CERTIK

Security Assessment

Unagii Vault V2 & Unagii Zap

Aug 7th, 2021

Table of Contents

Summary

This report has been prepared for StakeWithUs to discover issues and vulnerabilities in the source code of the Unagii Vault V2 & Unagii Zap project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases given they are currently missing in the repository;
- Provide more comments per each function for readability, especially contracts are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Majority of the findings are of informational nature with 7 minor findings. The minor findings comprise lack of validation for function parameters, ineffectual removal of token approval from dex protocols, volatile conditional statement when leveraging in `StrategyCompLev` and lack of validation for the sufficiency of Ether balance when forwarding them in TimeLock contract. The team responded to all of the findings by either remediating or declining the finding.

Overview

Project Summary

Project Name	Unagii Vault V2 & Unagii Zap
Description	The report represents audit of Strategy contracts that allow users to deposit funds that are then deposited in yield farming protocols of Compound and Protocol and the profits earned on strategies are sent to their respective `fundManager` contracts.
Platform	Ethereum
Language	Solidity, Vyper
Codebase	• unagii vault v2 • unagii zap
Commit	• pre-audit vault commit hash • post-audit vault commit hash • pre-audit zap commit hash • post-audit zap commit hash

Audit Summary

Delivery Date	Aug 07, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	Staking, Lending

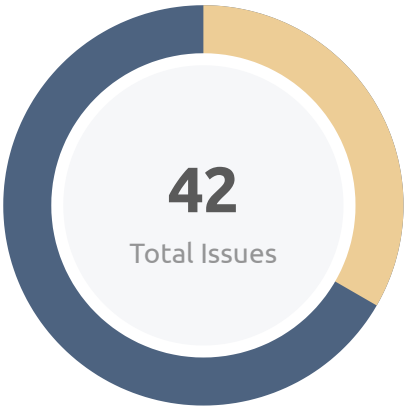
Vulnerability Summary

Vulnerability Level	Total	Pending	Partially Resolved	Resolved	Acknowledged	Declined
Critical	0	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	0	0	0	0	0	0
Minor	14	0	0	13	0	1
Informational	28	0	0	16	5	7
Discussion	0	0	0	0	0	0

Audit Scope

ID		File	SHA256 Checksum

Findings



Critical	0 (0.00%)
Major	0 (0.00%)
Medium	0 (0.00%)
Minor	14 (33.33%)
Informational	28 (66.67%)
Discussion	0 (0.00%)

ID	Title	Category	Severity	Status
EFM-01	Missing `event` Emission	Inconsistency	Informational	Resolved
EFM-02	Potential Re-Entrancy	Volatile Code	Minor	Resolved
EVS-01	Potential Over-centralization of Functionality	**Centralization / Privilege**	**Minor**	**Resolved**
FMS-01	Potential Re-Entrancy	Volatile Code	Minor	Resolved
SCA-01	Admin can change dex address	**Centralization / Privilege**	**Informational**	**Acknowledged**
SCA-02	Rewards are not claimed and transferred in migration of strategy	Volatile Code	Informational	Resolved
SCA-03	Usage of literal for arrays' lengths	Coding Style	Informational	Resolved
SCA-04	Inefficient storage read	Gas Optimization	Informational	Declined
SCA-05	Explicitly returning local variable	Gas Optimization	Informational	Resolved
SCB-01	Admin can change dex address	**Centralization / Privilege**	**Informational**	**Acknowledged**
SCB-02	Rewards are not claimed and transferred in migration of strategy	Volatile Code	Informational	Resolved
SCB-03	Usage of literal for arrays' lengths	Coding Style	Informational	Resolved
SCB-04	Inefficient storage read	Gas Optimization	Informational	Declined

