STRONGHOLD
SECURITY

Fathom Stablecoin Security Audit Report

October 25, 2023

Contents

Executive Summary

Title	Description
Client	Fathom
Project	fathom-stablecoin
Platform	XDC Network
Language	Solidity
Repository	https://github.com/Into-the-Fathom/fathom-stablecoin-smart-contracts
Initial commit	a7d2ff13fde5beab1a677bfc7171641b4023d9c8
Final commit	2098cefb842d19fbb310dee23df16eb2da59bc32
Timeline	August 03 2023 - September 11 2023

Project Overview

FXD is a stablecoin designed to maintain a soft peg to the US Dollar. The value of FXD is designed to remain stable, with a slight fluctuation allowed around the target value.

FXD may be borrowed when the user gains the required borrowing power through the collateralization mechanism. The first source of borrowing power in Fathom Protocol is the deposited XDC (the native coin of the XDC network) as collateral for FXD.

Audit Scope

File	Link
BookKeeper.sol	BookKeeper.sol
CollateralTokenAdapter.sol	CollateralTokenAdapter.sol
PositionManager.sol	PositionManager.sol
FathomStablecoinProxyActions.sol	FathomStablecoinProxyActions.sol

Audit Methodology

General Code Assessment

The code is reviewed for clarity, consistency, style, and whether it follows code best practices applicable to the particular programming language used, such as indentation, naming convention, commented code blocks, code duplication, confusing names, irrelevant
or missing comments, etc. This part is aimed at understanding the overall code structure and protocol architecture. Also, it seeks to learn overall system architecture and business logic and how different parts of the code are related to each other.

Code Logic Analysis

The code logic of particular functions is analyzed for correctness and efficiency. The code is checked for what it is intended for, the algorithms are optimal and valid, and the correct data types are used. The external libraries are checked for relevance and correspond to the tasks they solve in the code. This part is needed to understand the data structures used and the purposes for which they are used. At this stage, various public checklists are applied in order to ensure that logical flaws are detected.

Entities and Dependencies Usage Analysis

The usages of various entities defined in the code are analyzed. This includes both: internal usage from other parts of the code as well as possible dependencies and integration usage. This part aims to understand and spot overall system architecture flaws and bugs in integrations with other protocols.

Access Control Analysis

Access control measures are analyzed for those entities that can be accessed from outside. This part focuses on understanding user roles and permissions, as well as which assets should be protected and how.

Use of checklists and auditor tools

Auditors can perform a more thorough check by using multiple public checklists to look at the code from different angles. Static analysis tools (Slither) help identify simple errors and highlight potentially hazardous areas. While using Echidna for fuzz testing will speed up the testing of many invariants, if necessary.

Vulnerabilities

The audit is directed at identifying possible vulnerabilities in the project's code. The result of the audit is a report with a list of detected vulnerabilities ranked by severity level:

Severity	Description
Critical	Vulnerabilities leading to the theft of assets, blocking access to funds, or any other loss of funds.
High	Vulnerabilities that cause the contract to fail and that can only be fixed b y modifying or completely replacing the contract code.
Medium	Vulnerabilities breaking the intended contract logic but without loss of fun ds and need for contract replacement.
Low	Minor bugs that can be taken into account in order to improve the overall qu ality of the code

After the stage of bug fixing by the Customer, the findings can be assigned t he following statuses:

Status	Description
Fixed	Recommended fixes have been made to the project code and no longer affect it s security.
Acknowledged	The Customer took into account the finding. However, the recommendations wer e not implemented since they did not affect the project's safety.

