OPTIMUM
BLOCKCHAIN SECURITY

Yearn Finance

Smart contract
Security Assessment

Uni v3 Frax Strategy

May, 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/dudesahn/fraxUniswapStables/tree/usdc
Commit hash	7976d6a23493585419fc612d1fc2cfa6d6981dfc
Mitigations commit hash	
Language	Solidity

Contracts Assessed

Contract name	SHA-1
/contracts/StrategyFraxUniswapUSDC.sol	58bff3ae07612e39bb7611b1f0d2b2442faca204

Findings Summary

Severity	Found	Resolved	Partially resolved	Acknowledged
High	0	0	0	0
Medium	1	0	0	0
Low	0	0	0	0
Informational	0	0	0	0
Total	1	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	***StrategyFraxUniswapUSDC.mintNFT* - Possible denial of service**
Severity	**Medium**
Location	*StrategyFraxUniswapUSDC.sol - mintNFT* (L617-621)
Description	*mintNFT* acts as the final stage of the strategy initialization, and as such, it is supposed to be called only once by the vault managers. To support that, the function reverts in case the contract holds any Uniswap v3 nft. It leaves the possibility for a front-runner to front-run a call to *mintNFT* with a transaction that mints some Uniswap v3 nft with tiny liquidity and then transfers the ownership of it to the strategy contract. This way, the attacker causes any future call to *mintNFT* to revert.
Recommendation	Consider removing the Uniswap v3 nft balance check and consider reverting for the case where *nftId != 1* instead.
Resolution	

Trust Assumptions

	Frax smart contracts
Description	The *Frax* contracts are trusted in many ways, including but not only: • Funds deposited to *CommunalFarm* (aka *FraxLock*) are withdrawable only if withdrawals are not paused and only after the lock-up period is over.

Observations

1. *liquidatePosition* might revert in case *checkFraxPeg* reverts for too much USDC or too much FRAX (lines 448-462). Hence, an emergency withdraw may revert as well. In this case *slippageMax* will have to be adjusted before the emergency withdrawal.


OPTIMUM
BLOCKCHAIN SECURITY