CERTIK

Preliminary Comments

Bloq: Vesper Pools V3

Jun 26th, 2021

Table of Contents

Summary

Overview

Project Summary
Audit Summary
Vulnerability Summary
Audit Scope

Findings

ACV-01 : Inefficient storage read
ACV-02 : Mutability Specifiers Missing
AMS-01 : Inefficient storage read
AMS-02 : Potential unsafe allocation of allowance
AMS-03 : Governor has privilege to update the addresses in Aave Strategy
ASV-01 : Potential unsafe allocation of allowance
ASV-02 : Governor has privilege to update the addresses in Aave Strategy
AVS-01 : Governor has privilege to update the addresses in Aave Strategy
CMS-01 : Potential unsafe allocation of allowance
CPM-01 : Inefficient storage read for state array's length
CPM-02 : Inefficient code
CPM-03 : Usage of literal for arrays' lengths
CPM-04 : Unused function
CPM-05 : Depositing amounts are not validated
CPS-01 : Explicitly returning local variable
CPS-02 : Documentation discrepancy
CPS-03 : Inefficient storage read for state array's length
CPS-04 : Inefficient storage read
CPS-05 : Inefficient storage read
CPS-06 : Incorrect amount calculation
CPS-07 : Potential unsafe allocation of allowance
CSI-01 : Usage of `approve` instead of `safeApprove`
CSV-01 : Inefficient storage read
CSV-02 : Potential unsafe allocation of allowance
MSV-01 : Lack of validation for function parameter
MSV-02 : Inefficient storage read
MSV-03 : Potential unsafe allocation of allowance

MSV-04 : Redundant Statements
PEC-01 : Unlocked Compiler Version
PRV-01 : Lack of validation for function parameter
PRV-02 : `require` statement can be substituted with `modifier`
PRV-03 : Inefficient storage read
PRV-04 : Inefficient storage read
PST-01 : Lack of validation for constructor parameter
PST-02 : Lack of validation for function parameter
PST-03 : Unnecessary use of conditional
PST-04 : Data location can be changed from `memory` to `calldata`
PST-05 : Inheritance order does not allow expanding of `PoolStorageV1` contract with additional storage structures
STR-01 : Inefficient storage read
STR-02 : Inefficient storage read
STR-03 : Inefficient storage read
VPB-01 : Lack of validation for function parameter
VPB-02 : Return value of function call is ignored
VPB-03 : Inefficient storage read for state array's length
VPB-04 : Explicitly returning local variable
VPB-05 : Inefficient storage read
VPB-06 : Inefficient storage read
VPB-07 : Inefficient storage read
VPB-08 : Governor can change withdraw fee
YSV-01 : Unlocked Compiler Version

Appendix

Disclaimer

About

Summary

This report has been prepared for Bloq: Vesper Pools V3 smart contracts, to discover issues and vulnerabilities in the source code of their Smart Contract 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 relating to gas optimization and code legibility. There are 14 minor findings and 1 medium finding. The minor findings comprise the lack of validation for function parameters, ignoring return value of the function call, lack of check on the depositing amounts and potential unsafe allowances. The medium finding comprise the incorrect calculation of LP amount that is used to specify the minimum amount to receive when depositing liquidity.

Overview

Project Summary

Project Name	Bloq: Vesper Pools V3
Description	The audited codebase comprise ERC20 Pool Token contract and strategies of Aave, Compound, Vesper, Cream, Yearn and Curve, and the contracts that allow the interaction of strategies with Maker protocol. The users can deposit collateral in Pool Tokens and earn LP tokens in returns. The deposited collateral in Pool Token contract is then sent to the corresponding strategies to earn interest on their respective platforms. The collateral sent to strategies by Pool Token represent debt of strategies. The strategies report loss when they hold less collateral than their debt and report profit when they have more collateral than the debt. The strategies reporting profit drive the Pool Token's share price up and the LP holders can redeem their tokens for higher amount of collateral asset.
Platform	Ethereum
Language	Solidity
Codebase	https://github.com/bloqpriv/vesper-pools-v3/tree/dee2925c284f90ddc63df55d1fece236db094d5a/contracts
Commit	dee2925c284f90ddc63df55d1fece236db094d5a