ID	Title	Category	Severity	Status
SCB-05	Explicitly returning local variable	Gas Optimization	Informational	Resolved
SCL-01	Token approval is removed from wrong address	Logical Issue	Minor	Resolved
SCL-02	Admin can change dex address	**Centralization / Privilege**	**Informational**	**Acknowledged**
SCL-03	Incorrect conditional	Logical Issue	Minor	Resolved
SCL-04	Rewards are not claimed and transferred in migration of strategy	Volatile Code	Informational	Resolved
SCL-05	Explicitly returning local variable	Gas Optimization	Informational	Resolved
SCS-01	Admin can change dex address	**Centralization / Privilege**	**Informational**	**Acknowledged**
SCS-02	Rewards are not claimed and transferred in migration of strategy	Volatile Code	Informational	Resolved
SCS-03	Usage of literal for arrays' lengths	Coding Style	Informational	Resolved
SCS-04	Inefficient storage read	Gas Optimization	Informational	Declined
SCS-05	Explicitly returning local variable	Gas Optimization	Informational	Resolved
SES-01	Events are not emitted for state variables assignments	Volatile Code	Informational	Declined
SES-02	Lack of validation for function parameter	Logical Issue	Minor	Resolved
STR-01	Admin can change dex address	**Centralization / Privilege**	**Informational**	**Acknowledged**
STR-02	Rewards are not claimed and transferred in migration of strategy	Volatile Code	Informational	Resolved
STR-03	Usage of literal for arrays' lengths	Coding Style	Informational	Resolved
STR-04	Inefficient storage read	Gas Optimization	Informational	Declined
STR-05	Explicitly returning local variable	Gas Optimization	Informational	Resolved
STT-01	Events are not emitted for state variables assignments	Volatile Code	Informational	Declined

ID	Title	Category	Severity	Status
STT-02	Lack of validation for function parameter	Logical Issue	Minor	Resolved
STT-03	Inefficient storage read	Gas Optimization	Informational	Declined
TLS-01	Data location can be changed from `memory` to `calldata`	Gas Optimization	Informational	Resolved
TLS-02	Ether amount is not validated	Volatile Code	Minor	Resolved
TLS-03	Ether amount is not validated	Volatile Code	Minor	Resolved
TLS-04	Contract accepts arbitrary `ether`	Volatile Code	Minor	Declined
UTS-01	Possibility of Replay Attack in `Permit`	Logical Issue	Minor	Resolved
UTS-02	Susceptible to Signature Malleability	Volatile Code	Minor	Resolved
UTS-03	Missing `nextTimeLock` Clearance	Volatile Code	Minor	Resolved
VAU-01	Potential Over-centralization of Functionality	**Centralization / Privilege**	**Minor**	**Resolved**

EFM-01 | Missing `event` Emission

Category	Severity	Location	Status
Inconsistency	Informational	EthFundManager.vy (d1af693b837774c11c26ba930efc2c16f9a3346b): 221	Resolved

Description

The `EthFundManager` contract does not emit the already declared `ReceiveEth` event when receiving Ether.

Recommendation

We advise to un-comment the `ReceiveEth` event emission.

Alleviation

The development team opted to consider our references and utilized the `ReceiveEth` event.

EFM-02 | Potential Re-Entrancy

Category	Severity	Location	Status
Volatile Code	Minor	EthFundManager.vy (d1af693b837774c11c26ba930efc2c16f9a3346b): 908	Resolved

Description

The linked code segment updates the state of the contract after an external call.

Recommendation

We advise to execute the external call at the end of the function, hence following the Checks-Effects-Interactions pattern.

Alleviation

The development team opted to consider our references and added the re-entrancy lock decorator.

EVS-01 | Potential Over-centralization of Functionality

Category	Severity	Location	Status
Centralization / Privilege	Minor	EthVault.vy (d1af693b837774c11c26ba930efc2c16f9a3346b): 830	Resolved

Description

The linked function is meant to be used in an edge-case situation whereby the admin or time-lock can receive the excess token sent to the contract.

Recommendation

We advise this functionality to be guarded by either a time delay to ensure that the normal course of operation of the contract has progressed.

Alleviation

The development team opted to consider our references and restricted the access to the linked functions only to the time-lock address.

FMS-01 | Potential Re-Entrancy

Category	Severity	Location	Status
Volatile Code	Minor	FundManager.vy (d1af693b837774c11c26ba930efc2c16f9a3346b): 937	Resolved

Description

The linked code segment updates the state of the contract after an external call.

