OPTIMUM

BLOCKCHAIN SECURITY

Yearn Finance

Smart contract Security Assessment

Notional Finance LP Strategy

June, 2022

Table Of Contents

Disclaimer

This report does not provide any security warranty, investment advice, endorsement, or disapproval of any particular project or team. This report does not provide a warranty that the code in scope is completely free of vulnerabilities, bugs, or potential exploits. This report does not assess the financial risk of any asset. No third party should rely on this report to make any decisions to buy or sell any asset or product.

Delivering secured code is a continuous challenge that requires multiple steps. It is strongly recommended to use best code practices, write a full test suite, conduct an internal audit, and launch a bug bounty program as a complement to this report.

It is the sole responsibility of the project team to ensure that the code in scope is functioning as intended and that the recommendations presented in this report are carefully tested before deployment.

Overview Page

Summary

Project name	Yearn Finance
URL	https://yearn.finance/
Code	https://github.com/16slim/notional_lp_strategy
Commit hash	516ca1c962f2f9571dede382113ceb7e8975256a
Mitigations commit hash	
Language	Solidity

Contracts Assessed

Contract name	SHA-1
/contracts/Strategy.sol	d8a5d85d33c93fe8495319f8bdbc42287a5cf07e
/contracts/NotionalLpLib.sol	cdb3bcc629e980da305c15dff0d44361c204f344

Findings Summary

Severity	Found	Resolved	Partially resolved	Acknowledged
High	0	0	0	0
Medium	0	0	0	0
Low	2	0	0	0
Informational	0	0	0	0
Total	2	0	0	0

Classification of issues

Severity	
High	Vulnerabilities that may directly result in loss of funds, and thus require an urgent fix.
Medium	Issues that may not be directly exploitable, or with a limited impact, are still required to be fixed.
Low	Subjective issues with a negligible impact.
Informational	Subjective issues or observations with negligible or no impact.

Findings

Issue #01	Unsafe castings
Severity	Low
Location	*NotionalLpLib.sol* - Line 82 *NotionalLpLib.sol* - Line 123
Description	A few unsafe casting operations with different levels of likelihood were found in the codebase. It is hard to assess the exact impact of these. In general, it may lead to wrong code execution which will lead to unintended results.
Recommendation	Consider using SafeCast, or alternatively, add a check that validates that the value being casted is inside the bounds of the type. In case gas efficiency is crucial, consider adding a code comment proving that a harmful casting operation is impossible.
Resolution	

Issue #02	**Potential division by zero that may lead to a temporary denial of service**
Severity	Low
Location	*Strategy.sol* - Line 600
Description	*liquidatePosition* will revert for the case where *totalDebt = wantBalance* due to a division by zero. This may only lead to a temporary denial of service since it may be solved within the next call to *adjustPosition*.
Recommendation	The case of *totalDebt = wantBalance* implies that *unrealisedLosses = 0*, which means *lossesToBeRealised* can be just set to 0 in this case, and line 600 can be skipped (only for the case where *totalDebt = wantBalance*).
Resolution	

Trust Assumptions

	Notional finance smart contracts
Description	*Notional finance* contracts are trusted in many ways, including but not only: ● contracts in use are upgradeable.

Observations

1. Addresses that are used for external calls are injected in the initialization phase, which violates Yearn's production policy, specifying that addresses should be hardcoded instead.

Recommendations

Recommendation #01	**Safe math usage consistency**
Description	The safe-math library is used in *NotionalLpLib* multiple times. However, there are two unsafe operations (Lines 108, 214) which might not be exploitable, yet should utilize the safe-math add operation for the sake of code consistency.

Recommendation #02	***_getNTokenTotalValueFromPortfolio* - Outdated natspec**
Description	The natspec documentation for this function describes a return value, where in practice no value is returned. Consider updating the natspec comments for this function.

OPTIMUM

BLOCKCHAIN SECURITY