

# YEARN YVECRV VAULT SMART CONTRACT AUDIT

April 06, 2021

MixBytes()

# CONTENTS

1. INTRODUCTION.....	1
DISCLAIMER.....	1
PROJECT OVERVIEW.....	1
SECURITY ASSESSMENT METHODOLOGY.....	2
EXECUTIVE SUMMARY.....	4
PROJECT DASHBOARD.....	4
2. FINDINGS REPORT.....	6
2.1. CRITICAL.....	6
2.2. MAJOR.....	6
MJR-1 Correct migration.....	6
2.3. WARNING.....	7
WRN-1 Missed token.....	7
WRN-2 The approval value obtained in the constructor may not be enough for the long term of the smart contract.....	8
2.4. COMMENTS.....	9
CMT-1 There is no input parameter processing in the method.....	9
CMT-2 Event is probably missing.....	10
3. ABOUT MIXBYTES.....	11

# 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 Brownie 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 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 and Curve pools.

## 1.5 PROJECT DASHBOARD

Client	Yearn
Audit name	yveCRV-vault
Initial version	6706b9ad45e71ee9014454419f229adfa6409f1d bbecd326f2155e5160e5570d7b7270574ddf1dc8
Final version	bbecd326f2155e5160e5570d7b7270574ddf1dc8
SLOC	225
Date	2021-03-21 - 2021-04-06
Auditors engaged	2 auditors

## FILES LISTING

Strategy.sol	Strategy.sol
--------------	--------------

## FINDINGS SUMMARY

Level	Amount
Critical	0
Major	1
Warning	2
Comment	2

## CONCLUSION

Smart contracts have been audited and several suspicious places have been spotted. During the audit no critical issues were found, one issue was marked as major because it could lead to some undesired behavior, also several warnings and comments were found and fixed by the client. After working on the reported findings all of them were resolved or acknowledged. Final commit identifier with all fixes:

`bbecd326f2155e5160e5570d7b7270574ddf1dc8`

# 2. FINDINGS REPORT

## 2.1 CRITICAL

Not Found

## 2.2 MAJOR

MJR-1	Correct migration
File	Strategy.sol
Severity	Major
Status	Fixed at bbecd326

### DESCRIPTION

In constructor, rights are granted to spend tokens, which should be canceled when migrating the strategy.

- Strategy.sol#L171

### RECOMMENDATION

It is recommended to add in function `prepareMigration()`:

```
IERC20(crv).safeApprove(address(want), 0);
IERC20(usdc).safeApprove(sushiswap, 0);

IyveCRV(address(want)).claim();
want.safeTransfer(_newStrategy, want.balanceOf(address(this)));

IERC20(usdc).safeTransfer(_newStrategy, IERC20(usdc).balanceOf(address(this)));

want.safeApprove(vault, 0);
vault.approve(rewards, 0);
```



## 2.3 WARNING

WRN-1	Missed token
File	Strategy.sol
Severity	Warning
Status	Fixed at bbecd326

### DESCRIPTION

At the line: `Strategy.sol#L266-L277`

The function `protectedTokens()` is used for removing tokens from this Strategy that are not the type of tokens managed by this Strategy. This may be used in case of accidentally sending the wrong kind of token to this Strategy. This will fail if an attempt is made to sweep `want`, or any tokens that are protected by this Strategy. But in this implementation the `yveCrv` token is not protected.

### RECOMMENDATION

It is recommended to replace `want` token for `yveCrv` in the `protected[]` array.

<b>WRN-2</b>	The approval value obtained in the constructor may not be enough for the long term of the smart contract
<b>File</b>	Strategy.sol
<b>Severity</b>	Warning
<b>Status</b>	Fixed at bbecd326

## DESCRIPTION

Smart contracts call `safeApprove()` functions 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. It is at the following lines:

- `Strategy.sol#L96-L97`

## RECOMMENDATION

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

## 2.4 COMMENTS

<b>CMT-1</b>	There is no input parameter processing in the method
<b>File</b>	Strategy.sol
<b>Severity</b>	Comment
<b>Status</b>	Acknowledged

### DESCRIPTION

At the lines:

- `Strategy.sol#L151`  
the `adjustPosition()` method has an input variable `_debtOutstanding`.  
But there is no processing of this variable in the body of the function.

### RECOMMENDATION

It is recommended to either add handling to the variable or remove this variable.

<b>CMT-2</b>	Event is probably missing
<b>File</b>	Strategy.sol
<b>Severity</b>	Comment
<b>Status</b>	Fixed at bbecd326

## DESCRIPTION

At the lines: `Strategy.sol#L259`  
in method `setBuffer` should probably emit an event `SetBuffer`.

## RECOMMENDATION

It is recommended to create new event.

# 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://github.com/mixbytes/audits_public)



<https://mixbytes.io/>



[hello@mixbytes.io](mailto:hello@mixbytes.io)



<https://t.me/MixBytes>



<https://twitter.com/mixbytes>