Recommendation

We advise to execute the external call at the end of the function, hence following the Checks-Effects-Interactions pattern.

Alleviation

The development team opted to consider our references and added the re-entrancy lock decorator.

SCA-01 | Admin can change dex address

Category	Severity	Location	Status
Centralization / Privilege	Informational	strategies/StrategyConvexAlUsd.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 109	Acknowledged

Description

The contract's admin has the privilege to change dex's address for each reward token.

Recommendation

No recommendations.

Alleviation

The team revisited the codebase and safe-guarded the functionality that changes dex address to be only callable through TimeLock contract. The TimeLock contract is handled by the Admin and hence the functionality to change dex address is not fully decentralized.

SCA-02 | Rewards are not claimed and transferred in migration of strategy

Category	Severity	Location	Status
Volatile Code	Informational	strategies/StrategyConvexAlUsd.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 414	Resolved

Description

The function on the aforementioned line migrates strategy to a new address by transferring its token balance to the new strategy address. The transferred funds does not involve the possible rewards accrued by strategy

Recommendation

We advise to revisit the `migrate` function and claim rewards before transferring the funds to new strategy address.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`. The team added boolean `claimRewardsOnMigrate`. Rewards is claimed on migrate when `claimRewardsOnMigrate` is `true`. If `false`, we will call `claimRewards` before migration. If there are significant amount of rewards to be claimed after migration, we can call `claimRewards` again, re-activate the strategy and call report.

SCA-03 | Usage of literal for arrays' lengths

Category	Severity	Location	Status
Coding Style	Informational	strategies/StrategyConvexAlUsd.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 21, 28, 331, 432	Resolved

Description

The aforementioned lines declare fixed length arrays and utilize integer literals to specify their lengths.

Recommendation

We advise to introduce a constant variable and utilize it to specify the lengths of fixed length arrays. This will increase the legibility of codebase.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

SCA-04 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	Informational	strategies/StrategyConvexAlUsd.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 99~100	Declined

Description

The aforementioned lines read storage variable `dex[_i]` inefficiently which can optimized by storing it in a local variable and then utilizing it.

Recommendation

We advise to make use of local variables to store storage values where they are used multiple times for reducing gas costs.

Alleviation

The team did not consider the recommendation stating that the gas savings are insignificant.

SCA-05 | Explicitly returning local variable

Category	Severity	Location	Status
Gas Optimization	Informational	strategies/StrategyConvexAlUsd.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 128, 266	Resolved

Description

The aforementioned lines explicitly return local variables which increases overall cost of gas.

Recommendation

Since named return variables can be declared in the signature of a function, consider refactoring to remove the local variable declaration and explicit return statement in order to reduce the overall cost of gas.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

SCB-01 | Admin can change dex address

Category	Severity	Location	Status
Centralization / Privilege	Informational	strategies/StrategyConvexBbtc.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 105	Acknowledged

Description

The contract's admin has the privilege to change dex's address for each reward token.

Recommendation

No recommendations.

Alleviation

The team revisited the codebase and safe-guarded the functionality that changes dex address to be only callable through TimeLock contract. The TimeLock contract is handled by the Admin and hence the functionality to change dex address is not fully decentralized.

SCB-02 | Rewards are not claimed and transferred in migration of strategy

Category	Severity	Location	Status
Volatile Code	Informational	strategies/StrategyConvexBbtc.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 398	Resolved

Description

The function on the aforementioned line migrates strategy to a new address by transferring its token balance to the new strategy address. The transferred funds does not involve the possible rewards accrued by strategy

Recommendation

We advise to revisit the `migrate` function and claim rewards before transferring the funds to new strategy address.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`. The team added boolean `claimRewardsOnMigrate`. Rewards is claimed on migrate when `claimRewardsOnMigrate` is `true`. If `false`, we will call `claimRewards` before migration. If there are significant amount of rewards to be claimed after migration, we can call `claimRewards` again, re-activate the strategy and call report.

SCB-03 | Usage of literal for arrays' lengths

Category	Severity	Location	Status
Coding Style	Informational	strategies/StrategyConvexBbtc.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 21, 27, 314, 416	Resolved

Description

The aforementioned lines declare fixed length arrays and utilize integer literals to specify their lengths.

