LIDO EASY TRACK SMART CONTRACT AUDIT

September 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

WRN-1 Duplicate voting functionality 7

WRN-2 Missing address validation 8

2.4.COMMENT 9

CMT-1 No valid params 9

3.ABOUT MIXBYTES 10

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. 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

Easy Track motion is a lightweight voting considered to have passed if the minimum objections threshold hasn't been exceeded. As opposed to traditional Aragon votings, Easy Track motions are cheaper (no need to vote 'pro', token holders only have to vote 'contra' if they have objections) and easier to manage (no need to ask broad DAO community vote on proposals sparking no debate). Detailed specifications of Easy Track can be found here.
More in-depth feauture description and possible use cases can be found in the official LIP-3 (WIP). Community discussion takes place on Lido research forum: announcement post, v2 post.

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

Easy Track is strongly inspired by Aragon's Voting and based on Aragon's EVMScripts concept. Execution of EVMScripts is performed by standalone EVMScriptExecutor contract, which can be called only by EasyTrack and Aragon's Voting contracts. Implementation of EVMScriptExecutor used in EasyTrack contract delegates execution of EVMScripts to Aragon's CallsScript executor.As opposed to Aragon's Voting, EasyTrack contract doesn't allow to pass EVMScripts directly, and uses standalone EVMScript factory contracts to create EVMScripts. EVMScript factory - is a special contract, which implements IEVMScriptFactory interface. Each EVMScript factory has to be registered in the EasyTrack contract before it can be used for motion creation. Registration of EVMScript factory contracts is allowed only to Admins of Easy Track.To enhance the security of Easy Track, each EVMScript factory has its own Permissions set when a new EVMScript factory is being registered in the EasyTrack contract. Permissions is a list of tuples (address, bytes4) encoded into a bytes representation. Each tuple (address, bytes4) describes a method allowed to be called by EVMScript generated by the corresponding EVMScript factory. EasyTrack validates each EVMScript to satisfy permissions and reverts transaction if EVMScript tries to call a method not listed in its permissions.

1.5 PROJECT DASHBOARD

Client	Lido
Audit name	Easy Track
Initial version	ec694adb872877db814da960d96ce767ccbdf462
Final version	7acdfe0cc9d0f2fc34b03e094c8225c0c9c659a3
Date	August 03, 2021 - September 06, 2021
Auditors engaged	2 auditors

FILES LISTING

ContractProxy.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/ContractProxy.sol
EVMScriptFactoriesRegistry.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/EVMScriptFactoriesRegistry.sol

EasyTrack.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/EasyTrack.sol
EasyTrackStorage.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/EasyTrackStorage.sol
EvmScriptExecutor.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/EvmScriptExecutor.sol
MotionSettings.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/MotionSettings.sol
RewardProgramsRegistry.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/RewardProgramsRegistry.sol
TrustedCaller.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/TrustedCaller.sol
AddRewardProgram.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/EVMScriptFactories/AddRewardProgram.sol
IncreaseNodeOperatorStakingLimit.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/EVMScriptFactories/IncreaseNodeOperatorStakingLimit.sol
RemoveRewardProgram.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/EVMScriptFactories/RemoveRewardProgram.sol
TopUpLegoProgram.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/EVMScriptFactories/TopUpLegoProgram.sol
TopUpRewardPrograms.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/EVMScriptFactories/TopUpRewardPrograms.sol
BytesUtils.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/libraries/BytesUtils.sol
EVMScriptCreator.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/libraries/EVMScriptCreator.sol

EVMScriptPermissions.sol	https://github.com/lidofinance/easy-track/blob/ec694adb872877db814da960d96ce767ccbdf462/contracts/libraries/EVMScriptPermissions.sol

FINDINGS SUMMARY

Level	Amount
Critical	0
Major	0
Warning	2
Comment	1

CONCLUSION

Smart contracts have been audited and several suspicious places were found. During the audit no critical or major issues were identified. Several issues were marked as warnings and comments. After working on audit report all issues were fixed or acknowledged by the client (if the problem was not critical). Thus, contracts are assumed as secure to use according to our security criteria.Final commit identifier with all fixes: `7acdfe0cc9d0f2fc34b03e094c8225c0c9c659a3`

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

Not Found

2.3 WARNING

WRN-1	Duplicate voting functionality
File	EvmScriptExecutor.sol
Severity	Warning
Status	Fixed at 7acdfe0c

DESCRIPTION

EVMScriptExecutor has permissions for EasyTrack and Voting contracts to make financial transactions (finance contract). It is possible to create voting in this contracts at the same time. EasyTrack has permissions only for TrustedCaller. In voting contract every token holder able to do it and create Voting with own purposes
at line EvmScriptExecutor.sol#L71

RECOMMENDATION

We recommend to split up functionality between EasyTrack and Voting contacts

WRN-2	Missing address validation
File	EvmScriptExecutor.sol
Severity	Warning
Status	Fixed at 7acdfe0c

DESCRIPTION

A delegatecall on an callsScript address always returns true if callsScript is non-contract address.
At line EvmScriptExecutor.sol#L80

RECOMMENDATION

We recommend making the callsScript address constant or add check that callsScript is contract address in constructor.

2.4 COMMENT

CMT-1	No valid params
File	MotionSettings.sol
Severity	Comment
Status	No Issue

DESCRIPTION

There is some admin methods which don't check values:

- `setMotionDuration` - `_motionDuration` may be too large MotionSettings.sol#L23
- `setMotionsCountLimit` — `_motionsCountLimit` may be zero MotionSettings.sol#L40
  This allows admin to make an incorrect configuration.

RECOMMENDATION

If necessary, we recommend to insert additional checks.

CLIENT'S COMMENTARY

Acknowledged, we don't consider this a problem since motions are easy to object and also the complete Easy Track feature can be paused outright.
for _motionsCountLimit - design is intended, since setting limit to zero allows completing all the ongoing motions and prevents creating new ones.

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