PUBLIC

Code Assessment

of the Sulu Extensions XXIII Smart Contracts

April 04, 2025

Produced for

by

Contents

1 Executive Summary

Dear all,

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

Enzyme Foundation implements an external position for writing call and put options in Myso V3.

The most critical subjects covered in our audit are functional correctness and integration with Myso V3. The general subjects covered are specification, trustworthiness, error handling, and documentation.

In summary, we find that the codebase provides a high level of security. Furthermore, important consideration are outlined in the sections Notes and in Trust Model.

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
• Code Corrected	1
Low-Severity Findings	0

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 XXIII 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	13 Mar 2025	d03ce6747306dd79f2c3518b443e33b2483e189d	Initial Version
2	01 Apr 2025	9b61e90e4d0a6732c71f0c57d715d64297dcf557	After intermediate report

For the solidity smart contracts, the compiler version `0.8.19` was chosen.

The files in scope are:

```
contracts/release/extensions/external-position-manager/external-positions/myso-v3/
    IMysoV3OptionWritingPosition.sol
    MysoV3OptionWritingPositionLib.sol
    MysoV3OptionWritingPositionParser.sol
    bases/
        MysoV3OptionWritingPositionLibBase1.sol
```

2.1.1 Excluded from scope

All other files not listed above are out of scope. Myso v3 is out of scope and is expected to work as documented.

2.2 System Overview

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

Enzyme Foundation implements an external position that integrates with Myso V3, enabling funds to write call and put options. For a detailed description of how external positions function, please refer to the main code assessment report of Sulu.

Myso Option Writing Position. Myso V3 gives users access to on-chain structured products. Users can sell call and put option that, typically, are bought by trading firms. The lifecycle of options in Myso V3 can be outlined as below:

1. The option to be sold is created (initialized). The seller locks the underlying tokens in the escrow. While the option is unsold, the seller can withdraw the underlying (and any other) token from the escrow.

2. The option is bought by a buyer and a premium (either in the underlying or the settlement token) is paid directly to the seller.
3. Eventually, the option becomes exercisable. While the option is exercisable, the buyer can borrow the underlying token (and repay) as well as execute the option. Note that fees (either in the underlying or the settlement token) are paid directly to the seller.
4. Finally, the option expires. The seller can withdraw any token from the escrow. Typically, the withdrawable tokens will be the underlying or the settlement token.

The external position (EP) enables Enzyme vaults to sell options. The EP implements the following actions:

1. `CreateEscrowByTakingQuote`: Creates a new escrow and sells an option. The action leverages the `Router.takeQuote` function that facilitates OTC/RFQ-based option trading. The position is tracked accordingly.
2. `CreateEscrowByStartingAuction`: Creates a new escrow and initiates a dutch auction with `Router.createAuction`. Bidders can then buy the option according to the price defined by the auction. The position is tracked accordingly.
3. `Sweep`: For a set of tokens, sends the balances held by the EP (received as outlined above) to the vault.
4. `WithdrawTokensFromEscrows`: Withdraws arbitrary tokens from a set of escrows and sends them to the vault (only once the option is expired) with `Router.withdraw`.
5. `CloseAndSweepEscrows`: Withdraws the underlying and the settlement tokens from the escrow and sweeps the balances held by the EP. Finally, the position is untracked. Note that the withdrawal might be skipped to allow for closing a fully exercised position prior to expiry so that `getManagedAssets` can be unblocked earlier (see below).

If any position is open within the EP, `getManagedAssets` reverts and otherwise returns empty arrays accordingly. Also, Note that no debt is created. Thus, `getDebtAssets` returns empty arrays.

2.3 Trust Model

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

For funds using the Myso Option Writing EP, the asset managers are fully trusted. In the worst case, an asset manager could sell malicious options that could effectively lead to draining a vault. Additionally, an asset manager could lock funds indefinitely by selling options far in the future.

