	Pass
Signatures Replay	Pass
Pool Asset Security (backdoors in the underlying ERC-20)	Pass

We are grateful for the opportunity to work with the Heir/es team.

The statements made in this document should not be interpreted as an investment or legal advice, nor should its authors be held accountable for the decisions made based on them.

Zokyo Security recommends the Heir/es team put in place a bug bounty program to encourage further analysis of the smart contract by third parties.

