LIDO ANCHOR COLLATERAL STETH SMART CONTRACT AUDIT

December 29, 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. 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 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.
The Lido DAO is a Decentralized Autonomous Organization that manages the liquid staking protocol by deciding on key
parameters (e.g., setting fees, assigning node operators and oracles, etc.) through the voting power of governance token (DPG) holders.
The Lido DAO is an Aragon organization. The protocol smart contracts extend AragonApp base contract and can be managed by the DAO.Anchor Vault is a smart contract that allows to convert rebasing stETH token into a constant-balance bETH token and periodically send all accrued stETH rewards to the Terra blockchain through a bridge. The bETH token is used as a
collateral in the Terra Anchor protocol.
`RewardsLiquidator`, `InsuranceConnector` and `BridgeConnectorWormhole` contracts are installed as delegates to the `AnchorVault`
contract and are used by the latter for performing various tasks (see the description above).
These contracts can be replaced by the vault admin.
The `BridgeConnectorWormhole` contract is designed to interoperate with the Wormhole Ethereum-Terra bridge.Contracts:

- `AnchorVault` - the main vault contract
- `AnchorVaultProxy` - proxy contract for AnchorVault
- `bEth` - bETH token contract
- `RewardsLiquidator` - a contract for selling stETH rewards to UST
- `InsuranceConnector` - a contract for obtaining the total number of shares burnt for the purpose of insurance/cover application from the Lido protocol
- `BridgeConnectorWormhole` - an adapter contract for communicating with the Wormhole bridge

1.4 PROJECT DASHBOARD

Client	Lido
Audit name	Anchor Collateral stETH
Initial version	8d52ce72cb42d48dff1851222e3b624c941ddb30
Final version	8d52ce72cb42d48dff1851222e3b624c941ddb30
Date	December 07, 2021 - December 29, 2021
Auditors engaged	4 auditors

FILES LISTING

AnchorVault.vy	https://github.com/lidofinance/anchor-collateral-steth/blob/8d52ce72cb42d48dff1851222e3b624c941ddb30/contracts/AnchorVault.vy
AnchorVaultProxy.sol	https://github.com/lidofinance/anchor-collateral-steth/blob/8d52ce72cb42d48dff1851222e3b624c941ddb30/contracts/AnchorVaultProxy.sol
bEth.vy	https://github.com/lidofinance/anchor-collateral-steth/blob/8d52ce72cb42d48dff1851222e3b624c941ddb30/contracts/bEth.vy
RewardsLiquidator.vy	https://github.com/lidofinance/anchor-collateral-steth/blob/8d52ce72cb42d48dff1851222e3b624c941ddb30/contracts/RewardsLiquidator.vy
InsuranceConnector.vy	https://github.com/lidofinance/anchor-collateral-steth/blob/8d52ce72cb42d48dff1851222e3b624c941ddb30/contracts/InsuranceConnector.vy
BridgeConnectorWormhole.vy	https://github.com/lidofinance/anchor-collateral-steth/blob/8d52ce72cb42d48dff1851222e3b624c941ddb30/contracts/BridgeConnectorWormhole.vy

FINDINGS SUMMARY

Level	Amount
Critical	0
Major	1
Warning	4
Comment	2

CONCLUSION

Smart contracts have been audited and several suspicious places have been detected. During the audit no critical issues were found, one major, several warnings and comments were spotted. After working on the reported findings all of them were acknowledged by the client.Final commit identifier with all fixes: `8d52ce72cb42d48dff1851222e3b624c941ddb30`

CONTRACTS DEPLOYMENT

The following addresses contain deployed to the Ethereum mainnet and verified smart contracts code that matches audited scope:

Contract name | Deployed link

– | –
AnchorVaultProxy | https://etherscan.io/address/0xa2f987a546d4cd1c607ee8141276876c26b72bdf#code
AnchorVault.vy | https://etherscan.io/address/0x0627054d17eae63ec23c6d8b07d8db7a66ffd45a#code
bEth.vy | https://etherscan.io/address/0x707f9118e33a9b8998bea41dd0d46f38bb963fc8#code
RewardsLiquidator.vy | https://etherscan.io/address/0x082a5956d63b44685a7cca89379d565c439fdf3c#code
InsuranceConnector.vy | https://etherscan.io/address/0x2bdfd3de0ff23373b621cdad0ad3df1580efe701#code

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

MJR-1	Basic functions do not work
File	BridgeConnectorWormhole.vy
Severity	Major
Status	Acknowledged

DESCRIPTION

Line

- Bridge.sol#L93
  has a `transferTokens ()` function of type `payable`. In the body of this function, on line 133, a call to the internal
  function `logTransfer ()` is made and one of the parameters `msg.value` is passed.
  At line
- Bridge.sol#L151
  from the `logTransfer()` function, the `publishMessage()` function is called.
  The `publishMessage()` function of type `payable` on the line:
- Implementation.sol#L21
  the condition for payment of the commission must be met.

```
require(msg.value == messageFee(), "invalid fee");
```

In the checked contract, at line

- BridgeConnectorWormhole.vy#L49
  a call to the `transferTokens()` function from the `_transfer_asset()` function is made.
  But there is no `@payable` modifier anywhere and no `msg.value` value handling.
  Therefore, the `_transfer_asset()` function will not work.
  As a result, the `submit ()` and `collect_rewards ()` functions will not work in the `AnchorVault.vy` contract,
  because there is no commission fee for the `Wormhole` system.