Recommendation

We advise to introduce a constant variable and utilize it to specify the lengths of fixed length arrays. This will increase the legibility of codebase.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

SCB-04 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	Informational	strategies/StrategyConvexBbtc.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 95~96	Declined

Description

The aforementioned lines read storage variable `dex[_i]` inefficiently which can optimized by storing it in a local variable and then utilizing it.

Recommendation

We advise to make use of local variables to store storage values where they are used multiple times for reducing gas costs.

Alleviation

The team did not consider the recommendation stating that the gas savings are insignificant.

SCB-05 | Explicitly returning local variable

Category	Severity	Location	Status
Gas Optimization	Informational	strategies/StrategyConvexBbtc.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 124, 249	Resolved

Description

The aforementioned lines explicitly return local variables which increases overall cost of gas.

Recommendation

Since named return variables can be declared in the signature of a function, consider refactoring to remove the local variable declaration and explicit return statement in order to reduce the overall cost of gas.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

SCL-01 | Token approval is removed from wrong address

Category	Severity	Location	Status
Logical Issue	Minor	strategies/StrategyCompLev.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 76	Resolved

Description

The aforementioned line intends to remove token approval from previous dex address yet it erroneously removes token approval from the newly assigned dex address.

Recommendation

We advise to revisit the code and correctly provide the previous dex's address for the removal of token approval.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

SCL-02 | Admin can change dex address

Category	Severity	Location	Status
Centralization / Privilege	Informational	strategies/StrategyCompLev.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 84	Acknowledged

Description

The contract's admin has the privilege to change dex's address for each reward token.

Recommendation

No recommendations.

Alleviation

The team revisited the codebase and safe-guarded the functionality that changes dex address to be only callable through TimeLock contract. The TimeLock contract is handled by the Admin and hence the functionality to change dex address is not fully decentralized.

SCL-03 | Incorrect conditional

Category	Severity	Location	Status
Logical Issue	Minor	strategies/StrategyCompLev.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 301	Resolved

Description

The conditional on the aforementioned line is incorrect as if the `_targetSupply` is greater than `unleveraged` but less than `supplied` then the condition on L311 will never evaluate to `true` resulting in ineffectual call of the function.

Recommendation

We advise to revisit the conditional on `L301` such that the `_targetSupply` is greater than `supplied`.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

SCL-04 | Rewards are not claimed and transferred in migration of strategy

Category	Severity	Location	Status
Volatile Code	Informational	strategies/StrategyCompLev.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 655	Resolved

Description

The function on the aforementioned line migrates strategy to a new address by transferring its token balance to the new strategy address. The transferred funds does not involve the possible rewards accrued by strategy

Recommendation

We advise to revisit the `migrate` function and claim rewards before transferring the funds to new strategy address.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`. The team added boolean `claimRewardsOnMigrate`. Rewards is claimed on migrate when `claimRewardsOnMigrate` is `true`. If `false`, we will call `claimRewards` before migration. If there are significant amount of rewards to be claimed after migration, we can call `claimRewards` again, re-activate the strategy and call report.

SCL-05 | Explicitly returning local variable

Category	Severity	Location	Status
Gas Optimization	Informational	strategies/StrategyCompLev.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 509	Resolved

Description

The aforementioned lines explicitly return local variables which increases overall cost of gas.

Recommendation

Since named return variables can be declared in the signature of a function, consider refactoring to remove the local variable declaration and explicit return statement in order to reduce the overall cost of gas.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

SCS-01 | Admin can change dex address

Category	Severity	Location	Status
Centralization / Privilege	Informational	strategies/StrategyConvexStEth.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 85	Acknowledged

Description

The contract's admin has the privilege to change dex's address for each reward token.

Recommendation

No recommendations.

Alleviation

The team revisited the codebase and safe-guarded the functionality that changes dex address to be only callable through TimeLock contract. The TimeLock contract is handled by the Admin and hence the functionality to change dex address is not fully decentralized.