Findings Summary

Severity	# of Findings
Critical	2
High	1
Medium	10
Low	17

ID	Severity	Title	Status
C-1	Critical	Overpowered whitelisted users	Fixed
C-2	Critical	Access after the `give` position	Fixed
H-1	High	The modifier is missed for stablecoin deposits in BookKeeper	Fixed
M-1	Medium	Too many owner's rights in the protocol	Acknowledged
M-2	Medium	The `accrueStabilityFee` function does not have a range for `_debtAccumulatedRate`	Acknowledged
M-3	Medium	Non-guaranteed liquidation	Acknowledged
M-4	Medium	The `setTotalDebtCeiling` function does not have a range for `_totalDebtCeiling`.	Acknowledged
M-5	Medium	DOS by opening positions	Acknowledged
M-6	Medium	Null address checks	Fixed
M-7	Medium	An unchecked large `uint` cast to `int`	Fixed
M-8	Medium	Reentrancy in SafeToken library	Fixed
M-9	Medium	A position can't be exported to other position like described	Fixed
M-10	Medium	A possible incorrect collateral adapter in the vault deploy	Fixed
L-1	Low	A function description	Fixed

L-2	Low	Balance checks for fee token transfers	Acknowledged
L-3	Low	Null address checks	Fixed
L-4	Low	Typos	Fixed
L-5	Low	Gas optimizations	Fixed
L-6	Low	The `onlyDelegateCall` modifier is missed	Fixed
L-7	Low	An emitted event with null value	Fixed
L-8	Low	Sanity checks	Fixed
L-9	Low	Events in setters	Fixed
L-10	Low	Improve sanity checks	Fixed
L-11	Low	A division by zero	Fixed
L-12	Low	A unused modifier	Fixed
L-13	Low	Input value checks	Fixed
L-14	Low	An unexpected data type conversation	Fixed
L-15	Low	NatSpec for functions	Fixed
L-16	Low	A `SafeApprove` is deprecated.	Acknowledged
L-17	Low	More than 18 decimals precision for tokens	Acknowledged

Findings

Critical

C-1 Critical Overpowered whitelisted users Fixed

Description

The `allowManagePosition` function gives rights to `whitelisted` users to add another user as `whitelisted`. So, a `whitelisted` user can frontrun any calls to disallow a particular user and always have a `whitelisted` status.

PositionManager.sol#L92
PositionManager.sol#L55

Recommendation

We recommend adding a reliable way to remove whitelisted users.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

C-2 Critical Access after the `give` position Fixed

Description

Before transferring the ownership of a position (the `give` function in `PositionManager`) to another owner, the previous owner can add itself in the `ownerWhitelist` mapping using the `allowManagePosition` function, allowing the previous owner to access several functions after transferring the ownership.

Code snippet:
PositionManager.sol#L142

Recommendation

We recommend adding additional checks.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

High

H-1	High	The modifier is missed for stablecoin deposits in BookKeeper	Fixed

Description

In `FathomStablecoinProxyActions` `stablecoinAdapter` has a `depositRAD` method.

StablecoinAdapter.sol#L77

In plces where `RAY` is 10 ** 27, if we extract `_stablecoinAdapter` used in `FathomStablecoinProxyActions` and call the `depositRAD` function with `rad` as (10 ** 27 - 1) because of the integer division, `1` token will be burnt and `10 ** 27 - 1` will be deposited to the `BookKeeper`. Then it can be withdrawn.

Recommendation

We recommend adding a `onlyLiquidationStrategy` modifier for this function.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

Medium

M-1	Medium	Too many owner's rights in the protocol	Acknowledged

Description

The owner has too many rights in the protocol.

1. The owner can change `PriceOracle` and manipulate the price as he wants.
   PositionManager.sol#L325

2. The owner can change `AccessControlConfig` and remove governance from protocol.
   BookKeeper.sol#L252-L254

3. The owner can change the `BookKeeper`.
   PositionManager.sol#L330
   PriceOracle.sol#L82
   ShowStopper.sol#L70

Recommendation

We recommend adding the `OnlyGovernance` modifier and using it.

Client's commentary

> We plan to change ownership to be the Governance address.

M-2 Medium **The `accrueStabilityFee` function does not have a range for `_debtAccumulatedRate`** Acknowledged

Description

The `accrueStabilityFee` function can add an unlimited value of `_debtAccumulatedRate` to send to a `_stabilityFeeRecipient` an unlimited value of stablecoins.

BookKeeper.sol#L474

Recommendation

We recommend adding ranges for `_debtAccumulatedRate`.

