PUBLIC

Code Assessment

of the Sulu Extensions XI Smart Contracts

July 20, 2023

Produced for

by

Contents

1 Executive Summary

Dear Mona and Sean,

Thank you for trusting us to help Avantgarde Finance with this security audit. Our executive summary provides an overview of subjects covered in our audit of the latest reviewed contracts of Sulu Extensions XI according to Scope to support you in forming an opinion on their security risks.

Avantgarde Finance implements external positions for staking with Kiln (upgrade of old external position) and an integration with ERC-4626 tokenized vaults. Additionally, some changes to the existing code base have been performed.

The most critical subjects covered in our audit are asset solvency, functional correctness, front-running, and accurate fund valuation. However, front-running protection and accurate fund valuation are improvable due to inaccuracies, see Pricing ERC4626 and Unclaimed Staking Rewards Are Not Valued. Similarly, delayed fund valuation may be problematic, see Slashing Can Be Avoided.

The general subjects covered are code complexity, upgradeability, unit testing, and documentation.

In summary, we find that the codebase provides a good but improvable level of security.

It is important to note that security audits are time-boxed and cannot uncover all vulnerabilities. They complement but don't replace other vital measures to secure a project.

The following sections will give an overview of the system, our methodology, the issues uncovered and how they have been addressed. We are happy to receive questions and feedback to improve our service.

Sincerely yours,

ChainSecurity

1.1 Overview of the Findings

Below we provide a brief numerical overview of the findings and how they have been addressed.

Critical-Severity Findings	0
High-Severity Findings	0
Medium-Severity Findings	1
• Risk Accepted	1
Low-Severity Findings	2
• Risk Accepted	2

2 Assessment Overview

In this section, we briefly describe the overall structure and scope of the engagement, including the code commit which is referenced throughout this report.

2.1 Scope

The assessment was performed on the source code files inside the Sulu Extensions XI repository based on the documentation files. The table below indicates the code versions relevant to this report and when they were received.

V	Date	Commit Hash	Note
1	10 July 2023	86d45e025f520f2e1ebf7810df1fe8b978601d01	Initial Version

For the newly added solidity smart contracts and the upgrade Kiln external position, the compiler version `0.8.19` was chosen. For all other changes, the same compiler version as in the original audits was chosen.

In scope is the refactoring of the contracts (formatting, widening the compiler pragma for interfaces and related or similar changes).

Additionally, the following util files were added for compiler version `0.8.19`:

```
contracts/release/extensions/integration-manager/integrations/utils/0.8.19/AdapterBase.sol
contracts/utils/0.8.19/AddressArrayLib.sol
contracts/utils/0.8.19/AssetHelpers.sol
contracts/utils/0.8.19/Uint256ArrayLib.sol
contracts/persistent/address-list-registry/address-list-owners/utils/0.8.19/AddOnlyAddressListOwnerBase.sol
contracts/persistent/address-list-registry/address-list-owners/utils/0.8.19/AddOnlyAddressListOwnerConsumerMixin.sol
contracts/persistent/address-list-registry/address-list-owners/utils/IAddOnlyAddressListOwner.sol
```

The gated shares wrapper scope is:

```
contracts/persistent/shares-wrappers/gated-redemption-queue/GatedRedemptionQueueSharesWrapperLib.sol
contracts/persistent/shares-wrappers/gated-redemption-queue/bases/GatedRedemptionQueueSharesWrapperLibBase1.sol
```

For the Balancer v2 price feeds, the following file was in scope:

```
contracts/release/infrastructure/price-feeds/derivatives/feeds/BalancerV2StablePoolPriceFeed.sol
```

The Convex and Aura staking wrapper scope is:

```
contracts/external-interfaces/IAuraStashToken.sol
contracts/external-interfaces/IConvexStashTokenWrapper.sol
contracts/release/infrastructure/staking-wrappers/aura-balancer-v2-lp/AuraBalancerV2LpStakingWrapperLib.sol
contracts/release/infrastructure/staking-wrappers/convex-curve-lp/ConvexCurveLpStakingWrapperLib.sol
contracts/release/infrastructure/staking-wrappers/StakingWrapperBase.sol
```

Further, the following contracts are in scope for the Kiln external position:

```
contracts/release/extensions/external-position-manager/external-positions/kiln-staking/KilnStakingPositionDataDecoder.sol
contracts/release/extensions/external-position-manager/external-positions/kiln-staking/KilnStakingPositionLib.sol
contracts/persistent/external-positions/kiln-staking/KilnStakingPositionLibBase1.sol
contracts/release/extensions/external-position-manager/external-positions/kiln-staking/KilnStakingPositionParser.sol
contracts/external-interfaces/IKilnStakingContract.sol
contracts/persistent/external-positions/kiln-staking/KilnStakingPositionLibBase2.sol
```

The scope for the ERC-4626 integration is:

```
contracts/release/extensions/integration-manager/integrations/adapters/ERC4626Adapter.sol
contracts/release/infrastructure/price-feeds/derivatives/feeds/ERC4626PriceFeed.sol
```

2.1.1 Excluded from scope

Everything not mentioned above is out-of-scope (e.g. changes to Notional positions).

