macro

Maple Finance A-3

Security Audit

December 20, 2024
Version 2.0.0

Presented by 0xMacro

Table of Contents

Introduction

This document includes the results of the security audit for Maple Finance's smart contract code as found in the section titled ‘Source Code’. The security audit was performed by the Macro security team from December 9, 2024 to December 19, 2024.

The purpose of this audit is to review the source code of certain Maple Finance Solidity contracts, and provide feedback on the design, architecture, and quality of the source code with an emphasis on validating the correctness and security of the software in its entirety.

Disclaimer: While Macro’s review is comprehensive and has surfaced some changes that should be made to the source code, this audit should not solely be relied upon for security, as no single audit is guaranteed to catch all possible bugs.

Overall Assessment

The following is an aggregation of issues found by the Macro Audit team:

Severity	Count	Acknowledged	Won't Do	Addressed
Medium	1	-	-	1
Low	1	-	-	1

Maple Finance was quick to respond to these issues.

Specification

Our understanding of the specification was based on the following sources:

- Discussions with the Maple Finance team.
- Available public documentation and provided docs for the specific release.

Source Code

The following source code was reviewed during the audit:

Initial:

- maple-strategies @ 24d23131dedfcd7cb63edd5dc908003c10e7d4a7
- globals-v2 @ 52137cfcdaa1a3f3687e679e2cf7f211ff6fb3aa
- pool-v2 @ 5c3104948b4cbe7a6b0558c57d5bb38eaf0ea077

Final:

- maple-strategies @ 4b065a531f743ee89de445fbbe1efc77d71fd8b9

Specifically, we audited the following contracts within **maple-strategies** repository:

Source Code	SHA256
contracts/MapleAaveStrategy.sol	b0a7ce5ef10f9ae1af92fca917d0a4d2e9d81aef635c4b5f135d02d53d3a6229
contracts/MapleAbstractStrategy.sol	ce75c1450fbd157b72713cf405a904be783cd000e6cba89656b6eb4a8292f521
contracts/MapleBasicStrategy.sol	bab694872e24d3a5fddd593163364542088257047f6ba3b296cbf0ac87dc24fd
contracts/MapleSkyStrategy.sol	63416c228c6c3cf835478696d0c3d63a2865d12b9c933475dfa5b1444c2138fe
contracts/proxy/MapleStrategyFactory.sol	0e5d781845362613df25ac03df8919a6f4733d51b1900d0ea291af54014fde3c
contracts/proxy/aaveStrategy/MapleAaveStrategyInitializer.sol	8dccb99c32da26b76c0473136199ba4f6ee6e0db5d499d4f94b5a01a1037eadd
contracts/proxy/aaveStrategy/MapleAaveStrategyStorage.sol	d087e468068be6b6dbd2696fd56084f75d8610db927da50d2b893864b80358b8

Source Code	SHA256
contracts/proxy/basicStrategy/MapleBasicStrategyInitializer.sol	b2cac4482c5bbe3198f80f69c81daa44cf5e507ec0a3afec91f8bb79740b8bd7
contracts/proxy/basicStrategy/MapleBasicStrategyStorage.sol	2d14fe60ed939b65e766683007171fb1d8abaab3c465594cf2c0fc8ad85bf188
contracts/proxy/skyStrategy/MapleSkyStrategyInitializer.sol	8b93b0f6d98024504e94983412137e549c7c40af468b2ec29caa2d99b72eabea
contracts/proxy/skyStrategy/MapleSkyStrategyStorage.sol	f546374fba7701023d9ae3bd183347fbbbbfc4d74b44beafb912ff127afcb478

We also audited the following contract within **globals-v2** repository:

Source Code	SHA256
contracts/MapleGlobals.sol	24cae5f79f12d16145b2548553e0151e83a07658863d10aa0ccb880e7df3f131

We audited the following contracts within **pool-v2** repository:

Source Code	SHA256
contracts/MaplePool.sol	3bcf608a588e7948d6f46796c1c319527a3f579046512fb11cb11072721ed317
contracts/MaplePoolDelegateCover.sol	05f3733f72a257776e2777b471db1f10091a1040bdf81762332e52b4a93493e3
contracts/MaplePoolDeployer.sol	db9425bff1bca924e5074ec29fcc437766a057042edc778be3bc67b95ca0acd6
contracts/MaplePoolManager.sol	05ebf403d053b194c22581dd8517ec9b86a8e1d070fa505ba490ed4b710cab71
contracts/proxy/MaplePoolManagerInitializer.sol	590ea81e143f239477f067f22fd6e345b9ae1859f83c2b04ee7cb92a85229957

Source Code	SHA256
contracts/proxy/MaplePoolManagerStorage.sol	1840a4ba2d473c1bdf63a855cd37327e836cdd3c6534797e63317a9bb98c5289

Note: This document contains an audit solely of the Solidity contracts listed above. Specifically, the audit pertains only to the contracts themselves, and does not pertain to any other programs or scripts, including deployment scripts.

Issue Descriptions and Recommendations

Click on an issue to jump to it, or scroll down to see them all.

