

Security Audit Report

Mezo-Acre
stBTC Smart Contracts

Initial Report // July 29, 2024
Final Report // August 08, 2024

Team Members

Ahmad Jawid Jamiulahmadi // Senior Security Auditor
Mukesh Jaiswal // Senior Security Auditor
Bashir Abu-Amr // Head of Delivery

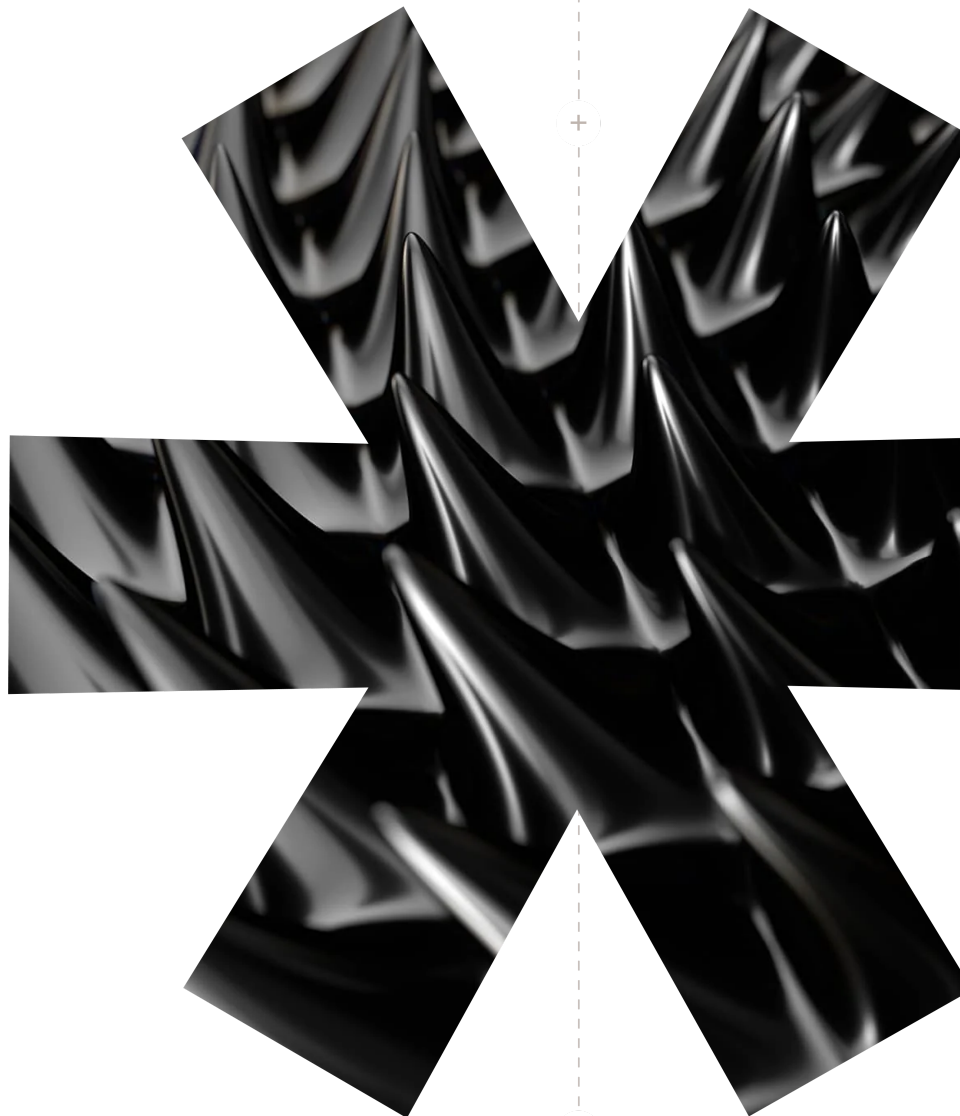


Table of Contents

<u>1.0 Scope</u>	3
↳ 1.1 Technical Scope	
↳ 1.2 Documentation	
<u>2.0 Executive Summary</u>	4
↳ 2.1 Schedule	
↳ 2.2 Overview	
↳ 2.3 Threat Model	
↳ 2.4 Security by Design	
↳ 2.5 Secure Implementation	
↳ 2.6 Use of Dependencies	
↳ 2.7 Tests	
↳ 2.8 Project Documentation	
<u>3.0 Key Findings Table</u>	6
<u>4.0 Findings</u>	7
↳ 4.1 User Can Mint Owed Fees as stBTC Resulting in Paying Zero Fees to the Portal	
^ High <input checked="" type="checkbox"/> Fixed	
↳ 4.2 Smart Contract Can Lose Track of feeCollected Due To Partial Withdrawals	
= Medium <input checked="" type="checkbox"/> Fixed	
↳ 4.3 Users Unable to Repay Debt When the stBTC Smart Contract is Paused	
= Medium <input checked="" type="checkbox"/> Not Fixed	
↳ 4.4 Incorrect Function Visibility Implemented in the mintDebt and repayDebt Functions	
✓ Low <input checked="" type="checkbox"/> Fixed	
↳ 4.5 If mintCap is 100 Percent, a User Can Mint 100 Percent of the Deposit Without Returning to the Portal	
∨ None <input checked="" type="checkbox"/> Not Fixed	
↳ 4.6 Implement Zero amount Checks	
∨ None <input checked="" type="checkbox"/> Fixed	
<u>5.0 Appendix A</u>	11
↳ 5.1 Severity Rating Definitions	
<u>6.0 Appendix B</u>	12
↳ 6.1 Thesis Defense Disclaimer	



About Thesis Defense

Thesis Defense serves as the auditing services arm within Thesis, Inc., the venture studio behind tBTC, Fold, Tahoe, Etcher, and Mezo. Our team of security auditors have carried out hundreds of security audits for decentralized systems across a number of technologies including smart contracts, wallets and browser extensions, bridges, node implementations, cryptographic protocols, and dApps. We offer our services within a variety of ecosystems including Bitcoin, Ethereum + EVMs, Stacks, Cosmos / Cosmos SDK, NEAR and more.

