CERTIK

StakeWith.Us

Vault Refactor

Security Assessment

May 13th, 2021

Audited By:
Alex Papageorgiou @ CertiK
alex.papageorgiou@certik.org
Reviewed By:
Camden Smallwood @ CertiK
camden.smallwood@certik.org

Disclaimer

CertiK reports are not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security review.

CertiK Reports do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

CertiK Reports should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

CertiK Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

What is a CertiK report?

- A document describing in detail an in depth analysis of a particular piece(s) of source code provided to CertiK by a Client.
- An organized collection of testing results, analysis and inferences made about the structure, implementation and overall best practices of a particular piece of source code.
- Representation that a Client of CertiK has completed a round of auditing with the intention to increase the quality of the company/product's IT infrastructure and or source code.

Overview

Project Summary

Project Name	StakeWith.Us - Vault Refactor
Description	Round four audit of the StakeWith.Us vault implementation codebase, with a new V3 approach to the vaults and strategies
Platform	Ethereum; Solidity, Yul
Codebase	GitHub Repository
Commits	1. 5cb4d37df18febe619f158b72c728e7e8a00c0ec 2. 057d59ab8a2507c4dc5f144dd3f9b93a6fa417d8

Audit Summary

Delivery Date	May 13th, 2021
Method of Audit	Static Analysis, Manual Review
Consultants Engaged	1
Timeline	May 3rd, 2021 - May 13th, 2021

Vulnerability Summary

Total Issues	3
Total Critical	0
Total Major	0
Total Medium	0
Total Minor	2
Total Informational	1

Executive Summary

We were tasked with performing a round three audit of the StakeWith.Us `uvault` codebase which revised the contract implementations to a V3 version migrating the logic from the generic handlers to the strategies themselves.

Over the course of the audit, we identified several issues related to the upcoming EIP-3074 in the security mechanisms implemented by the contracts that would be invalidated, such as utilizing `tx.origin == msg.sender` or a block-number mechanism based on `tx.origin`. These mechanisms were the core of some of the contracts in the scope of the audit and as they would need to be rigorously revised, they were ultimately omitted from the audit scope and they are not included in the report.

Additionally, a new strategy implementation was created that is meant to split its assets to multiple strategies. We identified a total of four issues on this contract with three being identified as `Minor`, however, the contract was also ultimately decided to be taken out of scope to be iterated over until a more mature implementation is presented for an audit.

The new curve-based strategy implementations were inspected against issues that arise from common misconceptions such as decimal differences, improper integration with the liquidity gauge mechanisms, and slippage risks. No issues were identified to this end.

A novel strategy was also introduced in the codebase that uses a leveraged Compound strategy whereby assets are borrowed and re-supplied into the protocol to achieve a desired leverage ratio based on a target supply. This operation can also be reversed via the respective deleverage functions and is loosely based on the PickleSwap leveraged compound implementation.

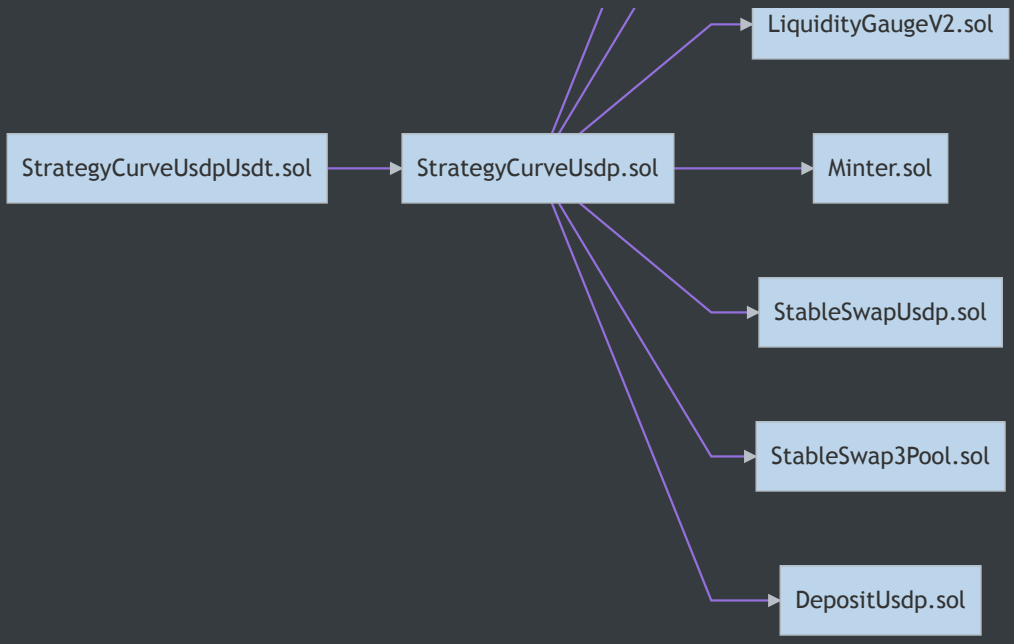
We were unable to identify any potential attack vectors in the strategy implementation or the mathematical formulas depicted therein.