Audit Summary

Delivery Date	Jun 26, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

Vulnerability Level	Total Count	Pending	Partially Resolved	Resolved	Acknowledged	Declined
● Critical	0	0	0	0	0	0
● Major	0	0	0	0	0	0
● Medium	1	1	0	0	0	0
● Minor	15	15	0	0	0	0
● Informational	34	34	0	0	0	0
● Discussion	0	0	0	0	0	0

Audit Scope

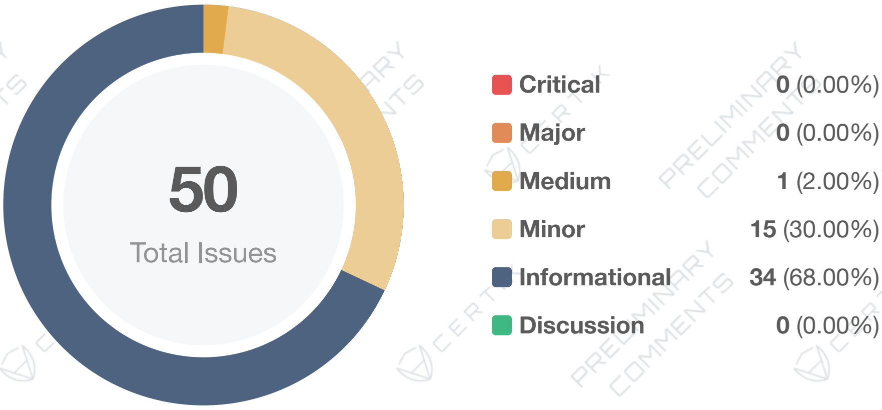
ID	file	SHA256 Checksum
GOV	Governed.sol	4d00ce81ba084144c7357282654b03e95cf71047d36d08a2bc2f6f39b73b2dc9
PAU	Pausable.sol	c84d80b10bd7f0575b60d3830c23362466ca160876611fd70a5cf239c3256285
IAV	interfaces/aave/IAave.sol	f37fc94ab102d148da318789b2990fb48b45e0d6a5b6acc7ed86ed97d2679bac
IAA	interfaces/aave/IAaveV1.sol	9da7e4bd62ceb9a87948ed5da395e28e23401a48bc309c1de7a5b47cdcf70dd2
IAL	interfaces/bloq/IAddressList.sol	f431040edba9585b390bb055b90e2156b1aa906f6e9fb706dc27096e52ee56c7
IAF	interfaces/bloq/IAddressListFactory.sol	9dabf087946020b8b9e416059242af08504bd4b6d6bf66e95e8c10f9572367ee
ISM	interfaces/bloq/ISwapManager.sol	1573a2917197c9091e6037006e8708a4d83cfe89aa014406d87df9ae67c63ae4
IAG	interfaces/chainlink/IAggregatorV3.sol	28d620b45eb4ef98d469cc355afaee129e4a9cd1f9c9b511af24edbf994272f0
ICV	interfaces/compound/ICompound.sol	24ad2de6b3e93fe3496e5a2219422b6d0e4a0d497d1047559604e65d78b84f18
ILG	interfaces/curve/ILiquidityGaugeV2.sol	44b4e70fc5111479bf490a1447c331ffb90a3076ae54a31f9ada80e00dee752b
ISS	interfaces/curve/IStableSwap3Pool.sol	eb12c116c882f1b9d4a33f4c97e421ccec3da54cbc22f520ee2f282d45400b08
ITM	interfaces/curve/ITokenMinter.sol	090fe54da8d125ef6591b51baa749e41836a2fa7ee22954a100c1843d37f4555
IMD	interfaces/maker/IMakerDAO.sol	cb95c17173d79716699c471e494070e1861381a5303ddaf27adbe1da0c65822b
ITV	interfaces/token/IToken.sol	990f1dc0ff99b1294179d1798e32982dca743551d341892860ce9e744f13a07b
IUV	interfaces/uniswap/IUniswapV2Factory.sol	4163a0fa308dd0479bb885be6f96cfe086731b4c47049784aaef4ce748ef40e0
IUR	interfaces/uniswap/IUniswapV2Router01.sol	d10f2bcaa65fec50ac7168e12462626008dcf09c282990b5a20cfbbeb119278b
IVR	interfaces/uniswap/IUniswapV2Router02.sol	246d587649b51d024f7c633056a9741f65e875a2ba7d408b8b2dbbc7afce974a
ICM	interfaces/vesper/ICollateralManager.sol	3a32ad21e5797698a98306e18ba3e6148a8d848f2b9bddc03c8f0f29270a2bde
IPR	interfaces/vesper/IPoolRewards.sol	87a9784ec3204c73b37f043a6a210237d0a07a8f365c534dba6d32aaf44c97b8
ISV	interfaces/vesper/IStrategy.sol	8cd669e9cf4716c5cb44eecb7cec605ec00051adbdd31224ba00981ff7a345d2
IVP	interfaces/vesper/IVesperPool.sol	c83cb1df5e3d27b7e7b0506268a54ec7b9a7701e11592003e3454708b1c79c97
IYT	interfaces/yearn/IYToken.sol	306d03607a44a8aeec276bd132209848709de6ff77a3b965d9f746d16fb0e758