RECOMMENDATION

It is required to add a commission payment for the `Wormhole` system.

CLIENT'S COMMENTARY

The issue is acknowledged. The `messageFee` is currently set to 0 in bridge settings. The Wormhole team stated that "there is no plan or timeline at the moment" for changing the fee value from zero. The `BridgeConnectorWormhole` is changed to support non-zero fee. Note that to accomodate to non-zero fee in Wormhole changes to AnchorVault would be required as well.

2.3 WARNING

WRN-1	There is no processing of the value returned by the function.
File	AnchorVault.vy RewardsLiquidator.vy BridgeConnectorWormhole.vy
Severity	Warning
Status	Acknowledged

DESCRIPTION

In the ERC-20 standard, when processing tokens, if successful, `true` is returned.
And this value should always be checked.
But in the audited code, checks are not done.
The following lines:

- AnchorVault.vy#L381
- AnchorVault.vy#L383
- AnchorVault.vy#L414
- AnchorVault.vy#L488
- RewardsLiquidator.vy#L278
- RewardsLiquidator.vy#L321
- RewardsLiquidator.vy#L345
- BridgeConnectorWormhole.vy#L44

RECOMMENDATION

It is recommended to add processing of the value returned by the function.

CLIENT'S COMMENTARY

The warning is considered no issue. `ERC20` calls in `AnchorVault`, `RewardsLiquidator` target `StETH`, `Wrapped UST` & `USDC` token contracts, which either returns `true` or revert. Check for returned value of `approve` call in `BridgeConnectorWormhole` is added.

WRN-2	No checking value for zero
File	AnchorVault.vy
Severity	Warning
Status	Acknowledged

DESCRIPTION

There is a possible scenario where collected rewards may be sent to zero address if `rewards_distributor` is uninitialized
AnchorVault.vy#L491

RECOMMENDATION

It is recommended to check if `rewards_distributor` is uninitialized.

CLIENT'S COMMENTARY

The warning is considered no issue. The contract `AnchorVault` is deployed https://etherscan.io/address/0xA2F987A546D4CD1c607Ee8141276876C26b72Bdf & the `rewards_distributor` field is initialized.

WRN-3	Adjusted stETH return transfer fee may be more expensive than dust amount
File	AnchorVault.vy
Severity	Warning
Status	Acknowledged

DESCRIPTION

`steth_amount - steth_amount_adj` value may be less than `transfer` fee cost.
AnchorVault.vy#L383
The same situation is there for the variable `steth_to_sell` at line:
AnchorVault.vy#L488

RECOMMENDATION

It is recommended to calculate if there is a sufficient difference between the adjusted eth and original amount to return.
You can add a check for the minimum value.

CLIENT'S COMMENTARY

The warning is considered no issue. The calculation would increase the gas costs of user operations for the `AnchorVault`, while not contributing to the safety or usability of the integration.

WRN-4	An unfavorable exchange may occur
File	RewardsLiquidator.vy
Severity	Warning
Status	Acknowledged

DESCRIPTION

At the lines:
RewardsLiquidator.vy#L306-L308
and
RewardsLiquidator.vy#L329-L331
if the initial USDC or UST balance + exchanged tokens are bigger than min required swap amount then an unfavorable swap may occur.
Actually, it is safe for the project.

RECOMMENDATION

It is recommended to calc swapped amount with difference balance from and balance after.
This is not a planned behavior that needs to be handled.

CLIENT'S COMMENTARY

The warning is considered no issue. If the exchange route would turn unfavorable for the integration, another implementation of the `RewardsLiquidator` could be developed & deployed.

2.4 COMMENT

CMT-1	The function can be decomposed
File	RewardsLiquidator.vy
Severity	Comment
Status	Acknowledged

DESCRIPTION

At the line
RewardsLiquidator.vy#L262-L357
`liquidate()` function can be decomposed to three internal functions.

RECOMMENDATION

It is recommended to make three separate internal functions instead of one long function.

CLIENT'S COMMENTARY

The comment is considered no issue. Current implementation is optimized for lower gas consumption.

CMT-2	An unnecessary comment
File	AnchorVault.vy
Severity	Comment
Status	Acknowledged

DESCRIPTION

At the line
AnchorVault.vy#L164
it is an unnecessary comment `# dev: unauthorized`.

RECOMMENDATION

It is recommended to delete this comment.

CLIENT'S COMMENTARY

The comment is considered no issue. The `AnchorVault` contract is deployed https://etherscan.io/address/0xA2F987A546D4CD1c607Ee8141276876C26b72Bdf, and the comment could be cleared out on the next implementation migration.

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