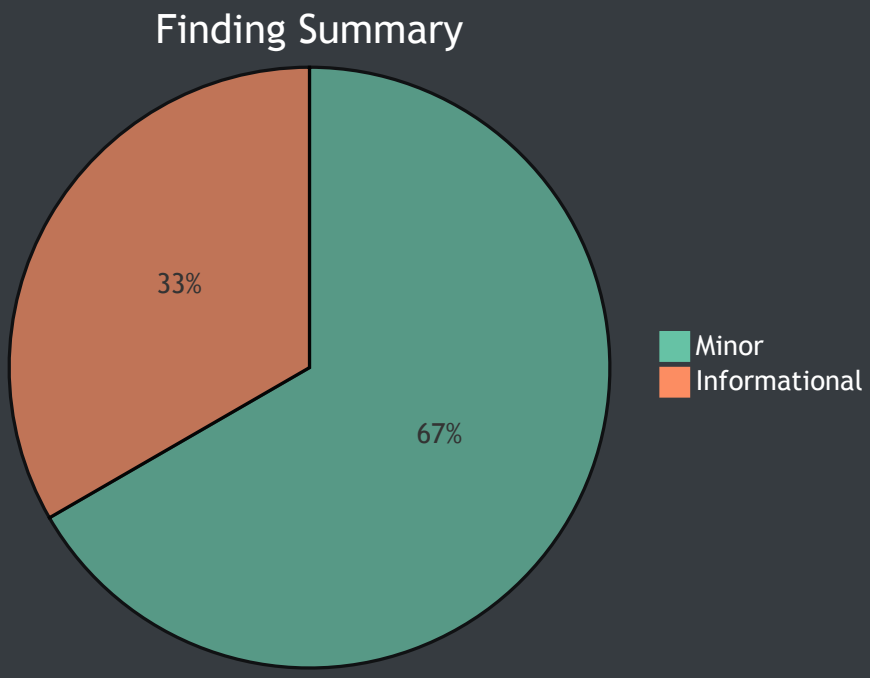
The StakeWith.Us team also shared with us a simulation for the leverage mathematical accuracy upon our request and showcased that the formulas have been rigorously evaluated to perform as intended.

