STADER LABS ETHX SECURITY AUDIT REPORT

Jul 11, 2024

MixBytes()

TABLE OF 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 the Client. 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. The security engineers check the provided source code independently of each other in accordance with the methodology described below:

1. Project architecture review:

- Project documentation review.
- General code review.
- Reverse research and study of the project architecture on the source code alone.

Stage goals
- Build an independent view of the project's architecture.
- Identifying logical flaws.

2. Checking the code in accordance with the vulnerabilities checklist:

- Manual code check for vulnerabilities listed on the Contractor's internal checklist. The Contractor's checklist is constantly updated based on the analysis of hacks, research, and audit of the clients' codes.
- Code check with the use of static analyzers (i.e Slither, Mythril, etc).

Stage goal

Eliminate typical vulnerabilities (e.g. reentrancy, gas limit, flash loan attacks etc.).

3. Checking the code for compliance with the desired security model:

- Detailed study of the project documentation.
- Examination of contracts tests.
- Examination of comments in code.
- Comparison of the desired model obtained during the study with the reversed view obtained during the blind audit.
- Exploits PoC development with the use of such programs as Brownie and Hardhat.

Stage goal

Detect inconsistencies with the desired model.

4. Consolidation of the auditors' interim reports into one:

- Cross check: each auditor reviews the reports of the others.
- Discussion of the issues found by the auditors.
- Issuance of an interim audit report.

Stage goals

- Double-check all the found issues to make sure they are relevant and the determined threat level is correct.
- Provide the Client with an interim report.

5. Bug fixing & re-audit:

- The Client either fixes the issues or provides comments on the issues found by the auditors. Feedback from the Customer must be received on every issue/bug so that the Contractor can assign them a status (either "fixed" or "acknowledged").
- Upon completion of the bug fixing, the auditors double-check each fix and assign it a specific status, providing a proof link to the fix.
- A re-audited report is issued.

Stage goals

- Verify the fixed code version with all the recommendations and its statuses.
- Provide the Client with a re-audited report.

6. Final code verification and issuance of a public audit report:

- The Customer deploys the re-audited source code on the mainnet.
- The Contractor verifies the deployed code with the re-audited version and checks them for compliance.
- If the versions of the code match, the Contractor issues a public audit report.

Stage goals

- Conduct the final check of the code deployed on the mainnet.
- Provide the Customer with a public audit report.

Finding Severity breakdown

All vulnerabilities discovered during the audit are classified based on their potential severity and have the following classification:

Severity	Description
Critical	Bugs leading to assets theft, fund access locking, or any other loss of funds.
High	Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement.
Medium	Bugs that can break the intended contract logic or expose it to DoS attacks, but do not cause direct loss funds.
Low	Bugs that do not have a significant immediate impact and could be easily fixed.

Based on the feedback received from the Customer 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 Customer is aware of the finding. Recommendations for the finding are planned to be resolved in the future.

1.3 Project Overview

ETHx leverages the Omnichain Fungible Token (OFT) implementation provided by LayerZero Labs. This implementation offers a standardized and efficient approach to cross-chain token transfers.

The ETHx OFT facilitates cross-chain transfers of ETHx tokens by employing a "lock and mint" mechanism. On the origin chain, where the ETHx token contract is already deployed, tokens are locked within the ETHx_OFTAdapter contract during the transfer process. Upon reaching the destination chain, an equivalent amount of ETHx tokens are minted for the recipient, ensuring a seamless transfer of value between chains while maintaining the integrity of the token supply.

1.4 Project Dashboard

Project Summary

Title	Description
Client	Stader Labs
Project name	ETHx
Timeline	08.04.2024 - 07.06.2024
Number of Auditors	3

Project Log

Date	Commit Hash	Note
08.04.2024	183c0615c98a88c0c4cb2fa089e9aed928ab9bae	Initial commit
08.05.2024	6a4ffaddfd6924dbe29017bc6530b372e311335d	Commit for the re-audit

Project Scope

The audit covered the following files:

File name	Link
contracts/ETHx.sol	ETHx.sol
contracts/ETHx_OFT.sol	ETHx_OFT.sol
contracts/ETHx_OFTAdapter.sol	ETHx_OFTAdapter.sol

Deployments

Ethereum:mainnet

File name	Contract deployed on mainnet	Comment
ETHx_OFTAdapter	0x39d0ce...820665E9	

