LIDO FINANCE WITHDRAWALS MANAGER STUB SMART CONTRACT AUDIT

May 24, 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 Lido 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 PROJECT OVERVIEW

LIDO protocol is a project for stacking Ether to use it in Beacon chain. Users can deposit Ether to the Lido smart contract and receive stETH tokens in return. The stETH token balance corresponds to the amount of Beacon chain Ether that the holder could withdraw if state transitions were enabled right now in the Ethereum 2.0 network.

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

Withdrawal manager project solves the following problem. Though the Beacon chain already supports setting withdrawal credentials pointing to a smart contract, the withdrawals specification is not yet final and might change before withdrawals are enabled in the Merge network. This means that Lido cannot deploy the final implementation of the withdrawals manager contract yet. At the same time, it's desirable to have withdrawal credentials pointing to a smart contract since this would avoid the need to migrate a lot of validators to new withdrawal credentials once withdrawals are enabled.
The WithdrawalsManagerProxy is proxy contract with a built in admin and upgrade the interface of the WithdrawalsManagerStub.The upgradeability mechanism is based of secure openZeppelin implementation based on ERC1967 proxy. In this scope the WithdrawalsManagerStub contract have very simple implementation and is inherently a stub.

1.5 PROJECT DASHBOARD

Client	Lido Finance
Audit name	Withdrawals manager stub
Initial version	c41292ed9c3be765d06c4249b9f2ad4d427b84bf 214d4773648134f970509bfe37184aee3aff4d24
Final version	214d4773648134f970509bfe37184aee3aff4d24
SLOC	59
Date	2021-05-14 - 2021-05-24
Auditors engaged	2 auditors

FILES LISTING

WithdrawalsManagerProxy.sol	WithdrawalsManagerPro...
WithdrawalsManagerStub.sol	WithdrawalsManagerStu...

FINDINGS SUMMARY

Level	Amount
Critical	0
Major	0
Warning	0
Comment	1

CONCLUSION

Smart contract has been audited one suspicious place has been spotted. During the audit no critical or major issues were found, one issue was marked comment. After working on the reported finding it was fixed by the client. So, the contract is assumed as secure to use according to our security criteria.Final commit identifier with all fixes: `214d4773648134f970509bfe37184aee3aff4d24`

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

Not Found

2.3 WARNING

Not Found

2.4 COMMENTS

CMT-1	Unnecessary check in initialization
File	WithdrawalsManagerProxy.sol
Severity	Comment
Status	Fixed at 214d4773

DESCRIPTION

Constructor of the proxy contains unnecessary check:
WithdrawalsManagerProxy.sol#L70

RECOMMENDATION

We recommend to delete this check.

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