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 Open Findings

In this section, we describe any open findings. Findings that have been resolved have been moved to the Resolved Findings section. The findings are split into these different categories:

- 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	0
Low-Severity Findings	0

6 Resolved Findings

Here, we list findings that have been resolved during the course of the engagement. Their categories are explained in the Open Findings section.

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

- Mispricing Due to Incorrect Received Assets Code Corrected

Low-Severity Findings	0

6.1 Mispricing Due to Incorrect Received Assets

Correctness Medium Version 1 Code Corrected

CS-SUL-23-002

If all escrows are closed, the share price could be severely mispriced due to EP's parser incorrectly returning the received assets.

More specifically, for the action `CloseAndSweepEscrows` the `assetsToReceive_` is expected to be the set of the underlying and settlement tokens for the provided escrow indices. However, only the underlying and settlement token for the first provided escrow index will be part of the set since the loop in `__decodeCloseAndSweepEscrows` will effectively only add the underlying and settlement tokens of the first provided escrow index.

Consider the following example:

1. The denomination asset is ETH.
2. Assume MLN is bought to enable selling options with MLN as the underlying. MLN is tracked.
3. Assume option `1` is sold where the settlement token is ETH and option `2` is sold where the settlement token is USDC. Additionally, assume that for both the premium token is MLN.
4. Eventually, both options are exercised.
5. The escrows are closed for options `[1, 2]`. `assetsToReceive_` will be `[MLN, ETH]` instead of `[MLN, ETH, USDC]` which an asset manager would expect.
6. The fund is mispriced due to USDC not being tracked.

To summarize, an incorrect aggregation of received assets could lead mispriced shares which could allow attackers to perform profitable share price arbitrage.

Code corrected:

The code has been adjusted to ensure that the correct escrow indices is accessed in the iterations.

7 Informational

We utilize this section to point out informational findings that are less severe than issues. These informational issues allow us to point out more theoretical findings. Their explanation hopefully improves the overall understanding of the project's security. Furthermore, we point out findings which are unrelated to security.

7.1 Gas Optimizations

Informational **Version 1** **Acknowledged**

CS-SUL-23-001

Several potential gas optimizations exist. Below is a non-comprehensive list of examples:

1. `__closeAndSweepEscrow`: The function could be optimized by providing `escrowIdxs` as indices of `openEscrowsIdxs` in *descending* order. That could effectively allow for removing the search in `__closeAndSweep` (related to `escrowIdxFound`) reducing the number of storage operations from roughly `escrowIdxs*openEscrowsIdxs` to roughly `openEscrowsIdxs`.
2. `__withdraw`: The function could be optimized by calling `Escrow.handleWithdraw` directly.
3. `__closeAndSweep`: The function validates that the owner is the EP. However, that is implied by `escrowIdxFound`.

Further, other optimizations could be possible. However, they offer the benefit of clearer error messages.

Acknowledged:

Enzyme Foundation is aware of and acknowledges the optimizations.

8 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.

8.1 Arbitrary Skipping of Withdrawals

Note Version 1

Action `CloseAndSweepEscrows` can re-enable the fund valuation. Note that a flag `_skipWithdrawFromEscrow` can be set for each escrow to be closed. If the flag is set to `true`, none of the underlying and the settlement token will not be withdrawn. That allows for closing fully exercised options by preventing a potential donation attack vector.

However, it further allows for skipping withdrawals for the following scenarios:

- option not sold yet
- option has expired

Note that in these scenarios it is typically unreasonable to skip the withdrawal.

Note that Enzyme Foundation specified that this design is intended to provide flexibility. For the EP, the asset managers are fully trusted and thus no violation Trust Model as worse actions could be performed.

8.2 Redeeming in Kind

Note Version 1

Asset Managers should be aware that while the EP is being used no deposits and redemptions against specific assets are possible. However, in-kind redemptions might be possible.

Users should be aware that redeeming in kind will only return a pro rata share of the tracked assets and will not consider the assets by external positions. Hence, an amount lower than expected could be received.