Further, the external systems are out-of-scope and are assumed to work correctly. We assume that users and managers are made aware of the (under given circumstances) breaking behaviour described in Notes.

2.2 System Overview

This system overview describes the initially received version (Version 1) of the contracts as defined in the Assessment Overview.

Furthermore, in the findings section, we have added a version icon to each of the findings to increase the readability of the report.

Avantgarde Finance updates to the whole codebase in the form of formatting (`forge fmt`). Additionally, Avantgarde Finance ports several utils to Solidity 0.8.19. Changes to the `GatedRedemptionQueueSharesWrapper`, `BalancerV2StablePoolPriceFeed` and the Convex/Aura shares wrapper are implemented. Further, withdrawal functionality has been added to the Kiln staking external position. Additionally, a generic integration adapter has been implemented for standard ERC-4626 tokens.

2.2.1 Changes

- `GatedRedemptionQueueSharesWrapper`: An event was adapted to emit more data.
- `BalancerV2StablePoolPriceFeed`: Read-only reentrancy applies to any Balancer V2 pool. Before, the price feed checked for reentrancy only for certain pools. Now, it always performs the read-only reentrancy checks.
- Both Convex and Aura set reward tokens now as "stash tokens". The wrappers have been adapted to handle both old (without stash token) and new types of pools. Further changes include adding `try/catch` to certain operations such as harvesting rewards and the ability to remove extra rewards. See, note Convex and Aura removeToken Clarification for more details and considerations of these changes.

2.2.2 Kiln Withdrawal functionality

The Kiln staking external position has been updated to support withdrawals.

Three new actions were added:

- `Unstake`: Requests an exit with `requestValidatorsExit` on the Kiln staking contract. It is expected that the exit will eventually be executed and that the consensus layer fee recipient will eventually receive ETH.
- `PausePositionValue`: Pause `getManagedAssets()` so that it reverts. See below for more details.
- `UnpausePositionValue`: Undo the previous action.

`getManagedAssets()` has been adapted to include the pausing functionality that makes calls to the function revert. Recall that the function returns the number of staking actions times 32 ETH as the managed assets. To handle exits (through `Unstake` or slashing), the fee-claiming mechanism has been adapted. Namely, the consensus layer fee withdrawal implements logic to guess whether a validator has been exited by setting a threshold for the consensus layer fee recipient with which it is decided whether a

position was exited (typically, the balance of an exited position will be significantly higher than the balance of a non-exited position). However, note that there are some important considerations, see Kiln Fees on Slashing, Kiln Bad Accounting, and Slashing Can Be Avoided.

2.2.3 ERC-4626 integration

Avantgarde Finance implements an adapter for acquiring standard ERC-4626 tokens which implements the functions:

- `lend()`: Calls `deposit()` on the ERC-4626 vault.
- `redeem()`: Calls `redeem()` on the ERC-4626 vault.

Note that the vault is expected to work as described in the EIP. Namely, all assets are deposited and all shares are exactly withdrawn. Further, it is expected that it follows the EIP so that if the approval is given on the vault token, the approved address can burn for the approver.

Also, the integration is only suited for standard vaults that do not have complex logic on top. The usage of the adapter must be carefully evaluated for each protocol.

Further, a price feed for such tokens is implemented that uses `convertToAssets()`.

Please see ERC4626 considerations for additional considerations.

2.2.4 Roles and Trust Model

Please refer to the main audit report and the extension audit reports for a general trust model of Sulu.

In general, we assume Enzyme only interacts with normal ERC-20 tokens that do not have multiple entry points, callbacks, fees-on-transfer, or other special behaviours.

Fund owners and asset managers are generally fully trusted for a fund. However, their powers can be limited through the fund's settings. The funds' settings/policies are assumed to be set up correctly for the intended configuration/usage.

The managers are expected to regularly claim the fees and to pause the position if under-/overvaluation of the fund occurs.