Arbitrum:mainnet

File name	Contract deployed on mainnet	Comment
ETHx	0xED65C5...5E944Dc7	proxy
ETHx	0x778714...dbf09a6d	implementation
ETHx_OFT	0x690460...B935C43C	

Optimism:mainnet

File name	Contract deployed on mainnet	Comment
ETHx	0xc54B43eaF921A5194c7973A4d65E055E5a1453c2	proxy
ETHx	0xbe23e1a64969cb28efdb6c3d2ce9e4bf16042187	implementation
ETHx_OFT	0x01aF04690d17DC27b891A7F67E9EEe4d14DE8EA8	

1.5 Summary of findings

Severity	# of Findings
Critical	0
High	0
Medium	1
Low	1

ID	Name	Severity	Status
M-1	Lack of pause functionality within the OFT Adapter	Medium	Acknowledged
L-1	Lack of caps for minting on the remote chain	Low	Acknowledged

1.6 Conclusion

This audit evaluated the LayerZero integration enabling cross-chain functionality for the ETHx token. The key components reviewed include:

- **ETHx Token:** Implementation of the ERC-20 token standard on sidechains.
- **ETHx_OFT, ETHx_OFTAdapter:** LayerZero applications for sending and receiving ETHx tokens on sidechains and the mainnet, respectively.

These contracts utilize the recommended LayerZero `OFT` and `OFTAdapter` libraries.

Security considerations and potential attack vectors

- **Security inheritance risk:** The overall system security inherits the weakest link from the connected blockchains. A sidechain compromise could potentially impact the entire system.
- **Centralization risks:**
  - **Excessive minting:** A malicious owner could mint unauthorized `ETHx` tokens on sidechains independently of the `ETHx_OFT` contract, exceeding the intended total supply. This can be mitigated by implementing a multi-signature or timelock mechanism for crucial governance functions.
  - **Peer address manipulation:** The owner could configure the `peer` address to an unauthorized recipient, leading to transaction reverts on the receiving side and potential unauthorized minting. Implementing a multi-signature or timelock mechanism for governance functions can also address this risk.
  - **Sidechain token pause:** Pausing the `ETHx` token on a sidechain results in reverts when receiving cross-chain messages, thus locking funds on the sending side and leaving them unreceived until the token is unpaused. To prevent this, it's recommended to pause sending tokens to the sidechain before pausing the token itself.
- **Cross-chain validations:** The use of LayerZero's `OFTCore` and `OAppCore` libraries helps to mitigate this concern, as they include essential checks for cross-chain messaging.

2.FINDINGS REPORT

2.1 Critical

Not Found

2.2 High

Not Found

2.3 Medium

M-1	Lack of pause functionality within the OFT Adapter
Severity	Medium
Status	Acknowledged

Description

The transferring of tokens is performed in two steps: locking (or burning) tokens on the local chain and then minting (or releasing) the same amount of tokens on the remote chain. If an issue arises with the remote chain endpoint (such as paused minting), the local chain still allows `send` operations, leading to the locking/burning of assets without the ability to release them.

This situation is partly mitigated by the option to disable the corresponding peer in the sending contract. However, disabling the peer necessitates a more privileged access role (`OWNER`) compared to the role capable of pausing the contract on the remote chain.

Recommendation

We recommend implementing pausing on the OFT Adapter.

Client's commentary

> From our standpoint we are comfortable that there is a workaround for the situation in the rare case that it may arise. We also note that nearly every LayerZero token will be susceptible to this potential soft locking therefore it is not something specific to the Stader implementation but more an aspect of the L2 bridge architecture in general.

2.4 Low

L-1	Lack of caps for minting on the remote chain
Severity	Low
Status	Acknowledged

Description

If the ETHx token is compromised on one of the supported chains (i.e., a huge amount of tokens is minted), this issue may spread to other supported chains because the amount of transferred tokens is unlimited (except when the receiving side is the mainnet).

Recommendation

We recommend implementing caps on the amount of tokens that can be minted, based on the estimated volume of tokens circulating in the source chain.

Client's commentary

> We acknowledge the issue and agree it is cause for consideration. For now we would like to keep this off of our short term work schedule and then circle back to some type of limits in a future upgrade.

3. ABOUT MIXBYTES

MixBytes is a team of blockchain developers, auditors and analysts keen on decentralized systems. We build opensource solutions, smart contracts and blockchain protocols, perform security audits, work on benchmarking and software testing solutions, do research and tech consultancy.

Contacts

https://github.com/mixbytes/audits_public

https://mixbytes.io/

hello@mixbytes.io

https://twitter.com/mixbytes