

AAVE INCENTIVES PROPOSAL SMART CONTRACT AUDIT

August 06, 2021

MixBytes()

CONTENTS

1. INTRODUCTION	2
DISCLAIMER	2
PROJECT OVERVIEW	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 Zero values	7
CMT-2 Local variable shadowing	8
3. ABOUT MIXBYTES	9

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 Aave. 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

The audited scope of contracts manages the incentives related with liquidity on the Aave Protocol.

1.3 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.

1.4 EXECUTIVE SUMMARY

The two smart contracts tested in this audit are designed for the logic for managing user rewards. Every time a user does actions on the protocol, e.g. supplying liquidity or borrowing - the `handleAction()` function is triggered, which takes into account the reward for this user.

1.5 PROJECT DASHBOARD

Client	Aave
Audit name	Incentives Proposal
Initial version	f6712e33db79210a7ae8106f7cfa1ce2adea8d69
Final version	-
Date	July 19, 2021 - August 06, 2021
Auditors engaged	2 auditors

FILES LISTING

DistributionManager.sol	https://github.com/aave/incentives-proposal/blob/f6712e33db79210a7ae8106f7cfa1ce2adea8d69/contracts/incentives/DistributionManager.sol
StakedTokenIncentivesController.sol	https://github.com/aave/incentives-proposal/blob/f6712e33db79210a7ae8106f7cfa1ce2adea8d69/contracts/incentives/StakedTokenIncentivesController.sol
IAaveDistributionManager.sol	https://github.com/aave/incentives-proposal/blob/f6712e33db79210a7ae8106f7cfa1ce2adea8d69/contracts/interfaces/IAaveDistributionManager.sol
DistributionTypes.sol	https://github.com/aave/incentives-proposal/blob/f6712e33db79210a7ae8106f7cfa1ce2adea8d69/contracts/lib/DistributionTypes.sol
SafeMath.sol	https://github.com/aave/incentives-proposal/blob/f6712e33db79210a7ae8106f7cfa1ce2adea8d69/contracts/lib/SafeMath.sol

IAaveIncentivesController.sol	https://github.com/aave/incentives-proposal/blob/f6712e33db79210a7ae8106f7cfa1ce2adea8d69/contracts/interfaces/IAaveIncentivesController.sol
IScaledBalanceToken.sol	https://github.com/aave/incentives-proposal/blob/f6712e33db79210a7ae8106f7cfa1ce2adea8d69/contracts/interfaces/IScaledBalanceToken.sol
IStakedTokenWithConfig.sol	https://github.com/aave/incentives-proposal/blob/f6712e33db79210a7ae8106f7cfa1ce2adea8d69/contracts/interfaces/IStakedTokenWithConfig.sol

FINDINGS SUMMARY

Level	Amount
Critical	0
Major	0
Warning	0
Comment	2

CONCLUSION

Smart contracts have been audited and several suspicious places have been spotted. During the audit no critical or major issues were found, two issues were marked as comments and discussed with the client. After working on the reported findings all of them were acknowledged (as the problems were not critical). So, the contracts are assumed as secure to use according to our security criteria. Final commit identifier with all fixes: `-`

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

Not Found

2.3 WARNING

Not Found

2.4 COMMENT

CMT-1	Zero values
File	StakedTokenIncentivesController.sol DistributionManager.sol
Severity	Comment
Status	Acknowledged

DESCRIPTION

No check for zero values at constructor for parameters `stakeToken` `StakedTokenIncentivesController.sol#L45` and `emissionManager` `DistributionManager.sol#L37`

RECOMMENDATION

We recommend to check for zero values

CMT-2	Local variable shadowing
File	DistributionManager.sol StakedTokenIncentivesController.sol
Severity	Comment
Status	Acknowledged

DESCRIPTION

At the line

- `DistributionManager.sol#L28`
and the lines:
- `StakedTokenIncentivesController.sol#L57`
- `StakedTokenIncentivesController.sol#L92`
- `StakedTokenIncentivesController.sol#L113`
- `StakedTokenIncentivesController.sol#L123`
- `StakedTokenIncentivesController.sol#L134`
- `StakedTokenIncentivesController.sol#L186`

detection of shadowing using `assets` local variables.

RECOMMENDATION

We recommend renaming the local variables that shadow another component.

CLIENT'S COMMENTARY

Acknowledge, but no action at the moment.

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>