Thesis Defense will employ the Thesis Defense Audit Approach and Audit Process to the in scope service. In the event that certain processes and methodologies are not applicable to the in scope services, we will indicate as such in individual audit or design review SOWs. In addition, Thesis Defense provides clear guidance on successful Security Audit Preparation.

Section 1.0

Scope

Technical Scope

Acre

- **Repository:** <https://github.com/thesis/acre>
- **Audit Commit:** daf9b82a84e6995364a310f76219390cc026f543
- **Verification commit:** 45368f4f6b5a70c3626fbbd529082d57553a2299
- **Files in Scope:**
 - stBTC.sol
 - lib/ERC4626Fees.sol
 - lib/ERC4626NonFungibleWithdrawals.sol
 - MezoAllocator.sol

Mezo Portal

- **Repository:** <https://github.com/thesis/mezo-portal>
- **Audit Commit:** 0d86bd9b473d490d8d6041078abbb9934d08461e
- **Verification Commit:** 2878507ef2a374fbc03d3831832c3fa481557bdc
- **Files in Scope:**
 - Portal.sol

Documentation

- [Product pitch for stBTC](#)
- [Mezo Repository Documentation](#)



Executive Summary

Schedule

This security audit was conducted from July 18, 2024 to July 29, 2024 by 2 senior security auditors for a total of 3 person-weeks.

Overview

This report details the findings and outcomes of our security audit of Acre and Mezo Portal stBTC smart contracts.

Acre is a native Bitcoin staking platform enabling users to deposit BTC and earn yield in BTC. Each BTC deposited into Acre will mint stBTC, a liquid staked token that is 1:1 redeemable for BTC. Users earn yield from staked BTC which is compounded into liquid stBTC.

Acre's stBTC smart contract is intended to enable minting through the Mezo Portal. Users can mint stBTC tokens using WBTC and tBTC deposits as collateral in Mezo Portal. A variable interest rate is applied to the stBTC tokens minted through the Mezo Portal, with the interest generated from tBTC and WBTC deposits being directed to the Treasury.