SCS-02 | Rewards are not claimed and transferred in migration of strategy

Category	Severity	Location	Status
Volatile Code	Informational	strategies/StrategyConvexStEth.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 376	Resolved

Description

The function on the aforementioned line migrates strategy to a new address by transferring its token balance to the new strategy address. The transferred funds does not involve the possible rewards accrued by strategy

Recommendation

We advise to revisit the `migrate` function and claim rewards before transferring the funds to new strategy address.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`. The team added boolean `claimRewardsOnMigrate`. Rewards is claimed on migrate when `claimRewardsOnMigrate` is `true`. If `false`, we will call `claimRewards` before migration. If there are significant amount of rewards to be claimed after migration, we can call `claimRewards` again, re-activate the strategy and call report.

SCS-03 | Usage of literal for arrays' lengths

Category	Severity	Location	Status
Coding Style	Informational	strategies/StrategyConvexStEth.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 20, 27, 293, 392	Resolved

Description

The aforementioned lines declare fixed length arrays and utilize integer literals to specify their lengths.

Recommendation

We advise to introduce a constant variable and utilize it to specify the lengths of fixed length arrays. This will increase the legibility of codebase.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

SCS-04 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	Informational	strategies/StrategyConvexStEth.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 75~76	Declined

Description

The aforementioned lines read storage variable `dex[_i]` inefficiently which can optimized by storing it in a local variable and then utilizing it.

Recommendation

We advise to make use of local variables to store storage values where they are used multiple times for reducing gas costs.

Alleviation

The team did not consider the recommendation stating that the gas savings are insignificant.

SCS-05 | Explicitly returning local variable

Category	Severity	Location	Status
Gas Optimization	Informational	strategies/StrategyConvexStEth.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 104, 230	Resolved

Description

The aforementioned lines explicitly return local variables which increases overall cost of gas.

Recommendation

Since named return variables can be declared in the signature of a function, consider refactoring to remove the local variable declaration and explicit return statement in order to reduce the overall cost of gas.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

SES-01 | Events are not emitted for state variables assignments

Category	Severity	Location	Status
Volatile Code	Informational	StrategyEth.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 48	Declined

Description

The constructor on the aforementioned line assigns contract's state variables but does not emit their corresponding events.

Recommendation

We advise to emit the events corresponding to the state variables that are assigned in the body of aforementioned constructor.

Alleviation

The team did not consider our recommendation.

SES-02 | Lack of validation for function parameter

Category	Severity	Location	Status
Logical Issue	Minor	StrategyEth.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 122, 132	Resolved

Description

The address type parameters of the functions on aforementioned lines are used to update contract's state yet they are not validated against zero address value. If they are passed as zero address then it will result in unwanted state of the contract.

Recommendation

We advise to validate the address type function parameters of the aforementioned functions against zero address value.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

STR-01 | Admin can change dex address

Category	Severity	Location	Status
Centralization / Privilege	Informational	strategies/StrategyConvexUsdp.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 107	Acknowledged

Description

The contract's admin has the privilege to change dex's address for each reward token.

Recommendation

No recommendations.

Alleviation

The team revisited the codebase and safe-guarded the functionality that changes dex address to be only callable through TimeLock contract. The TimeLock contract is handled by the Admin and hence the functionality to change dex address is not fully decentralized.

STR-02 | Rewards are not claimed and transferred in migration of strategy

Category	Severity	Location	Status
Volatile Code	Informational	strategies/StrategyConvexUsdp.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 407	Resolved

Description

The function on the aforementioned line migrates strategy to a new address by transferring its token balance to the new strategy address. The transferred funds does not involve the possible rewards accrued by strategy

Recommendation

We advise to revisit the `migrate` function and claim rewards before transferring the funds to new strategy address.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`. The team added boolean `claimRewardsOnMigrate`. Rewards is claimed on migrate when `claimRewardsOnMigrate` is `true`. If `false`, we will call `claimRewards` before migration. If there are significant amount of rewards to be claimed after migration, we can call `claimRewards` again, re-activate the strategy and call report.