Files In Scope

ID	Contract	Location
SER	StrategyERC20_V3.sol	contracts/StrategyERC20_V3.sol
SET	StrategyETH_V3.sol	contracts/StrategyETH_V3.sol
SCL	StrategyCompLev.sol	contracts/strategies/StrategyCompLev.sol
SCD	StrategyCompLevDai.sol	contracts/strategies/StrategyCompLevDai.sol
SLE	StrategyCompLevEth.sol	contracts/strategies/StrategyCompLevEth.sol
SLU	StrategyCompLevUsdc.sol	contracts/strategies/StrategyCompLevUsdc.sol
SCW	StrategyCompLevWbtc.sol	contracts/strategies/StrategyCompLevWbtc.sol
SCE	StrategyCurveEurs.sol	contracts/strategies/StrategyCurveEurs.sol
SEE	StrategyCurveEursEurs.sol	contracts/strategies/StrategyCurveEursEurs.sol
SCI	StrategyCurveIb.sol	contracts/strategies/StrategyCurveIb.sol
SID	StrategyCurveIbDai.sol	contracts/strategies/StrategyCurveIbDai.sol
SIU	StrategyCurveIbUsdc.sol	contracts/strategies/StrategyCurveIbUsdc.sol
CIU	StrategyCurveIbUsdt.sol	contracts/strategies/StrategyCurveIbUsdt.sol
CON	StrategyCurveUsdp.sol	contracts/strategies/StrategyCurveUsdp.sol
SUD	StrategyCurveUsdpDai.sol	contracts/strategies/StrategyCurveUsdpDai.sol
CON	StrategyCurveUsdpUsdc.sol	contracts/strategies/StrategyCurveUsdpUsdc.sol
CON	StrategyCurveUsdpUsdp.sol	contracts/strategies/StrategyCurveUsdpUsdp.sol
CON	StrategyCurveUsdpUsdt.sol	contracts/strategies/StrategyCurveUsdpUsdt.sol
SCU	StrategyCurveUst.sol	contracts/strategies/StrategyCurveUst.sol
CUD	StrategyCurveUstDai..sol	contracts/strategies/StrategyCurveUstDai..sol
SUU	StrategyCurveUstUsdc.sol	contracts/strategies/StrategyCurveUstUsdc.sol
CUU	StrategyCurveUstUsdt.sol	contracts/strategies/StrategyCurveUstUsdt.sol

File Dependency Graph


LiquidityGaugeV2.sol
StrategyCurveUsdpUsdt.sol
StrategyCurveUsdp.sol
Minter.sol
StableSwapUsdp.sol
StableSwap3Pool.sol
DepositUsdp.sol


Finding Summary
33%
67%
Minor
Informational

Manual Review Findings

ID	Title	Type	Severity	Resolved
SER-01	Pull-Over-Push Pattern	Logical Issue	Minor	⟳
SET-01	Pull-Over-Push Pattern	Logical Issue	Minor	⟳
SCL-01	Potential for Approval Lock	Logical Issue	Informational	✓

SER-01: Pull-Over-Push Pattern

Type	Severity	Location
Logical Issue	Minor	StrategyERC20_V3.sol L77-L80

Description:

The `setAdmin` function overwrites the previously set `admin` without ensuring that the `_admin` is able to conduct transactions on the blockchain.

Recommendation:

We advise the pull-over-push pattern to be applied here whereby a new administrator is first proposed and consequently needs to accept ownership thereby ensuring that they are able to transact and are aware of the particular contract's ownership.

Alleviation:

The StakeWith.Us - Vault Refactor development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

SET-01: Pull-Over-Push Pattern

Type	Severity	Location
Logical Issue	Minor	StrategyETH_V3.sol L76-L79

Description:

The `setAdmin` function overwrites the previously set `admin` without ensuring that the `_admin` is able to conduct transactions on the blockchain.

Recommendation:

We advise the pull-over-push pattern to be applied here whereby a new administrator is first proposed and consequently needs to accept ownership thereby ensuring that they are able to transact and are aware of the particular contract's ownership.

Alleviation:

The StakeWith.Us - Vault Refactor development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

SCL-01: Potential for Approval Lock

Type	Severity	Location
Logical Issue	Informational	StrategyCompLev.sol L80

Description:

The approval of the cToken address is set only once during the contract's `constructor`.

Recommendation:

Although the scenario under which the approval is depleted is practically impossible, we still recommend a new function to be introduced that allows resetting the approval to ensure that the contract can perpetually operate as the `leverage` and `deleverage` functions consume a substantial amount of approvals on each iteration.

Alleviation:

A new `approve` function was introduced that enables the administrator to reset the approval of the `cToken` by a specified amount. This function can also act as a safety mechanism whereby the approval is set to `0` thereby freezing the strategy.

Appendix

Finding Categories

Logical Issue

Logical Issue findings are exhibits that detail a fault in the logic of the linked code, such as an incorrect notion on how `block.timestamp` works.