We audited the new features on previously audited Mezo Portal and Acre smart contracts. The scope of this audit was limited to those features specifically:

- Mezo's external stBTC minting/burning;
- Mezo's support for liquidity treasury Multi-sig in the Portal smart contract;
- Acre's Non-Fungible withdrawals; and
- Acres's external stBTC shares minting.

We performed an assessment of the overall security posture of Acre and Mezo Portal smart contracts, including identifying strengths and areas for improvement in the system design and implementation, in addition to feedback on the project documentation, test suite, and the use of dependencies.

Threat Model

For this audit we considered a threat model that included malicious users and observers of the protocol as the main threat actors. We considered a threat model where a malicious user can take advantage of receipt minting by circumventing stBTC minting fees. Additionally, due to the complexity of fee calculations we considered attacks that can take advantage of incorrect fee calculations. Also, we considered unauthorized minting of receipts as a potential area of concern.

We considered the protocol's privileged roles out of scope for this audit, and to be sufficiently decentralized and not malicious.

Security by Design

We identified issues in some design elements of the smart contracts. We found that partial withdrawals can be used to circumvent receipt minting fees resulting in stBTC receipt minting fees being skipped ([Issue 1](#)). Additionally, the Portal smart contract can lose track of collected fees due to partial withdrawals ([Issue 2](#)). Moreover, due to emergency pause mechanism in the stBTC smart contract, users will not be able to repay receipts for the duration of the pause period ([Issue 3](#)).



Secure Implementation

We identified an issue in the implementation of the `stBTC` smart contract where incorrect function visibility has been used in the implementation of the `mintDebt` and `repayDebt` functions ([Issue 4](#)). Additionally, the `mintReceipt` and `repayReceipt` functions are not consistent with the rest of the functions in terms of zero `amount` checks in the `Portal` smart contract ([Issue 6](#)).

Use of Dependencies

Both Mezo Portal and Acre `stBTC` use OpenZeppelin libraries which are well audited and battle tested. We did not identify any issues in the use of dependencies.

Tests

We found that there are sufficient tests written for both the `Portal` and `stBTC` smart contracts. However, a more thorough test suite could have uncovered ([Issue 1](#)) and ([Issue 2](#)).

Project Documentation

We discovered that the project has comprehensive documentation, which provides ample information to understand the intended functioning of the protocol.



Key Findings Table

Issues	Severity	Status
ISSUE #1 User Can Mint Owed Fees as stBTC Resulting in Paying Zero Fees to the Portal	High	Fixed
ISSUE #2 Smart Contract Can Lose Track of <code>feeCollected</code> Due To Partial Withdrawals	Medium	Fixed
ISSUE #3 Users Unable to Repay Debt When the stBTC Smart Contract is Paused	Medium	Not Fixed
ISSUE #4 Incorrect Function Visibility Implemented in the <code>mintDebt</code> and <code>repayDebt</code> Functions	Low	Fixed
ISSUE #5 If <code>mintCap</code> is 100 Percent, a User Can Mint 100 Percent of the Deposit Without Returning to the Portal	None	Not Fixed
ISSUE #6 Implement Zero <code>amount</code> Checks	None	Fixed

Severity definitions can be found in [Appendix A](#)



Findings

We describe the security issues identified during the security audit, along with their potential impact. We also note areas for improvement and optimizations in accordance with best practices. This includes recommendations to mitigate or remediate the issues we identify, in addition to their status before and after the fix verification.