STR-03 | Usage of literal for arrays' lengths

Category	Severity	Location	Status
Coding Style	Informational	strategies/StrategyConvexUsdp.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 21, 27, 324, 423	Resolved

Description

The aforementioned lines declare fixed length arrays and utilize integer literals to specify their lengths.

Recommendation

We advise to introduce a constant variable and utilize it to specify the lengths of fixed length arrays. This will increase the legibility of codebase.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

STR-04 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	Informational	strategies/StrategyConvexUsdp.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 97~98	Declined

Description

The aforementioned lines read storage variable `dex[_i]` inefficiently which can optimized by storing it in a local variable and then utilizing it.

Recommendation

We advise to make use of local variables to store storage values where they are used multiple times for reducing gas costs.

Alleviation

The team did not consider the recommendation stating that the gas savings are insignificant.

STR-05 | Explicitly returning local variable

Category	Severity	Location	Status
Gas Optimization	Informational	strategies/StrategyConvexUsdp.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 126, 259	Resolved

Description

The aforementioned lines explicitly return local variables which increases overall cost of gas.

Recommendation

Since named return variables can be declared in the signature of a function, consider refactoring to remove the local variable declaration and explicit return statement in order to reduce the overall cost of gas.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

STT-01 | Events are not emitted for state variables assignments

Category	Severity	Location	Status
Volatile Code	Informational	Strategy.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 46	Declined

Description

The constructor on the aforementioned line assigns contract's state variables but does not emit their corresponding events.

Recommendation

We advise to emit the events corresponding to the state variables that are assigned in the body of aforementioned constructor.

Alleviation

The team did not consider our recommendation.

STT-02 | Lack of validation for function parameter

Category	Severity	Location	Status
Logical Issue	Minor	Strategy.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 117, 127	Resolved

Description

The address type parameters of the functions on aforementioned lines are used to update contract's state yet they are not validated against zero address value. If they are passed as zero address then it will result in unwanted state of the contract.

Recommendation

We advise to validate the address type function parameters of the aforementioned functions against zero address value.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

STT-03 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	Informational	Strategy.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 153~154	Declined

Description

The aforementioned lines read storage variable `fundManager` inefficiently which can optimized by storing it in a local variable and then utilizing it.

Recommendation

We advise to make use of local variables to store storage values where they are used multiple times for reducing gas costs.

Alleviation

The team decline the recommendation stating the gas savings are low.

TLS-01 | Data location can be changed from `memory` to `calldata`

Category	Severity	Location	Status
Gas Optimization	Informational	TimeLock.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 76, 104	Resolved

Description

The aforementioned lines specify `memory` as data location for the function parameter `data`. The `data` is received externally in `calldata` and hence the aforementioned parameters can have their data location changed to `calldata` to save gas cost associated with copying of bytes from `calldata` to `memory`.

Recommendation

We advise to change data location of the aforementioned parameters from `memory` to `calldata` to save gas cost associated with copying of parameters from `memory` to `calldata`.

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

TLS-02 | Ether amount is not validated

Category	Severity	Location	Status
Volatile Code	Minor	TimeLock.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 183	Resolved

Description

The function on the aforementioned line executes relayed transaction and sends `ether` along the relayed transaction yet it does not validate if the forwarding `ether` amount is received by function call or the contract has sufficient `ether` balance.

Recommendation

We advise to introduce a check ensuring that either the function call received the forwarding `ether` or the contract has sufficient balance to successfully execute the relayed call.

```
require(
    msg.value == value
    || value <= address(this).balance,
    "not enough ether balance"
);
```

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

TLS-03 | Ether amount is not validated

Category	Severity	Location	Status
Volatile Code	Minor	TimeLock.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 196	Resolved

Description

The function on the aforementioned line executes relayed transactions and sends `ether` along the relayed transactions yet it does not validate if the forwarding `ether` amount is received by function call or the contract has sufficient `ether` balance.

Recommendation

We advise to introduce a check ensuring that either the function call received the forwarding `ether` or the contract has sufficient balance to successfully execute the relayed call.

```
uint256 requiredEtherBalance;
for (uint i = 0; i < targets.length; i++) {
    requireEtherBalance += values[i];
}

require(
    msg.value == requiredEtherBalance
    || requiredEtherBalance <= address(this).balance,
    "not enough ether balance"
);
```

Alleviation

Alleviations are applied as of commit hash `0cdc6074ac49797b3d5a30d5243caefd29fb0563`.

