YEARN WETH-MAKER SMART CONTRACT AUDIT

May 07, 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

Part of Yearn Strategy Mix.

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 main purpose of the project is to allow users to add additional ability to use the Maker Protocol managed by the strategy.

1.5 PROJECT DASHBOARD

Client	Yearn
Audit name	Weth-maker
Initial version	39cfbee59bcebfce19a5b9ac6f11fb84f3ab7b23
Final version	3ed8174c550e1b8f23c0ea62151b05dfd70a566b
SLOC	329
Date	2021-04-21 - 2021-05-07
Auditors engaged	2 auditors

FILES LISTING

Strategy.sol	Strategy.sol

FINDINGS SUMMARY

Level	Amount
Critical	0
Major	1
Warning	3
Comment	2

CONCLUSION

Smart contract has been audited and several suspicious places were found. During audit no critical issues were found. One issue was marked major as it may lead to unintended behavior. Several issues were marked as warnings and comments. After working on audit report some issues were fixed by client, but the major issue about `Losses are not taken into account in the strategy` was partially fixed only.
Final commit identifier with all fixes: `3ed8174c550e1b8f23c0ea62151b05dfd70a566b`

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

MJR-1	Losses are not taken into account in the strategy
File	Strategy.sol
Severity	Major
Status	Partially fixed at 3ed8174c

DESCRIPTION

At line: Strategy.sol#L260 when strategy calls the `liquidatePosition()` function if at this point yvdai have losses we have three branches:

- if `_amountNeeded` > real balance of strategy after all liquidations we got revert (it's not very good, because users can't withdraw their funds);
- if `_amountNeeded` < real balance of strategy user will withdraw funds, but vault won't consider losses and all losses will be distributed across all users who left after first withdraw;
- if `yvdai` have losses more than 0.01(Vault.vy#L928) `liquidatePosition()` will reverted while Strategy.sol#L377 (so that means that if yvdai got losses of more than 0.01% any strategies depending on it will be blocked.

Fixes commentary:

- Fixed:
  - Withdrawals from underlying DAI vault with losses can be approved.
  - Withdrawals on failed to liquidate will not be reverted.
- Problem remains:
  - If the strategy suffer a loss from DAI vault, a bad debt will be formed. This debt is hidden until the vault is trying to liquidate it. So we have a bit unfair distribution of losses between vault users.

RECOMMENDATION

It is recommended to rewrite logic of `liquidatePosition()` considering the losses.

2.3 WARNING

WRN-1	The approval value obtained in the constructor may not be enough for the long term of the smart contract
File	Strategy.sol
Severity	Warning
Status	Fixed at 3ed8174c

DESCRIPTION

At line: Strategy.sol#L68 the smart contract constructor calls `_approveAll()` function for different tokens. But in the process of work, the obtained value will only decrease. If this value decreases to zero, then the tokens will remain locked in the contract forever.

RECOMMENDATION

It is recommended to add a function to increase the value of approvals.

WRN-2	There is no check on the result of the function
File	Strategy.sol
Severity	Warning
Status	Acknowledged

DESCRIPTION

According to the ERC20 standard, the `approve()` function returns a boolean value. But in the contract at lines Strategy.sol#L68-L75, after the call to the `_approveAll()` function, this values are not processed. A situation may arise that a False will return.

RECOMMENDATION

It is recommended to add a check of the return value.

WRN-3	Wrong comparison
File	Strategy.sol
Severity	Warning
Status	Acknowledged

DESCRIPTION

At line: Strategy.sol#L217 left operand has unit `1e+6` and the right operand has unit `1e+10`.

```
if (_current > DENOMINATOR.mul(c_safe).mul(1e2)) {//  1e6 > 10000 * 40000 * 1e2
    _current = DENOMINATOR.mul(c_safe).mul(1e2);

```

In this implementation where `c_safe = 40000;` this code will be ignored.

RECOMMENDATION

It is recommended to change `c_safe` value after deploy.

2.4 COMMENTS

CMT-1	Using "magic" numbers
File	Strategy.sol
Severity	Comment
Status	Fixed at 3ed8174c

DESCRIPTION

At line Strategy.sol#L60-L61 use some of unknown variables impairs its understanding.

RECOMMENDATION

It is recommended that you create variables with meaningful names for using numeric values or add comments.

CMT-2	Use constants
File	Strategy.sol
Severity	Comment
Status	Fixed at 3ed8174c

DESCRIPTION

At lines Strategy.sol#L32-L41 you may add constants for save gas.

RECOMMENDATION

It is recommended to add constants to hardcoded address variables.

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