Client's commentary

> The `accrueStabilityFee` function has an `onlyStabilityFeeCollector` modifier and the `accrueStabilityFee` function is called in the collect function inside the `stabilityFeeCollector` contract. The collect function calls the `accrueStabilityFee` function's `_debtAccumulateRate` value after calculation, so it's safe enough.

M-3 Medium Non-guaranteed liquidation Acknowledged

Description

When a position becomes uncollateralized, only users from `whiteList` call `LiquidationEngine.liquidate`. These users check the positions and determine how many should be liquidated.
Thus, whitelisted users can non-liquidate their positions.

LiquidationEngine.sol#L290

Recommendation

We recommend exploring the possibility of liquidating positions without `whitelist`.

Client's commentary

> Expected behavior. We plan to have a whitelist for liquidators until we are confident that XDC and the Fathom community are ready to open it to the general public. Also, we will need to solve some problems like race conditions or simply the situation when the one providing the higher gas price will win.

M-4 Medium The `setTotalDebtCeiling` function does not have a range for `_totalDebtCeiling`. Acknowledged

Description

The owner can stop the protocol by setting the value to 0.

BookKeeper.sol#L146

Recommendation

We recommend adding ranges for `_debtAccumulatedRate`.

Client's commentary

> We acknowledge this issue, and the owner role bearer must be careful not to set this value to 0 accidentally.

M-5 Medium DOS by opening positions Acknowledged

Description

The `open` function allows to open positions for another address. An attacker can flood protocol users by opening millions of positions because transactions on the Xinfin network are cheap.

PositionManager.sol#L110

Recommendation

We recommend opening a position for `msg.sender`.

Client's commentary

> If somebody opens many positions to the point that blockchain network halts, it's an issue, but opening millions of positions doesn't hurt the protocol.

M-6 Medium Null address checks Fixed

Description

Below is a list of functions with no null address check which can lead users to lose their funds.

`moveCollateral`:
PositionManager.sol#L212

`moveCollateral`:
PositionManager.sol#L230

`moveStablecoin`:
PositionManager.sol#L245

`exportPosition`:
PositionManager.sol#L253

`deposit`:
CollateralTokenAdapter.sol#L138

`emergencyWithdraw`:
CollateralTokenAdapter.sol#L161

Recommendation

We recommend adding null address checks.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

M-7 Medium An unchecked large `uint` cast to `int` Fixed

Description

In the `_withdraw` function CollateralTokenAdapter.sol#L138, the argument _amount is cast to `int256` with no overflow checks.

If the amount is `2**256 - 1337`, the result of `-int256(_amount)` will be `1337`, not `-(2**256 - 1337)` as intended. So, the collateral available for withdrawal is increased, not decreased, and the token will be minted. The additional collateral can be withdrawn with the following function call.

Recommendation

We recommend adding the`int256(_amount) > 0` check.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

M-8 Medium Reentrancy in SafeToken library Fixed

Description

SafeToken library implements operations that can be used for reentrancy.

SafeToken.sol

In FathomStablecoinProxyActions in methods wipeAndUnlockXDC and wipeAllAndUnlockXDC the `safeTransferETH` call is used with the `msg.sender` address before the price check.

FathomStablecoinProxyActions.sol#L114-L142

So, an attacker can implement the fallback function and execute their logic, for example the one that calls the protocol back before the price is updated but after their actions.

Recommendation

We recommend moving call transfers after all the side effects and price updates. Consider adding a gas limit to the `safeTransferETH` or other SafeToken calls and adding a reentrancy guard to functions like `wipeAndUnlockXDC`.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

M-9 Medium **A position can't be exported to other position like described** Fixed

Description

The `allowMigratePosition` function in PositionManager considers that we pass a user who can manage our positions to allow or disallow the positions access.

PositionManager.sol#L98-L105

The `onlyMigrationAllowed` modifier reads this `migrationWhitelist` mapping to check msg.sender was appoved by _migrantAddress.
PositionManager.sol#L60-L63

In the `exportPosition` function, this modifier is used with a `_destination` argument that states to be the address of PositionHandler, not the user address.

