WHITE HAT DAO

Hundred Finance

Smart Contract Audit Report

hundred.finance

By White Hat DAO

www.whitehatdao.com

Date: 26/02/2022

WHITE HAT DAO

Table of Contents

WHITE HAT DAO

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report.

In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and White Hat DAO and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives)

White Hat DAO owes no duty of care towards you or any other person, nor does White Hat DAO make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties, or other terms of any kind except as set out in this disclaimer, and White Hat DAO hereby excludes all representations, warranties, conditions, and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report.

Except and only to the extent that it is prohibited by law, White Hat DAO hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against White Hat DAO, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of the use of this report, and any

reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security.

No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service, or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

WHITE HAT DAO

Executive Summary

White Hat DAO was contracted by "HundredDAO" team to conduct a smart contract security audit. This report presents the findings of the security assessment conducted between Feb. 8, 2022, and Feb. 20, 2022. There were 4 smart contracts reviewed during this audit. The smart contracts were manually reviewed and analyzed with static analysis tools.

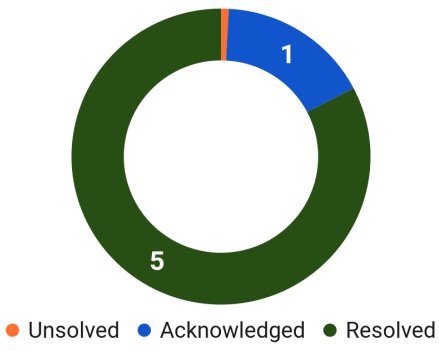
Based on our audit, the customers' smart contracts' safety rating is shown below:

We found 1 major issue and some minor and informational issues. The major issue falls under the "Centralization/Privilege" category. For a full list of these issues please refer to the Findings section of the report.

The code had good comments and documentation. The code uses the natspec standard for comments. Commenting can make the maintenance of the code much easier, as well as help, make finding bugs faster. Also, commenting is very important when writing functions that may be used in other contracts.

Here is a high-level overview of the issues found in this report:

Total Issues	6 (5 Resolved)
Critical Issues	0 (0 Resolved)
Major Issues	1 (0 Resolved)
Medium Issues	1 (1 Resolved)
Minor Issues	3 (3 Resolved)
Informational	1 (1 Resolved)



WHITE HAT DAO

Summary of Findings

Issue ID	Issue Title	Category	Severity	Status
HND-1	Centralization Risk	Centralization/ Privilege	Major	Acknowledged
HND-2	Unchecked Admin Change	Volatile Code	Medium	Resolved
HND-3	Unused Variable	Gas Optimization	Minor	Resolved
HND-4	Unnecessary Check	Gas Optimization	Minor	Resolved
HND-5	Unnecessary Storage Read	Gas Optimization	Minor	Resolved
HND-6	Lack of Event Emission for Significant Transactions	Coding Style	Informational	Resolved

WHITE HAT DAO

Introduction

This security assessment has been prepared for "HundredDAO" to find any safety concerns, bad practices, and vulnerabilities in the source code as well as any contract dependencies in scope that were not part of an officially recognized library. Comprehensive tests have been conducted, utilizing manual code review, static analysis, and techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors
- Assessing the codebase to ensure compliance with current best practices and industry standards
- Ensuring contract logic meets the specifications and intentions of the client. Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders
- Thorough line-by-line manual review of the entire codebase by industry experts
- Reviewing unit tests to ensure full coverage of the codebase

The Project Summary, Scope, Audit Details and Methodology of the audit is described in the following sections.

The security assessment resulted in findings that ranged from Major to informational. We recommend addressing these findings to ensure a high level of security standards. These can be found in the Findings section of the report.