TLS-04 | Contract accepts arbitrary `ether`

Category	Severity	Location	Status
Volatile Code	Minor	TimeLock.sol (d1af693b837774c11c26ba930efc2c16f9a3346b): 38	Declined

Description

The `receive` function on the aforementioned line allows contract to accept arbitrary `ether`.

Recommendation

We advise to introduce a check ensuring that only a whitelisted address is able to sent plain `ether` to avoid any address from mistakenly sending the `ether`.

Alleviation

The team did not consider the recommendation stating "Accidentally sent ETH can be sent back by time lock (queue + execute)".

UTS-01 | Possibility of Replay Attack in `Permit`

Category	Severity	Location	Status
Logical Issue	Minor	UnagiiToken.vy (d1af693b837774c11c26ba930efc2c16f9a3346b): 200~208	Resolved

Description

The `permit` function performs the operation of deriving signer address from the signature values of `v`, `r` and `s`. The state variable `DOMAIN_SEPARATOR` that is used to calculate hash has a value of `chainid` that is derived only once in the constructor, which does not change after contract deployment. The issue arises in the event of fork when the cross-chain replay attacks can be executed. The attack scenario can be thought of as if a fork of Ethereum happens and two different networks have id of for example `1` and `9`. The `chainid` coded in `DOMAIN_SEPARATOR` will be the same on contracts residing in both of the forks. If the chainid `1` is stored in the contract then the `permit` transaction signed for chainid `1` will be executable on both of the forks.

Recommendation

We advise to construct the `DOMAIN_SEPARATOR` hash inside the permit function so the current `chainid` could be fetched and only the transactions signed for current network could succeed.

Alleviation

The development team opted to consider our references and updated the `DOMAIN_SEPARATOR` hash inside the permit function as proposed.

UTS-02 | Susceptible to Signature Malleability

Category	Severity	Location	Status
Volatile Code	Minor	UnagiiToken.vy (d1af693b837774c11c26ba930efc2c16f9a3346b): 200~208	Resolved

Description

The signature malleability is possible within the Elliptic Curve cryptographic system. An Elliptic Curve is symmetric on the X-axis, meaning two points can exist with the same `X` value. In the `r`, `s` and `v` representation this permits us to carefully adjust `s` to produce a second valid signature for the same `r`, thus breaking the assumption that a signature cannot be replayed in what is known as a replay-attack.

Recommendation

We advise to utilize a `recover()` function similar to that of the `ECDSA.sol` implementation of OpenZeppelin.

Alleviation

The development team opted to consider our references, implemented and utilized the `recover()` function as proposed.

UTS-03 | Missing `nextTimeLock` Clearance

Category	Severity	Location	Status
Volatile Code	Minor	UnagiiToken.vy (d1af693b837774c11c26ba930efc2c16f9a3346b): 127	Resolved

Description

The `acceptTimeLock` function does not clear `self.nextTimeLock`, which allows the `nextTimeLock` address to repeatedly call the `acceptTimeLock` function until `self.nextTimeLock` is changed.

Recommendation

We advise to revise the `acceptTimeLock` function.

Alleviation

The development team opted to consider our references and reset the `nextTimeLock` state variable after accepting the role

VAU-01 | Potential Over-centralization of Functionality

Category	Severity	Location	Status
Centralization / Privilege	Minor	Vault.vy (d1af693b837774c11c26ba930efc2c16f9a3346b): 895	Resolved

Description

The linked function is meant to be used in an edge-case situation whereby the admin or time-lock can receive the excess token sent to the contract.

Recommendation

We advise this functionality to be guarded by either a time delay to ensure that the normal course of operation of the contract has progressed.

Alleviation

The development team opted to consider our references and restricted the access to the linked functions only to the time-lock address.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

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

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a constructor assignment imposing different require statements on the input variables than a setter function.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is 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 assessment. This report does 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.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND

WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS

AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.