Governance is fully trusted and expected to not only behave honestly but also to fully understand the systems they are interacting which includes choosing appropriate parameters.

All external systems are expected to be non-malicious and work correctly as documented. Kiln is a closed-source project. However, we received their code but documentation was lacking.

Please see Notes for additional expectations.

3 Limitations and use of report

Security assessments cannot uncover all existing vulnerabilities; even an assessment in which no vulnerabilities are found is not a guarantee of a secure system. However, code assessments enable the discovery of vulnerabilities that were overlooked during development and areas where additional security measures are necessary. In most cases, applications are either fully protected against a certain type of attack, or they are completely unprotected against it. Some of the issues may affect the entire application, while some lack protection only in certain areas. This is why we carry out a source code assessment aimed at determining all locations that need to be fixed. Within the customer-determined time frame, ChainSecurity has performed an assessment in order to discover as many vulnerabilities as possible.

The focus of our assessment was limited to the code parts defined in the engagement letter. We assessed whether the project follows the provided specifications. These assessments are based on the provided threat model and trust assumptions. We draw attention to the fact that due to inherent limitations in any software development process and software product, an inherent risk exists that even major failures or malfunctions can remain undetected. Further uncertainties exist in any software product or application used during the development, which itself cannot be free from any error or failures. These preconditions can have an impact on the system's code and/or functions and/or operation. We did not assess the underlying third-party infrastructure which adds further inherent risks as we rely on the correct execution of the included third-party technology stack itself. Report readers should also take into account that over the life cycle of any software, changes to the product itself or to the environment in which it is operated can have an impact leading to operational behaviors other than those initially determined in the business specification.

4 Terminology

For the purpose of this assessment, we adopt the following terminology. To classify the severity of our findings, we determine the likelihood and impact (according to the CVSS risk rating methodology).

- *Likelihood* represents the likelihood of a finding to be triggered or exploited in practice
- *Impact* specifies the technical and business-related consequences of a finding
- *Severity* is derived based on the likelihood and the impact

We categorize the findings into four distinct categories, depending on their severity. These severities are derived from the likelihood and the impact using the following table, following a standard risk assessment procedure.

Likelihood	Impact		
	High	Medium	Low
High	Critical	High	Medium
Medium	High	Medium	Low
Low	Medium	Low	Low

As seen in the table above, findings that have both a high likelihood and a high impact are classified as critical. Intuitively, such findings are likely to be triggered and cause significant disruption. Overall, the severity correlates with the associated risk. However, every finding's risk should always be closely checked, regardless of severity.

5 Findings

In this section, we describe our findings. The findings are split into these different categories:

- Design: Architectural shortcomings and design inefficiencies
- Correctness: Mismatches between specification and implementation

Below we provide a numerical overview of the identified findings, split up by their severity.

Critical-Severity Findings	0
High-Severity Findings	0
Medium-Severity Findings	1

- Slashing Can Be Avoided Risk Accepted

Low-Severity Findings	2

- Pricing ERC4626 Risk Accepted
- Unclaimed Staking Rewards Are Not Valued Risk Accepted

5.1 Slashing Can Be Avoided

Design Medium Version 1 Risk Accepted

CS-SUL11-001

For Kiln, the smart contract layer does not immediately know when a slashing event on the Consensus Layer has happened. Offchain, however, it is easy to immediately know when a validator has been slashed.

A user of an enzyme vault that uses one of the staking external positions could monitor for slashings and immediately withdraw from the vault when such an event happens. By doing this, they will be able to redeem their assets (up to the available liquidity) at the pre-slashing price.

Once the slashing is accounted for in the vault (after about 36+ days for Kiln), the slashing loss of the users that withdrew previously will instead be taken by those users that are still deposited in the vault.

Risk accepted:

Avantgarde Finance acknowledged the issue and replied:

```
Since consensus layer slashing is not readable directly from the execution layer,
slashing can only be made known by posting to the execution layer, and there will
always be an opportunity to front-run posting (the same goes for Chainlink aggregators).

Fund managers must be aware of this risk and take any necessary precautions to mitigate
the risk where needed, e.g., via policies and/or queued redemptions.
```

5.2 Pricing ERC4626

Correctness | Low | Version 1 | Risk Accepted

CS-SUL11-002

The pricing of ERC-4626 tokens could be off. Namely, `convertToAssets()` does not include:

- fees
- variations among callers (no price per user but rather an average among users)
- slippage or other on-chain conditions

Ultimately, the fund could be under-/overvalued.