WHITE HAT DAO

Project Summary

Project	Hundred DAO
Description	Hundred Finance is a decentralized application (dApp) that enables the lending and borrowing of crypto-currencies. A multi-chain protocol, it integrates with Chainlink oracles to ensure market health and stability, while specializing in providing markets for long-tail assets.
Platform	ETH (Multichain)
Language used	Vyper
Codebase	https://github.com/hundred-finance/hundred-dao
Commit	0d0e431da4849d5e353b8252f7aac113dc25f9fa

Project Scope

White Hat DAO was commissioned by The HundredDAO to perform security assessments on the following smart contracts:

Source Code	Acknowledgement	SHA-256
GaugeControllerV2.vy	Accepted	033953D227B26155CAA6EA1717E3B59FB4 5560600E8F379B0CF65F1CC994833B
LiquidityGaugeV4_1.vy	Accepted	3DF58EB754E012D7FD0F7800DBE87F967 9F2324E1628BAA8F36CDB69F6B76D3F
MirroredVotingEscrow.vy	Accepted	FE03F8DED807B3A5C0682A368D2865836 C89CEC05F606193071039B7475BBC64
VotingEscrowV2.vy	Accepted	7A4DF4179BA145952F2E7CA3C9FCBFA48 037597081F7C1B1D989E437CBBCDA65

WHITE HAT DAO

Audit Details

Delivery Date	26/02/2022
Submission Date	08/02/2022
Key Components	GaugeControllerV2.vy, MirroredVotingEscrow.vy, LiquidityGaugeV4_1.vy, GaugeControllerV2.vy

Methodology

White Hat DAO auditing team reviewed the code base of "HundredDAO" from Feb. 8, 2022, and Feb. 20, 2022. The team conducted the assessment based on the codebase in the repository at commit [0d0e431da4849d5e353b8252f7aac113dc25f9fa](0d0e431da4849d5e353b8252f7aac113dc25f9fa).

The team launched the audit by analyzing the specifications of the project and the key areas of interest and went through the documentation.

The code was manually reviewed in an attempt to identify potential vulnerabilities, code has good unit tests coverage. Automated analysis of the codebase was performed and results were reviewed.

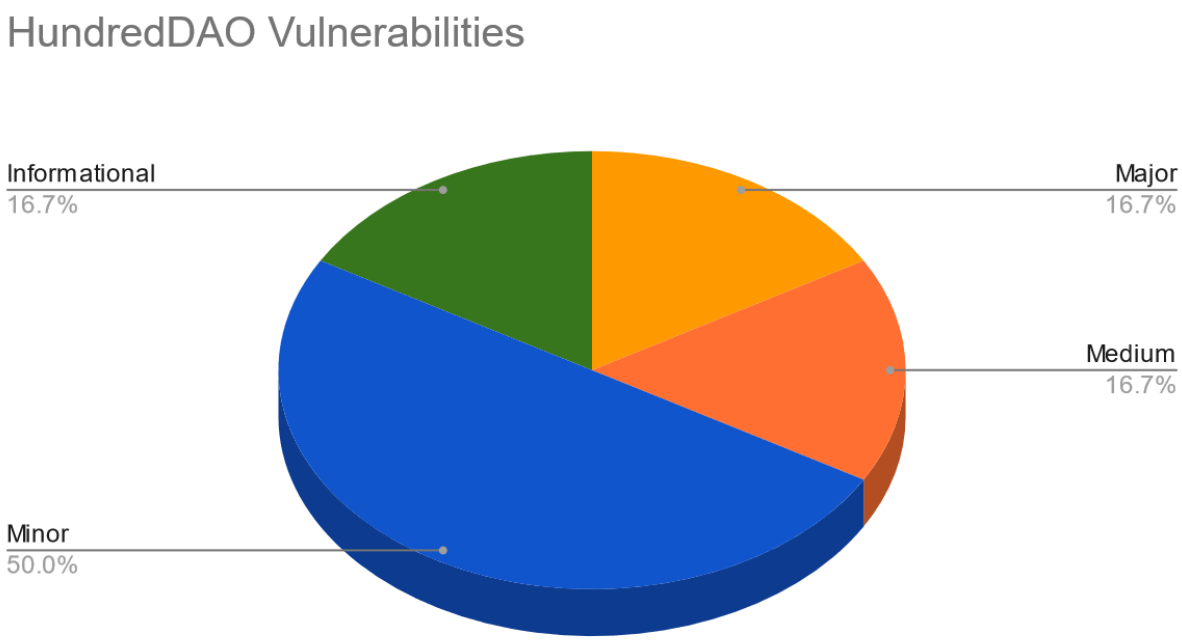
The smart contracts were scanned for commonly known and more specific vulnerabilities. Following is the list of some of the vulnerabilities that were considered during the audit of the smart contract:

- Access Control
- Arbitrary token minting
- Business Logics Review
- Centralization of power
- Code clones, functionality duplication
- Conditional Completion attack

- Costly Loop
- Ownership Takeover
- Redundant fallback function
- Reentrancy
- Remote code execution
- User Balances manipulation
- Logic Flaws
- Scoping and Declarations
- Integer Overflow and Underflow attacks

Findings

We found 1 major issue and some minor and informational issues. The major issue falls under the “Centralization/Privilege” category. There were also some minor/information vulnerabilities around “Gas Optimization” and “Coding Styles”. Additional information on these vulnerabilities is provided in the following sections.



Critical - 0 | Major - 1 | Medium Issue - 1 | Minor - 3 | Informational -1

WHITE HAT DAO

Severity Definitions

Severity	Definitions
Critical	These vulnerabilities have a catastrophic impact on the security of the project. They can lead to loss, data manipulation, take over, etc. It is strongly recommended to fix these vulnerabilities.
Major	These vulnerabilities have a significant impact on the security of the project. They can lead to loss, data manipulation, take over, etc. It is strongly recommended to fix these vulnerabilities.
Medium	These vulnerabilities are important to fix. These vulnerabilities alone can't lead to asset loss or data manipulation. However, medium vulnerabilities can be chained to create a more severe vulnerability. It is highly recommended to review and address these vulnerabilities.
Minor	These vulnerabilities are mostly related to outdated, unused code snippets and don't have a significant impact on execution. It is suggested that the project party evaluate and consider whether these vulnerabilities need to be fixed.
Informational	These vulnerabilities don't pose an immediate risk but are relevant to security best practices. They could be code-style violations and informational statements that don't affect smart contract execution. They may be able to be ignored.

Critical Vulnerabilities

No Critical severity vulnerabilities were found.

Major Vulnerabilities

HND-01 | Centralization Risk

Type: Centralization/Privilege
Level: Major

Description: `whitelisted_mirrors` have total control of setting mirrored locks, `whitelisted_mirrors` are also set by the owner, `whitelisted_mirrors` can set any amount and lock without any checks.

Recommendation: We recommend adding more restrictions to the owner and whitelisted mirrors.

Details:

File: `MirroredVotingEscrow.vy (344)`

```
def mirror_lock(_user: address, _chain: uint256, _escrow_id: uint256, _value: uint256,
_unlock_time: uint256):
```

Update: The HundredDAO team has acknowledged the findings, and explained that The `whitelisted_mirrors` is meant to hold the bridge contract addresses that will be used to trigger a user lock mirroring. This might change in the future when they have a defined bridging solution. Meanwhile, it will be under the control of the team multisig and only the team multisig will be able to execute lock mirroring actions. We also have plans for organizing mirroring events through the use of special Merkle contracts, in which case that contract will be whitelisted for `mirror_creation`.

Medium Vulnerabilities

HND-02 | Unchecked Admin Change

Type: Volatile Code
Level: Medium

Description: `_new_admin` is set as admin without any checks this can lead to accidental admin being set to zero address.

Recommendation: We recommend making admin changes to a Commit/Apply routine where the new admin has to accept ownership for the change to be made.