PositionManager.sol#L249-L266

If the PositionHandler address is provided here like described in NatSpec, export will revert because PositionHandler can't be msg.sender in the `allowMigratePosition` call.
The only way to call exportPosition is to provide the user address which called `allowMigratePosition` like in the test case.

PositionManager.test.js#L544

Recommendation

We recommend updating NatSpec.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

M-10 Medium **A possible incorrect collateral adapter in the vault deploy** Fixed

Description

The `collateralTokenAdapter` sets the vault that is used to store deposits.

CollateralTokenAdapter.sol#L127-L134

The vault receives `bytes32 _collateralPoolId, address _collateralToken, address _collateralAdapter` arguments to the constructor.

Vault.sol#L35-L43

The vault contract defines modifier `onlyAdapter` that checks that call made by the `collateralAdapter` address provided earlier in the constructor.

Vault.sol#L24

In the protocol's AccessControlConfig we have an `ADAPTER_ROLE` role that is used to mark an adapter and determine calls made by the adapter.
However, when we set the vault with the `_collateralAdapter` inside to the CollateralTokenAdapter using the `setVault` setter, nothing checks that the `_collateralAdapter` inside the vault set has the `ADAPTER_ROLE`. So, the protocol deployer can set a vault with a poisoned adapter and this adapter will have access to withdraw deposits.

Recommendation

We recommend checking in the `setVault` that the vault collateral adapter has the `ADAPTER_ROLE`.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

Low

L-1	Low	A function description	Fixed

Description

In the `BookKeeper`, the `settleSystemBadDebt` function does not have the `onlySystemDebtEngine` modifier but it is stated in the description that it does - `only be called by the SystemDebtEngine`.

BookKeeper.sol#L439

Recommendation

We recommend fixing NatSpec for this function.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-2 Low Balance checks for fee token transfers Acknowledged

Description

The `deposit` function in `CollateralTokenAdapter` receives tokens from a user with the `safeTransferFrom` function without the balance check. It can lead to assets lost due to fee-on-transfer tokens usage (USDT, USDC, etc.).

CollateralTokenAdapter.sol#L182

Recommendation

We recommend checking the balances before and after the transfer and calculating the amount.

Client's commentary

> We don't expect deflationary tokens to be used in the protocol in the foreseeable future and are aware of actions we must take to onboard such tokens.

L-3 Low Null address checks Fixed

Description

for `from` and `to`:

BookKeeper.sol#L454

for `toBeRemoved`:

CollateralTokenAdapter.sol#L108

for `_user`:

PositionManager.sol#L92

PositionManager.sol#L101

Recommendation

We recommend adding null address checks.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-4 Low Typos Fixed

Description

`withdrawl` to `withdrawal`:
CollateralTokenAdapter.sol#L168
`recevied` to `received`:
PositionManager.sol#L228
`neeeded` to `needed`:
FathomStablecoinProxyActions.sol#L455
`postion` to `position`:
BookKeeper.sol#L414
`systemDebyEngine` to `systemDebtEngine`:
BookKeeper.sol#L427
`stalbecoin` to `stablecoin`:
BookKeeper.sol#L38

Recommendation

We recommend fixing the typos.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-5 Low Gas optimizations Fixed

Description

In the `open` function the `lastPositionId` variable is located in storage, but it reads multiple times.

PositionManager.sol#L117

Recommendation

We recommend creating a memory variable for `lastPositionId`.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-6 Low The `onlyDelegateCall` modifier is missed Fixed

Description

The `allowManagePosition` function does not have a `onlyDelegateCall` modifier.

FathomStablecoinProxyActions.sol#L42

Recommendation

We recommend adding a `onlyDelegateCall` modifier.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-7 Low An emitted event with null value Fixed

Description

If an `amount` is 0, then the `_withdraw` function emits a `LogWithdraw(_amount)` event:

CollateralTokenAdapter.sol#L216

Recommendation

We recommend moving the event to a 'if' block.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-8 Low Sanity checks Fixed

Description

In the `setVault` function:

CollateralTokenAdapter.sol#L128.