ID	file	SHA256 Checksum
ERR	pool/Errors.sol	a87e84bb014d6e60edf5449c860106013281b6f73f0da59cc5321acb09b3e7dc
PER	pool/PoolERC20.sol	b8c8903f9832f9a7b3cb537ba375735606748caf90fceb4c8013cf332664e46f
PEC	pool/PoolERC20Permit.sol	d902c35118c30139571f4b0898e8e6a28e18f81e2561ec3c2435c3aeac9c5d8b
PRV	pool/PoolRewards.sol	603fee46fc0fe7a3e3e7a5c0437cc1d4e9bb2018ee64cfc4588e682dc567621c
PST	pool/PoolShareToken.sol	006b33548714028a5a7e2cd1f3ddb087a8224d8f179b862fa741201bf1a46c4a
PSV	pool/PoolStorage.sol	c4666cbdef24444bcecf4245a0eee5b5be137f678f2b96d2dd61a202014ae1ea
VET	pool/VETH.sol	5b262467fef932955008fea0c87ce9d4388b8b62a09e6be97f5ab7b7f421599b
VPV	pool/VPool.sol	c586699ffcd9905b28249398e2fba34a14748b737435e583b1ff100d3b2b8e5b
VPB	pool/VPoolBase.sol	b638792ba4d2c86ea670515c718f8183e97a95b0b9386f5a08c7a72f02e87e46
STR	strategies/Strategy.sol	6fdb2a105e1132deb3dc5c36b4fe4d7ac12f21c88ba479bfde2fb3fd886c652b
ACV	strategies/aave/AaveCore.sol	0e8eb5cb5b7b5b1cb4720cb66acf11118a3824ac2f20bf97b04126e9f82a9338
ASV	strategies/aave/AaveStrategy.sol	ccdaa06b7295ae8b4f422d631df0ce519f7355d6ebb38ecf5e05f1a4aad5619d
ASD	strategies/aave/AaveStrategyDAI.sol	76e0145170e31b7176a49fb559207baab9e354173d2b31d00ac817b2a8a18e81
ASU	strategies/aave/AaveStrategyUSDC.sol	9b4ce806158725c942167b8f2aa802a5725335a0009bbff9c969f417190ebbd4
AVS	strategies/aave/AaveV1Strategy.sol	11cfa676afd3331980d49aff21c609628ab6d8f467f8024ac555bcbd31af63ad
AVU	strategies/aave/AaveV1StrategyUSDC.sol	3aedb0919501ef8c9b70b2d94c9588aa3e3ae9933823fbdc2759079eeb9ad896
CSV	strategies/compound/CompoundStrategy.sol	0d0f44b6c3606d547b7737855c68cb722d7602b2bf2241287d60a770b804c9f1
CSD	strategies/compound/CompoundStrategyDAI.sol	c659bcf090b25e39bdfd28ac4cff7feaaf8db1c3e1aec8c2b0dd7a8ca388f29e
CSE	strategies/compound/CompoundStrategyETH.sol	0807b6e71b33954b36a20fad306a275dc9b8ec6039a69397da452c87a732c282
CSU	strategies/compound/CompoundStrategyUNI.sol	b5fbb6bf9c8dbe8cff519294b2c33c133ec15cbf2a71ea96fde2592482dad140
CSS	strategies/compound/CompoundStrategyUSDC.sol	90638e459b3a3841ecd827c5658e140819bdda7b0c660b6b33c5b9028aefd792

