YEARN CURVE-VOTER-PROXY SMART CONTRACT AUDIT

April 15, 2021

MixBytes()

CONTENTS

1. INTRODUCTION

1.1 DISCLAIMER

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only. The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of Yearn. If you are not the intended recipient(s) of this document, please note that any disclosure, copying or dissemination of its content is strictly forbidden.

1.2 PROJECT OVERVIEW

This is a mock strategy. It shows the flow of how to implement customized functions after inheriting the CurveVoterProxy template.

1.3 SECURITY ASSESSMENT METHODOLOGY

At least 2 auditors are involved in the work on the audit who check the provided source code independently of each other in accordance with the methodology described below:

01 "Blind" audit includes:
> Manual code study
> "Reverse" research and study of the architecture of the code based on the source code only

Stage goal:
Building an independent view of the project's architecture
Finding logical flaws

02 Checking the code against the checklist of known vulnerabilities includes:
> Manual code check for vulnerabilities from the company's internal checklist
> The company's checklist is constantly updated based on the analysis of hacks, research and audit of the clients' code

Stage goal:
Eliminate typical vulnerabilities (e.g. reentrancy, gas limit, flashloan attacks, etc.)

03 Checking the logic, architecture of the security model for compliance with the desired model, which includes:
> Detailed study of the project documentation
> Examining contracts tests
> Examining comments in code
> Comparison of the desired model obtained during the study with the reversed view obtained during the blind audit

Stage goal:
Detection of inconsistencies with the desired model

04 Consolidation of the reports from all auditors into one common interim report document
> Cross check: each auditor reviews the reports of the others
> Discussion of the found issues by the auditors
> Formation of a general (merged) report

Stage goal:
Re-check all the problems for relevance and correctness of the threat level
Provide the client with an interim report

05 Bug fixing & re-check.
> Client fixes or comments on every issue
> Upon completion of the bug fixing, the auditors double-check each fix and set the statuses with a link to the fix

Stage goal:
Preparation of the final code version with all the fixes

06 Preparation of the final audit report and delivery to the customer.

Findings discovered during the audit are classified as follows:

FINDINGS SEVERITY BREAKDOWN

Level	Description	Required action
Critical	Bugs leading to assets theft, fund access locking, or any other loss funds to be transferred to any party	Immediate action to fix issue
Major	Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement.	Implement fix as soon as possible
Warning	Bugs that can break the intended contract logic or expose it to DoS attacks	Take into consideration and implement fix in certain period
Comment	Other issues and recommendations reported to/acknowledged by the team	Take into consideration

Based on the feedback received from the Customer's team regarding the list of findings discovered by the Contractor, they are assigned the following statuses:

Status	Description
Fixed	Recommended fixes have been made to the project code and no longer affect its security.
Acknowledged	The project team is aware of this finding. Recommendations for this finding are planned to be resolved in the future. This finding does not affect the overall safety of the project.
No issue	Finding does not affect the overall safety of the project and does not violate the logic of its work.

1.4 EXECUTIVE SUMMARY

The contracts examined in this audit are for the work of the Strategy that is being used for the Vault. The Strategy is working with several tokens at once, which depend on each other. In some cases, some tokens are exchanged for others using the Sushiswap/Uniswap and Curve pools.

1.5 PROJECT DASHBOARD

Client	Yearn
Audit name	Curve-voter-proxy
Initial version	78c92e4ffc0f76651914565744a3607d0248d254 38dc4389981bee6aa7e40e22805157b21f38ac26
Final version	38dc4389981bee6aa7e40e22805157b21f38ac26
SLOC	240
Date	2021-04-05 - 2021-04-15
Auditors engaged	2 auditors

FILES LISTING

Strategy.sol	Strategy.sol

FINDINGS SUMMARY

Level	Amount
Critical	0
Major	0
Warning	1
Comment	3

CONCLUSION

Smart contract has been audited and several suspicious places were found. During audit no critical and one major issues were identified. Several issues were marked as warning and comments. After working on audit report all issues were fixed or acknowledged(if issue is not critical or major) by client, so contracts assumed as secure to use according our security criteria.Final commit identifier with all fixes: `38dc4389981bee6aa7e40e22805157b21f38ac26`

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

Not Found

2.3 WARNING

WRN-1	Variables used but not declared
File	Strategy.sol
Severity	Warning
Status	Fixed at https://github.com/orbxball/curve-voter-proxy/tree/38dc4389981bee6aa7e40e22805157b21f38ac26

DESCRIPTION

At this lines:
Strategy.sol#L226-L227
Strategy.sol#L230
Strategy.sol#L287
the Strategy using address type variables such as `curve`, `gauge`, `reward` and `target` but there is no initialization procedure of these variables.

RECOMMENDATION

It is recommended to initialize your local variables at declaration.

2.4 COMMENTS

CMT-1	Events are probably missing
File	Strategy.sol
Severity	Comment
Status	Acknowledged

DESCRIPTION

At the line Strategy.sol#L157 in method `setKeepCRV()` should probably emit an event `newKeepCRV`.
At the line Strategy.sol#L165 in method `switchDex()` should probably emit an event `newDex`.

RECOMMENDATION

It is recommended to create new events.

CMT-2	Unobvious exchange on Uniswap
File	Strategy.sol
Severity	Comment
Status	Acknowledged

DESCRIPTION

At the lines:

- Strategy.sol#L263
- Strategy.sol#L278
  are not clear why if `_crv > 0`, should be called `swapExactTokensForTokens` of `dex` (which can be address of UniSwap or SushiSwap), but when `_reward > 0`, the same call is made for `uniswap`.

RECOMMENDATION

If this is done on purpose, an explanation is needed. It is recommended to make an explanation.

CMT-3	Code smell
File	Strategy.sol
Severity	Comment
Status	Fixed at https://github.com/orbxball/curve-voter-proxy/tree/38dc4389981bee6aa7e40e22805157b21f38ac26

DESCRIPTION

At the lines Strategy.sol#L239-L300

code has several problems at once.

- Too long function. The function takes 61 lines. It is difficult to understand.
- In one place collected various logic. It is recommended to split into separate methods. For example, lines from 249 to 264 is the function `checkCrv()`, and the lines from 268 to 279 are the `checkReward()` function.
- Duplication code. Lines 255-256, 270-271, 283-284, 289-290 can be combined to call one function with parameters.

RECOMMENDATION

It is recommended to make a refactoring source code.

3. ABOUT MIXBYTES

MixBytes is a team of blockchain developers, auditors and analysts keen on decentralized systems. We build open-source solutions, smart contracts and blockchain protocols, perform security audits, work on benchmarking and software testing solutions, do research and tech consultancy.

BLOCKCHAINS

 Ethereum

 Cosmos

 EOS

 Substrate

TECH STACK

 Python

 Solidity

 Rust

 C++

CONTACTS

 https://github.com/mixbytes/audits_public

 https://mixbytes.io/

 hello@mixbytes.io

 https://t.me/MixBytes

 https://twitter.com/mixbytes