Recommendation

We recommend adding sanity checks for input params with the `address` type.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-9 Low Events in setters Fixed

Description

Add events for old and new values:

- in the `setPriceOracle` function in the `PositionManager` contract for the `priceOracle` value: PositionManager.sol#L325
- in the `setBookKeeper` function in the `PositionManager` contract for the `BookKeeper` value: PositionManager.sol#L330

Recommendation

We recommend adding events.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-10 Low Improve sanity checks Fixed

Description

A `stableCoinReferencePrice` param in the oracle could be equal to zero. Thus, you can replace `>= 0` by `> 0` in the `initialize` and `setPriceOracle` functions:

PositionManager.sol#L326

PositionManager.sol#L84.

Recommendation

We recommend improving sanity checks.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-11 Low A division by zero Fixed

Description

A `_y` input value in the `divup`,`rdiv` and `wdiv` functions in the `CommonMath` contract could be equal to zero:

CommonMath.sol#L34

CommonMath.sol#L38

CommonMath.sol#L42

Recommendation

We recommend adding zero checks for the `_y` value.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-12 Low A unused modifier Fixed

Description

A `onlyCollateralManager` modifier is not used in the `CollateralTokenAdapter` contract: CollateralTokenAdapter.sol#L68.

Recommendation

We recommend removing the unused code.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-13 Low Input value checks Fixed

Description

A `require(_amount > 0)` check in the `moveCollateral` function in the `BookKeeper` contract: BookKeeper.sol#L242.

A `require(_value > 0)` check in the `moveStablecoin` function in the `BookKeeper` contract: BookKeeper.sol#L264.

A `require(_value > 0)` check in the `mintUnbackedStablecoin` function in the `BookKeeper` contract: BookKeeper.sol#L454.

A `require(_src != _dst)` check in the `moveCollateral` function in the `BookKeeper` contract: BookKeeper.sol#L242.

A `require(_src != _dst)` check in the `moveStablecoin` function in the `BookKeeper` contract: BookKeeper.sol#L264.

Recommendation

We recommend adding the checks if needed.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-14 Low An unexpected data type conversation Fixed

Description

A `collateralToken` variable has an `address` type, and this variable converts into an `address` type:
CollateralTokenAdapter.sol#L214.

Recommendation

We recommend removing the unexpected data type conversation.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-15 Low NatSpec for functions Fixed

Description

Some functions do not have NatSpec:

PositionManager.sol#L343

BookKeeper.sol#L146

BookKeeper.sol#L173

CollateralTokenAdapter.sol#L127.

Recommendation

We recommend adding NatSpec for the functions.

Client's commentary

> Fixed in commit: 2098cefb842d19fbb310dee23df16eb2da59bc32.

L-16 Low A `SafeApprove` is deprecated. Acknowledged

Description

The use of `safeApprove` is deprecated now in favor of `safeIncreaseAllowance` and `safeDecreaseAllowance`.

CollateralTokenAdapter.sol#L195
FathomStablecoinProxyActions.sol#L196
FathomStablecoinProxyActions.sol#L207
FathomStablecoinProxyActions.sol#L228

Recommendation

We recommend replacing `safeApprove` with `safeIncreaseAllowance` and `safeDecreaseAllowance`.

L-17 Low More than 18 decimals precision for tokens Acknowledged

Description

For collaterals that have more than 18 decimals precision, the `_convertTo18` function loses precision:

FathomStablecoinProxyActions.sol#L462.

The greater token decimals, the more precision lost.

Recommendation

We recommend considering limiting the maximum amount of collateral token decimals supported.

Client's commentary

> We don't expect collaterals with more than 18 decimals precision to be used in the protocol in the foreseeable future and are aware of the actions we must take to onboard such tokens.

Conclusion

During the audit process 2 CRITICAL, 1 HIGH, 10 MEDIUM and 17 LOW severity findings have been spotted.

Disclaimer

The Stronghold audit makes no statements or warranties about the utility of the code, the safety of the code, the suitability of the business model, investment advice, endorsement of the platform or its products, the regulatory regime for the business model, or any other statements about the fitness of the contracts to purpose, or their bug-free status. The audit documentation is for discussion purposes only.