ISSUE#1

User Can Mint Owed Fees as stBTC Resulting in Paying Zero Fees to the Portal

 High

 Fixed

Location

[contracts/Portal.sol#L660-L667](#)

[contracts/Portal.sol#L500-L506](#)

[contracts/Portal.sol#L498](#)

[contracts/Portal.sol#L489-L491](#)

Description

The Mezo Portal allows partial withdrawal of deposits including deposits that have been used to mint stBTC receipts. If a deposit has been used to mint stBTC receipts, the accrued fees are only calculated upon full withdrawal requests and partial withdrawals do not incur stBTC minting fees. A user can only partially withdraw up to a maximum amount of the deposit balance minus owed fees. Hence, he cannot withdraw part of the owed fees. However, the user can mint an stBTC receipt again up to a maximum amount of accrued fees and not repay that amount resulting in paying zero fees causing the Portal to lose track of the fees for that particular deposit.

Impact

Users can skip stBTC receipt minting fees resulting in paying zero fees to the Mezo Portal.

Recommendation

We recommend including the `depositInfo.feeOwed` as part of mint limit check in the `mintReceipt` function as the code snippet below which requires moving the `_updateFee` function call before the check.

```
updateFee(depositInfo.token):
uint256 mintLimit = (depositInfo.balance * fee.mintCap) / 100;
if (amount + depositInfo.receiptMinted > mintLimit) {
    revert ReceiptMintLimitExceeded(
        mintLimit,
        depositInfo.receiptMinted,
        amount
    );
}
```

In addition to the above recommendation, we recommend disabling partial withdrawals for deposits which have been used to mint stBTC receipts.



ISSUE#2

Smart Contract Can Lose Track of `feeCollected` Due To Partial Withdrawals

Medium

Fixed

Location

[contracts/Portal.sol#L500-L506](#)

[contracts/Portal.sol#L498](#)

[contracts/Portal.sol#L489-L491](#)

[contracts/Portal.sol#L458-L460](#)

Description

The `withdraw` function in the `Portal` smart contract allows partials withdrawals of deposits. Deposits which have been used to mint `stBTC` receipts can be withdrawn up to a maximum amount of the deposit balance minus owed fees. `feeInfo[token].feeCollected` which is used to track `stBTC` receipts minting fees in the `Portal` smart contract is updated only upon full withdrawals. Therefore, a user can intentionally withdraw up to the maximum amount that can be partially withdrawn, without unlocking the deposit, causing the `Portal` smart contract to lose track of fees.

Additionally, in such a scenario which results in `fee == depositAmount`, the user has to withdraw zero `amount` to unlock the fees. However, the `withdraw` function doesn't allow this as a result of a zero `amount` check.

Impact

The `Portal` smart contract loses track of `feeCollected` incorrectly accounting for the token balance in the smart contract.

Recommendation

We recommend implementing the suggested fix in ([Issue 1](#)).



ISSUE#3

Users Unable to Repay Debt When the stBTC Smart Contract is Paused

Medium

Not Fixed

Location

[contracts/stBTC.sol#L347](#)

Description

When the smart contract is paused, the `repayDebt` function becomes inoperative. As a result, users are unable to repay their debts and must wait until the smart contract is unpaused. During this period, the contract continues to accumulate fees, even though the system is paused. This delay can lead to increased fees for users, as they are unable to take any action to mitigate their debt until the system is operational again.

Impact

Users might end up paying more fees than they anticipated.

Recommendation

We recommend allowing users to repay debt during the pause period, or implement a mechanism where users don't pay fees for the period of time that the smart contract is paused.

Verification Status

The Acre team stated that they will keep the pause feature which will be used in the event of a significant threat to user funds.

ISSUE#4

Incorrect Function Visibility Implemented in the `mintDebt` and `repayDebt` Functions

Low

Fixed

Location

[contracts/stBTC.sol#L304](#) [stBTC.sol#L349](#)

Description

The `mintDebt` and `repayDebt` functions have external visibility, which makes them inaccessible when called within the `mintReceipt` and `burnReceipt` functions, respectively. This will result in an `undeclared identifier` error during compilation.

Impact

The smart contract will fail to compile due to the use of an incorrect function visibility specifier.

Recommendation

Use `public` as function visibility in `mintDebt` and `repayDebt` functions.



ISSUE#5

If `mintCap` is 100 Percent, a User Can Mint 100 Percent of the Deposit Without Returning to the Portal

None

Not Fixed

Location

[contracts/Portal.sol#L78](#)

Description

The `Portal` smart contract allows a `mintCap` of 100 percent. In such a case, a user can mint 100 percent of the deposit and not return to the Portal.