ID	file	SHA256 Checksum
CST	strategies/compound/CompoundStrategyUSDT.sol	703a45789b484844b6d98813f5d56660d616488480ab4a0feae5408afe65f1ee
CSW	strategies/compound/CompoundStrategyWBTC.sol	4328da01d8c57feac1e46bae07d86267388351b2b1649129e319df12ee9937f5
CSI	strategies/cream/CreamStrategy.sol	f008789b3713a363174d960d8e7c996cc74928492601a363a88b516bf25ed937
CSA	strategies/cream/CreamStrategyDAI.sol	072d44ae9761562edcb6b1a04301f612f6a781e349026d3acb0bb8f2b7ab35e0
CSH	strategies/cream/CreamStrategyETH.sol	e6cefcec9f909bc0b5b39d636e05b27938528e8953fea61183a7cefa3d7404e5
CSL	strategies/cream/CreamStrategyLINK.sol	c7a391d755cec1349cdd30e6cbd77c93cfad2a23d3fcc9578d2029be3acc0ca1
CSC	strategies/cream/CreamStrategyUSDC.sol	59cf966a22278e3a8560b35fd69ba4b19122da8a289428cdc8d9d406cea4d06c
CSB	strategies/cream/CreamStrategyWBTC.sol	44318e1fa58ebbc9908afdcf644555834e4a3ee498a09d159a1643cd3da1bbb7
CPM	strategies/curve/Crv3PoolMgr.sol	84a89f8f773db1cdfec1e28bf7a461515ba9774b1377ed7a39f7205e81c28319
CPS	strategies/curve/Crv3PoolStrategy.sol	c8857606f52bdff179ce09c3f2b884d59973feb4906b34696fb8b4c8ca0562a1
CPD	strategies/curve/Crv3PoolStrategyDAI.sol	e19beec8431d641e5e5f2f2e872a00633697c5f2bb7b555f2a957cfb6e4d27ce
CPU	strategies/curve/Crv3PoolStrategyUSDC.sol	66ccff45320323a51b655dadbd683fc22a52aa95593239a3c1c33eb010bfaef3
CPB	strategies/curve/CrvPoolMgrBase.sol	d6e4042f1b67dd54f079f51a689653ea6070ec95b2f74af66c0c03db8cd17618
AMS	strategies/maker/AaveMakerStrategy.sol	c2b5e5b69d661dd4c91a49799b5bf2df4c4764327f87609e5e06d1df4a7cee44
AME	strategies/maker/AaveMakerStrategyETH.sol	694c9398970e0f1b3763d75a8c2988b5702724daac1eb7d5572e232ea8216108
CMV	strategies/maker/CollateralManager.sol	bcc2484024c5905e7197ae26d7509284e8aeca4cb72abdd94b039f7a5a1dcd97
CMS	strategies/maker/CompoundMakerStrategy.sol	f7c280809cdfeecfaf5bea0d722be589fc9001ed5d55dd9f8bc82ab3ea34a243
CME	strategies/maker/CompoundMakerStrategyETH.sol	bfce3b549ad6038cccbb568fd39d85190ec0347e988b565c6365a6bf74addc2c
MSV	strategies/maker/MakerStrategy.sol	7634202586a03dd1b3102cb956f1a789e978fafe7ea3d9e1085cd2dca1c3bb8b
VMS	strategies/maker/VesperMakerStrategy.sol	130f7bf1a03f9f2939a391343d2d4a06a189c37f4efa7533b09267eab813f707

ID	file	SHA256 Checksum
VME	strategies/maker/VesperMakerStrategyETH.sol	d3181de5d23316088b33eb31252a524dd8479ce1d0dcc00cf5cb0738261ca742
YSV	strategies/yearn/YearnStrategy.sol	f25dca39af49454e4b5337cf7a47ee43b34cb1fb0ce7bb4188b0e55f44f3a74c
YSD	strategies/yearn/YearnStrategyDAI.sol	f5022717c6a72844a671e7326f0ec8305e410c2ede012924e46409c1083fbfe0
YSU	strategies/yearn/YearnStrategyUSDC.sol	a4baff9e9c45ea39972a25332b8e865dbe088a6bbb7d0d20f17be9ba56060b13