Security Level Reference

We quantify issues in three parts:

1. The high/medium/low/spec-breaking **impact** of the issue:
    - How bad things can get (for a vulnerability)
    - The significance of an improvement (for a code quality issue)
    - The amount of gas saved (for a gas optimization)
2. The high/medium/low **likelihood** of the issue:
    - How likely is the issue to occur (for a vulnerability)
3. The overall critical/high/medium/low **severity** of the issue.

This third part – the severity level – is a summary of how much consideration the client should give to fixing the issue. We assign severity according to the table of guidelines below:

Severity	Description
(C-x) Critical	We recommend the client **must** fix the issue, no matter what, because not fixing would mean **significant funds/assets WILL be lost.**
(H-x) High	We recommend the client **must** address the issue, no matter what, because not fixing would be very bad, *or* some funds/assets will be lost, *or* the code's behavior is against the provided spec.
(M-x) Medium	We recommend the client to **seriously consider** fixing the issue, as the implications of not fixing the issue are severe enough to impact the project significantly, albiet not in an existential manner.
(L-x) Low	The risk is small, unlikely, or may not relevant to the project in a meaningful way. Whether or not the project wants to develop a fix is up to the goals and needs of the project.
(Q-x) Code Quality	The issue identified does not pose any obvious risk, but fixing could improve overall code quality, on-chain composability, developer ergonomics, or even certain aspects of protocol design.
(I-x) Informational	Warnings and things to keep in mind when operating the protocol. No immediate action required.
(G-x) Gas Optimizations	The presented optimization suggestion would save an amount of gas significant enough, in our opinion, to be worth the development cost of implementing it.

Issue Details

~~M-1~~ Missing slippage protection in withdrawFromStrategy()

TOPIC	STATUS	IMPACT	LIKELIHOOD
Input Validation	Fixed	Medium	Medium

In the `MapleBasicStrategy` contract, `withdrawFromStrategy()` is responsible for obtaining requested `assetsOut` in exchange for burning an appropriate amount of `shares` through interaction with the IERC4626Like `strategyVault` . When `strategyVault` has low liquidity, slippage may significantly affect this swap operation. This has been adequately addressed in `fundStrategy()` operation with `minSharesOut_` argument and corresponding validation.

However, `withdrawFromStrategy()` does not feature any slippage protection mechanism and, therefore, is susceptible to potentially burning much more than the expected amount of shares to obtain the requested assets by `assetsOut` .

Remediations to Consider

Add slippage protection to withdrawFromStrategy by:

- Adding a `maxSharesBurned_` parameter
- Adding a check to ensure the actual shares burned don't exceed the maximum specified
- Maintaining consistency with the protection model used in `fundStrategy()`

~~L-1~~ Missing validation in setPsm()

TOPIC	STATUS	IMPACT	LIKELIHOOD
Input Validation	Fixed	Low	Low

In the `MapleSkyStrategy` contract, the `setPsm()` enables protocol admins to change the Peg Stability Module (PSM) in use. However, in contrast to the initialization logic in `MapleSkyStrategyInitializer` , the feature for changing the PSM module address does not contain validation logic that the new PSM module uses adequate assets.

```
// validation present in MapleSkyStrategyInitializer._initialize()
require(IPSMLike(psm_).gem() == fundsAsset_, "MSSI:I:INVALID_GEM_PSM");
require(IPSMLike(psm_).usds() == usds_, "MSSI:I:INVALID_USDS_PSM");
```

If the instance of MapleSkyStrategy is misconfigured with an incorrect PSM module, it may lead to operations that revert or that perform incorrect calculations (e.g., due to different `to18ConversionFactor`)

Remediations to Consider

- Add corresponding missing validations to `MapleSkyStrategy.setPsm()`

Disclaimer

Macro makes no warranties, either express, implied, statutory, or otherwise, with respect to the services or deliverables provided in this report, and Macro specifically disclaims all implied warranties of merchantability, fitness for a particular purpose, noninfringement and those arising from a course of dealing, usage or trade with respect thereto, and all such warranties are hereby excluded to the fullest extent permitted by law.

Macro will not be liable for any lost profits, business, contracts, revenue, goodwill, production, anticipated savings, loss of data, or costs of procurement of substitute goods or services or for any claim or demand by any other party. In no event will Macro be liable for consequential, incidental, special, indirect, or exemplary damages arising out of this agreement or any work statement, however caused and (to the fullest extent permitted by law) under any theory of liability (including negligence), even if Macro has been advised of the possibility of such damages.

The scope of this report and review is limited to a review of only the code presented by the Maple Finance team and only the source code Macro notes as being within the scope of Macro's review within this report. This report does not include an audit of the deployment scripts used to deploy the Solidity contracts in the repository corresponding to this audit. Specifically, for the avoidance of doubt, this report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project. In this report you may through hypertext or other computer links, gain access to websites operated by persons other than Macro. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such websites' owners. You agree that Macro is not responsible for the content or operation of such websites, and that Macro shall have no liability to your or any other person or entity for the use of third party websites. Macro assumes no responsibility for the use of third party software and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.