C.R.E.A.M. FINANCE FLASHLOAN SMART CONTRACT AUDIT

March 12, 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

C.R.E.A.M. Finance is a blockchain agnostic, decentralized peer to peer lending platform based on a fork of Compound Finance.

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 reviewed scope includes flashloan functionality implementation in `CCapableErc20` contract of C.R.E.A.M. finance project.

1.5 PROJECT DASHBOARD

Client	Yearn
Audit name	C.R.E.A.M. Finance Flashloan
Initial version	e414160eb8a774140457c885bb099baae528df04
Final version	49ae515f9edb1338ec8eed8077ba6592c20a5570
SLOC	120
Date	2021-03-11 - 2021-03-12
Auditors engaged	2 auditors

FILES LISTING

CCapableErc20.sol	CCapableErc20.sol

FINDINGS SUMMARY

Level	Amount
Critical	0
Major	1
Warning	1
Comment	1

CONCLUSION

Smart contracts have been audited and several suspicious places were found. During audit one major issue was identified as it could lead to some undesired behavior also two issues were marked as warning and comment. After working on audit report all issues were fixed or acknowledged(if issue is not critical) by client or concluded as not an issue.Final commit identifier with all fixes: `49ae515f9edb1338ec8eed8077ba6592c20a5570`

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

MJR-1	Excess reserve amount
File	CCapableErc20.sol
Severity	Major
Status	No issue

DESCRIPTION

At lines CCapableErc20.sol#L250-L252 contract increases `internalCash` and `totalReserves` values, but it's so strange that `internalCash` increased by `totalFee` and `totalReserves` increased by `reservesFee` so we totally increased assets amount by `reservesFee + totalFee` however user paid only `totalFee`. It seems there are some uncollateralized `reservesFee`. May be there `totalFee` paid by user should be splitted to `internalCash` and `totalReserves` ?

RECOMMENDATION

We recommend to double check that place

CLIENT'S COMMENTARY

By CToken's design, totalReserves is included in internalCash too.

2.3 WARNING

WRN-1	Suspicious manipulation of `totalReserves` and `internalCash`
File	CCapableErc20.sol
Severity	Warning
Status	Fixed at 9f4a9223

DESCRIPTION

At the line CCapableErc20.sol#L132 contract have `gulp` function that change `totalReserves` and `internalCash` variables. That function can be re-entered from while flashloan execution. For now, in `flashloan` contract captured affected variables and also `accrueInterest` depends on it, to be honest it's really precarious place.

RECOMMENDATION

We recommend to double check this and add re-entrancy guard to `gulp` function

2.4 COMMENTS

CMT-1	Magic number used
File	CCapableErc20.sol
Severity	Comment
Status	Fixed at 49ae515f

DESCRIPTION

At the line CCapableErc20.sol#L237 contract uses "magic number" `10000` to calculate `totalFee`.

RECOMMENDATION

We recommend to use constant instead or add comments

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