Findings



ID	Title	Category	Severity	Status
ACV-01	Inefficient storage read	Gas Optimization	● Informational	Pending
ACV-02	Mutability Specifiers Missing	Gas Optimization	● Informational	Pending
AMS-01	Inefficient storage read	Gas Optimization	● Informational	Pending
AMS-02	Potential unsafe allocation of allowance	Volatile Code	● Minor	Pending
AMS-03	Governor has privilege to update the addresses in Aave Strategy	**Centralization / Privilege**	● **Informational**	**Pending**
ASV-01	Potential unsafe allocation of allowance	Volatile Code	● Minor	Pending
ASV-02	Governor has privilege to update the addresses in Aave Strategy	**Centralization / Privilege**	● **Informational**	**Pending**
AVS-01	Governor has privilege to update the addresses in Aave Strategy	**Centralization / Privilege**	● **Informational**	**Pending**
CMS-01	Potential unsafe allocation of allowance	Volatile Code	● Minor	Pending
CPM-01	Inefficient storage read for state array's length	Gas Optimization	● Informational	Pending
CPM-02	Inefficient code	Gas Optimization	● Informational	Pending
CPM-03	Usage of literal for arrays' lengths	Coding Style	● Informational	Pending
CPM-04	Unused function	Coding Style	● Informational	Pending
CPM-05	Depositing amounts are not validated	Logical Issue	● Minor	Pending

ID	Title	Category	Severity	Status
CPS-01	Explicitly returning local variable	Gas Optimization	● Informational	Pending
CPS-02	Documentation discrepancy	Inconsistency	● Informational	Pending
CPS-03	Inefficient storage read for state array's length	Gas Optimization	● Informational	Pending
CPS-04	Inefficient storage read	Gas Optimization	● Informational	Pending
CPS-05	Inefficient storage read	Gas Optimization	● Informational	Pending
CPS-06	Incorrect amount calculation	Logical Issue	● Medium	Pending
CPS-07	Potential unsafe allocation of allowance	Volatile Code	● Minor	Pending
CSI-01	Usage of `approve` instead of `safeApprove`	Volatile Code	● Minor	Pending
CSV-01	Inefficient storage read	Gas Optimization	● Informational	Pending
CSV-02	Potential unsafe allocation of allowance	Volatile Code	● Minor	Pending
MSV-01	Lack of validation for function parameter	Volatile Code, Logical Issue	● Minor	Pending
MSV-02	Inefficient storage read	Gas Optimization	● Informational	Pending
MSV-03	Potential unsafe allocation of allowance	Volatile Code	● Minor	Pending
MSV-04	Redundant Statements	Inconsistency	● Informational	Pending
PEC-01	Unlocked Compiler Version	Language Specific	● Informational	Pending
PRV-01	Lack of validation for function parameter	Logical Issue	● Minor	Pending
PRV-02	`require` statement can be substituted with `modifier`	Language Specific	● Informational	Pending
PRV-03	Inefficient storage read	Gas Optimization	● Informational	Pending
PRV-04	Inefficient storage read	Gas Optimization	● Informational	Pending
PST-01	Lack of validation for constructor parameter	Logical Issue	● Minor	Pending
PST-02	Lack of validation for function parameter	Logical Issue	● Minor	Pending
PST-03	Unnecessary use of conditional	Coding Style	● Informational	Pending

ID	Title	Category	Severity	Status
PST-04	Data location can be changed from `memory` to `calldata`	Gas Optimization	● Informational	Pending
PST-05	Inheritance order does not allow expanding of `PoolStorageV1` contract with additional storage structures	Logical Issue, Language Specific	● Minor	Pending
STR-01	Inefficient storage read	Gas Optimization	● Informational	Pending
STR-02	Inefficient storage read	Gas Optimization	● Informational	Pending
STR-03	Inefficient storage read	Gas Optimization	● Informational	Pending
VPB-01	Lack of validation for function parameter	Logical Issue	● Minor	Pending
VPB-02	Return value of function call is ignored	Logical Issue	● Minor	Pending
VPB-03	Inefficient storage read for state array's length	Gas Optimization	● Informational	Pending
VPB-04	Explicitly returning local variable	Gas Optimization	● Informational	Pending
VPB-05	Inefficient storage read	Gas Optimization	● Informational	Pending
VPB-06	Inefficient storage read	Gas Optimization	● Informational	Pending
VPB-07	Inefficient storage read	Gas Optimization	● Informational	Pending
VPB-08	Governor can change withdraw fee	**Centralization / Privilege**	● **Informational**	**Pending**
YSV-01	Unlocked Compiler Version	Language Specific	● Informational	Pending

ACV-01 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/aave/AaveCore.sol: 64~65	ⓘ Pending

Description

The aforementioned lines read storage variable `aaveAddressesProvider` 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.