Risk accepted:

Avantgarde Finance replied:

```
The Enzyme Council will need to review each new class of erc4626 asset to determine
whether convertToAssets() is potentially significantly deviant from actual redemption
```

5.3 Unclaimed Staking Rewards Are Not Valued

Design | Low | Version 1 | Risk Accepted

CS-SUL11-005

In KilnStakingPositionLib, getManagedAssets simply returns `validatorCount * 32ETH`.

However, there may be a significant amount of staking rewards that are owed to the vault, if they haven't been claimed recently. Rewards are only accounted for in the vault's valuation once they are claimed, which only the vault manager can do using the claimFeesAction. This action is very gas-intensive, so it will likely not be called often.

The actual value of the external position will be under-represented, which may lead to share price arbitrage.

Risk accepted:

Avantgarde Finance replied:

```
Since accrued rewards live in distinct execution layer and consensus layer contracts
per validator, it is too gas-intensive to query balances across all validators on
every position value query.

Fund managers must be aware of this risk and take any necessary precautions to mitigate
the risk where needed, e.g., claiming rewards with reasonable frequency, policies, and/or
queued deposits and redemptions.
```

6 Notes

We leverage this section to highlight further findings that are not necessarily issues. The mentioned topics serve to clarify or support the report, but do not require an immediate modification inside the project. Instead, they should raise awareness in order to improve the overall understanding.

6.1 Convex and Aura removeToken Clarification

Note Version 1

The Convex and Aura staking wrappers implement the function `removeExtraRewardToken()` to remove certain extra tokens. The function intends to remove extra reward tokens that led to errors in the previous implementation so that they can be readded later on in a correct way using the logic for the stash tokens. Its intention is not to remove extra reward tokens forever (since all reward tokens will be added automatically even if they had been removed).

The wrappers wrap the addition of extra reward tokens in a `try/catch` block. When adding multiple tokens, if the addition of any of the extra reward tokens would fail, none of the reward tokens would be added. This is the intended behaviour that enzyme expects.

6.2 ERC4626 Considerations

Note Version 1

No extra steps needed to deposit or redeem (besides ERC-20 approval) should be present. Otherwise, the adapter will not work as intended. Any additional logic on top of what is specified in the ERC should be carefully evaluated.

6.3 ERC4626 minIncomingShares Must Be Used

Note Version 1

ERC4626 vaults can be affected by rounding issues when depositing.

This is mitigated in the ERC4626Adapter by letting managers specify a `minIncomingSharesAmount`.

It is important that managers specify a reasonable amount, to limit the impact of rounding errors.

6.4 Kiln Bad Accounting

Note Version 1

In Kiln, the admin of the staking contract can withdraw from the CL fee recipient contract. If that occurs after an exit, and the vault manager performs an action that sweeps ETH from the external position (e.g. claim fees, sweep ETH), then the accounting will be off. Namely, the exited validator will be still accounted for (32 ETH) while the exited ETH will be in the vault proxy. Ultimately, double-counting assets could be possible. Hence, the share price will be too high.

Further, donations could be made to the CL fee recipient contract so that the threshold is reached. Ultimately, when claiming CL fees, the validator count would be reduced even though there was no validator exit. The value of the fund could be reduced.

Slashed validators will not be removed in the case of a mass slashing event, where the slashed amount is larger than 32 ETH - EXITED_VALIDATOR_THRESHOLD. The value of the fund would be overestimated in this case.

Note that the threshold should be very carefully chosen so that the system works properly all of the time while not allowing arbitrage opportunities. Fund managers and asset managers are expected to actively monitor for bad scenarios so that they can pause the external position's valuation. Once paused, they are expected to reconcile with Avantgarde Finance so that a contract upgrade can be released that fixes the issue.

Further, note that Avantgarde Finance is aware of this but decided to implement more complex logic for these rather unlikely corner cases only if necessary.

6.5 Kiln Fees on Slashing

Note Version 1

Note the following documented behaviour in Kiln:

> In case of slashing, the exit is not requested we don't exempt anything [from fees]. This is in case of slashing, the staker will be rebated manually. A slashed validator may have accumulated enough skimmed rewards to still have a balance > 32 ETH. All of this will be taken into account and the staker will be compensated for the commission taken. on its principal and the loss according to the SLA described in the Terms&Conditions.

Kiln takes a fee on the total leftover ETH balance if a slashing event leaves a position under 31 ETH. As this amount must be manually returned by Kiln, it will not be considered in the valuation of the enzyme vault until that process is completed. This may lead to undervaluing the vault. If Kiln returns the owed amount to the external position rather than the vault directly, the amount will only be counted once the vault manager calls the sweepETH action.