Details:

File: `MirroredVotingEscrow.vy (452)`

```
@external
def set_admin(_new_admin: address):
    assert msg.sender == self.admin # dev: only admin

    self.admin = _new_admin
```

Update: The HundredDAO team has acknowledged the findings and implemented WhiteHatDAO's recommendation in commit 27334753a0ad7a3cb75483683af124e6a36c4f86

Minor Vulnerabilities

HND-03 | Unused Variable

Type: Gas Optimization
Level: Minor

Description: Variable _n_gauges is declared but never used.

Recommendation: Declare as immutable [ref].

Details:

File: `GaugeControllerV2.vy (489)`

```
_n_gauges: int128 = self.n_gauges
```

Update: The HundredDAO team has acknowledged the findings and implemented WhiteHatDAO's recommendation in commit 27334753a0ad7a3cb75483683af124e6a36c4f86

HND-04 | Unnecessary Check

Type: Gas Optimization
Level: Minor

Description: `power_used` is declared as uint256. Vyper will throw if `power_used` is negative and so the check on power >= 0 is unnecessary.

Recommendation: Remove the unnecessary check.

Details:

File: GaugeControllerV2.vy: (97)

```
assert (power_used >= 0) and (power_used <= 10000), 'Used too much power'
```

Update: The HundredDAO team has acknowledged the findings and implemented WhiteHatDAO's recommendation in commit 27334753a0ad7a3cb75483683af124e6a36c4f86

HND-05 | Unnecessary Storage Read

Type: Gas Optimization
Level: Minor

Description: `_balance` is memory copy of storage state `self.balanceOf[addr]` and it should be used instead of re-accessing `self.balanceOf[addr]` (12 gas for `_balance` vs 400 gas for `self.balanceOf[addr]`).

Recommendation: Use the memory variable `_balance`.

Details:

File: LiquidityGaugeV4_1.vy (425)

```
    _balance: uint256 = self.balanceOf[addr]

    assert ERC20(_voting_escrow).balanceOf(addr) == 0 or t_ve > t_last # dev: kick not
allowed
    assert self.working_balances[addr] > _balance * TOKENLESS_PRODUCTION / 100  # dev:
kick not needed

    self._checkpoint(addr)
    self._update_liquidity_limit(addr, self.balanceOf[addr], self.totalSupply)
```

Update: The HundredDAO team has acknowledged the findings and implemented WhiteHatDAO's recommendation in commit 27334753a0ad7a3cb75483683af124e6a36c4f86

Informational Vulnerabilities

HND-06 | Lack of Event Emission for Significant Transactions

Type: Coding Style
Level: Informational

Description: Functions that affect important and sensitive state variables should emit events as notifications to users.

Recommendation: Emit events on important state changes.

Details:

File: `MirroredVotingEscrow.vy (452,459,466)`

```
def set_admin(_new_admin: address):

def set_mirror_whitelist(_addr: address, _is_whitelisted: bool):

def add_voting_escrow(_addr: address):
```

Update: The HundredDAO team has acknowledged the findings and implemented WhiteHatDAO's recommendation in commit 27334753a0ad7a3cb75483683af124e6a36c4f86

Conclusion

White Hat DAO has worked with the "HundredDAO" team to perform this audit. There were 4 smart contracts reviewed during this audit. The smart contracts were manually reviewed and analyzed with static analysis tools. The findings of these reviews were provided in this report.

The code had good unit tests coverage for all functionalities. We constructed some unit tests to test edge cases. The code was commented well. Comments are helpful in understanding the overall architecture and the logic flow of the contracts.

This audit has found 1 major, 1 medium, 3 minor, and 1 informational vulnerability. Please refer to the Findings section for details.

Update: 02/22/22 - All issues have been resolved except for issue HND-01 where the Hundred DAO Team acknowledged the centralization issue and gave a justification. Please refer to the findings above for more details.

WHITE HAT DAO

Change Log

- 20-02-2022 - Initial report
- 21-02-2022 - Draft v0.1
- 26-02-2022 - Final Report updated with Hundred DAO's response and updates