

C.R.E.A.M Finance Liquidity Mining SMART CONTRACT AUDIT

September 20, 2021

MixBytes()

CONTENTS

1.INTRODUCTION	2
DISCLAIMER	2
SECURITY ASSESSMENT METHODOLOGY	3
EXECUTIVE SUMMARY	5
PROJECT DASHBOARD	5
2.FINDINGS REPORT	7
2.1.CRITICAL	7
2.2.MAJOR	7
2.3.WARNING	7
2.4.COMMENT	7
CMT-1 No reporting on failed claimRewards attempt	7
3.ABOUT MIXBYTES	8

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 C.R.E.A.M Finance. 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 SECURITY ASSESSMENT METHODOLOGY

A group of 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 Project architecture review:
 - > Reviewing project documentation
 - > General code review
 - > Reverse research and study of the architecture of the code based on the source code only
 - > Mockup prototyping

Stage goal:
Building an independent view of the project's architecture and identifying logical flaws in the code.
- 02 Checking the code against the checklist of known vulnerabilities:
 - > 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
 - > Checking with static analyzers (i.e Slither, Mythril, etc.)

Stage goal:
Eliminate typical vulnerabilities (e.g. reentrancy, gas limit, flashloan attacks, etc.)
- 03 Checking the code for compliance with the desired security model:
 - > 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
 - > Exploits PoC development using Brownie

Stage goal:
Detection of inconsistencies with the desired model
- 04 Consolidation of interim auditor reports into a general one:
 - > 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 and 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.

1.3 EXECUTIVE SUMMARY

C.R.E.A.M. Finance is a decentralized lending protocol for individuals and protocols to access financial services. The protocol is permissionless, transparent, and non-custodial. C.R.E.A.M. Finance's team modularized the liquidity mining reward feature into a separate contract and extend it to able handle multiple token reward for a market.

1.4 PROJECT DASHBOARD

Client	C.R.E.A.M Finance
Audit name	Liquidity Mining
Initial version	e7caa316e8a206b60996c199f26047b388b4ea31
Final version	2e70f61fc7645442b9db66c03e1529d42d25a0af
Date	August 30, 2021 - September 20, 2021
Auditors engaged	3 auditors

FILES LISTING

LiquidityMiningLens.sol	https://github.com/CreamFi/liquidity-mining/blob/e7caa316e8a206b60996c199f26047b388b4ea31/contracts/LiquidityMiningLens.sol
LiquidityMiningStorage.sol	https://github.com/CreamFi/liquidity-mining/blob/e7caa316e8a206b60996c199f26047b388b4ea31/contracts/LiquidityMiningStorage.sol
LiquidityMining.sol	https://github.com/CreamFi/liquidity-mining/blob/e7caa316e8a206b60996c199f26047b388b4ea31/contracts/LiquidityMining.sol
CTokenInterface.sol	https://github.com/CreamFi/liquidity-mining/blob/e7caa316e8a206b60996c199f26047b388b4ea31/contracts/interfaces/CTokenInterface.sol
ComptrollerInterface.sol	https://github.com/CreamFi/liquidity-mining/blob/e7caa316e8a206b60996c199f26047b388b4ea31/contracts/interfaces/ComptrollerInterface.sol

LiquidityMiningInterface.sol

<https://github.com/CreamFi/liquidity-mining/blob/e7caa316e8a206b60996c199f26047b388b4ea31/contracts/interfaces/LiquidityMiningInterface.sol>

FINDINGS SUMMARY

Level	Amount
Critical	0
Major	0
Warning	0
Comment	1

CONCLUSION

During the audit no critical, major or warning issues were found, one comment were spotted. After working on the reported findings it was fixed by the client. So, the contracts are assumed as secure to use according to our security criteria. Final commit identifier with all fixes: [2e70f61fc7645442b9db66c03e1529d42d25a0af](#)

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

Not Found

2.3 WARNING

Not Found

2.4 COMMENT

CMT-1	No reporting on failed claimRewards attempt
File	LiquidityMining.sol
Severity	Comment
Status	Fixed at LiquidityMining.sol

DESCRIPTION

While a user is in debtors list, any attempt to claim rewards will be [LiquidityMining.sol#L469](#).

RECOMMENDATION

It is recommended to report the user that claim action has been failed because of bad debt.

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>