Impact

The user has no incentive to return to Mezo Portal.

Recommendation

To encourage users to stay on the Mezo Portal and pay fees, we recommend limiting the `mintCap` to a value less than 100 percent.

Verification Status

The Mezo team stated that the `mintCap` value is set by the trusted smart contract owner, if this results in allowing users to mint 100% of their deposit then the smart contract implementation should not prevent it.

ISSUE#6

Implement Zero amount Checks

None

Fixed

Location

[contracts/Portal.sol#L641](#) [contracts/Portal.sol#L682](#)

Description

There are missing zero `amount` checks in the `mintReceipt` and `repayReceipt` functions.

Impact

None – no security impact.

Recommendation






We recommend implementing zero `amount` checks in the referenced functions.



Appendix A

Severity Rating Definitions

At Thesis Defense, we utilize the [Immunefi Vulnerability Severity Classification System - v2.3](#).

Severity	Definition
 Critical	<ul style="list-style-type: none"> • Manipulation of governance voting result deviating from voted outcome and resulting in a direct change from intended effect of original results • Direct theft of any user funds, whether at-rest or in-motion, other than unclaimed yield • Direct theft of any user NFTs, whether at-rest or in-motion, other than unclaimed royalties • Permanent freezing of funds • Permanent freezing of NFTs • Unauthorized minting of NFTs • Predictable or manipulable RNG that results in abuse of the principal or NFT • Unintended alteration of what the NFT represents (e.g. token URI, payload, artistic content) • Protocol insolvency
 High	<ul style="list-style-type: none"> • Theft of unclaimed yield • Theft of unclaimed royalties • Permanent freezing of unclaimed yield • Permanent freezing of unclaimed royalties • Temporary freezing of funds • Temporary freezing NFTs
 Medium	<ul style="list-style-type: none"> • Smart contract unable to operate due to lack of token funds • Enabling/disabling notifications • Griefing (e.g. no profit motive for an attacker, but damage to the users or the protocol) • Theft of gas • Unbounded gas consumption
 Low	<ul style="list-style-type: none"> • Contract fails to deliver promised returns, but doesn't lose value
 None	<ul style="list-style-type: none"> • We make note of issues of no severity that reflect best practice recommendations or opportunities for optimization, including, but not limited to, gas optimization, the divergence from standard coding practices, code readability issues, the incorrect use of dependencies, insufficient test coverage, or the absence of documentation or code comments.



Appendix B

Thesis Defense Disclaimer

Thesis Defense conducts its security audits and other services provided based on agreed-upon and specific scopes of work (SOWs) with our Customers. The analysis provided in our reports is based solely on the information available and the state of the systems at the time of review. While Thesis Defense strives to provide thorough and accurate analysis, our reports do not constitute a guarantee of the project's security and should not be interpreted as assurances of error-free or risk-free project operations. It is imperative to acknowledge that all technological evaluations are inherently subject to risks and uncertainties due to the emergent nature of cryptographic technologies.

Our reports are not intended to be utilized as financial, investment, legal, tax, or regulatory advice, nor should they be perceived as an endorsement of any particular technology or project. No third party should rely on these reports for the purpose of making investment decisions or consider them as a guarantee of project security.

Links to external websites and references to third-party information within our reports are provided solely for the user's convenience. Thesis Defense does not control, endorse, or assume responsibility for the content or privacy practices of any linked external sites. Users should exercise caution and independently verify any information obtained from third-party sources.

The contents of our reports, including methodologies, data analysis, and conclusions, are the proprietary intellectual property of Thesis Defense and are provided exclusively for the specified use of our Customers. Unauthorized disclosure, reproduction, or distribution of this material is strictly prohibited unless explicitly authorized by Thesis Defense. Thesis Defense does not assume any obligation to update the information contained within our reports post-publication, nor do we owe a duty to any third party by virtue of making these analyses available.