ACV-02 | Mutability Specifiers Missing

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/aave/AaveCore.sol: 17	Pending

Description

The linked variables are assigned to only once, either during their contract-level declaration or during the `constructor`'s execution.

Recommendation

For the former, we advise that the `constant` keyword is introduced in the variable declaration to greatly optimize the gas cost involved in utilizing the variable. For the latter, we advise that the `immutable` mutability specifier is set at the variable's contract-level declaration to greatly optimize the gas cost of utilizing the variables. Please note that the `immutable` keyword only works in Solidity versions `v0.6.5` and up.

AMS-01 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/maker/AaveMakerStrategy.sol: 48	ⓘ Pending

Description

The aforementioned line calls `swapManager.N_DEX()` inefficiently which involves storage read of storage read of variable `swapManager` and can optimized by storing the call's result in a local variable and then utilizing it.

Recommendation

We advise to make use of local variables to the function call's result to reduce gas cost.

AMS-02 | Potential unsafe allocation of allowance

Category	Severity	Location	Status
Volatile Code	● Minor	strategies/maker/AaveMakerStrategy.sol: 49	ⓘ Pending

Description

The aforementioned lines set maximum allowance to `routers` addresses on Swap Manager contract. As Swap Manager contract is not the core part of the system and if it is compromised then the allowances might be set for malicious router contracts and the funds of the contract will be at risk.

Recommendation

We advise not to set the maximum allowances for the router addresses and only set allowance that is needed for the swaps.

AMS-03 | Governor has privilege to update the addresses in Aave Strategy

Category	Severity	Location	Status
Centralization / Privilege	● Informational	strategies/maker/AaveMakerStrategy.sol: 36	ⓘ Pending

Description

An address with Governor role can update the addresses in the Aave Strategy that are used to interact with Aave platform.

ASV-01 | Potential unsafe allocation of allowance

Category	Severity	Location	Status
Volatile Code	● Minor	strategies/aave/AaveStrategy.sol: 68	ⓘ Pending

Description

The aforementioned lines set maximum allowance to `routers` addresses on Swap Manager contract. As Swap Manager contract is not the core part of the system and if it is compromised then the allowances might be set for malicious router contracts and the funds of the contract will be at risk.

Recommendation

We advise not to set the maximum allowances for the router addresses and only set allowance that is needed for the swaps.

ASV-02 | Governor has privilege to update the addresses in Aave Strategy

Category	Severity	Location	Status
Centralization / Privilege	● Informational	strategies/aave/AaveStrategy.sol: 35	ⓘ Pending

Description

An address with Governor role can update the addresses in the Aave Strategy that are used to interact with Aave platform.

AVS-01 | Governor has privilege to update the addresses in Aave Strategy

Category	Severity	Location	Status
Centralization / Privilege	● **Informational**	strategies/aave/AaveV1Strategy.sol: 48	ⓘ **Pending**

Description

An address with Governor role can update the addresses in the Aave Strategy that are used to interact with Aave platform.

CMS-01 | Potential unsafe allocation of allowance

Category	Severity	Location	Status
Volatile Code	● Minor	strategies/maker/CompoundMakerStrategy.sol: 75	ⓘ Pending

Description

The aforementioned lines set maximum allowance to `routers` addresses on Swap Manager contract. As Swap Manager contract is not the core part of the system and if it is compromised then the allowances might be set for malicious router contracts and the funds of the contract will be at risk.

Recommendation

We advise not to set the maximum allowances for the router addresses and only set allowance that is needed for the swaps.

CPM-01 | Inefficient storage read for state array's length

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/curve/Crv3PoolMgr.sol: 37	ⓘ Pending

Description

The aforementioned lines redundantly reads length of storage array which results in increased gas cost.

Recommendation

We advise to introduce a local variable for storing arrays' length to save gas cost.

CPM-02 | Inefficient code

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/curve/Crv3PoolMgr.sol: 39~40	ⓘ Pending

Description

The aforementioned lines retrieve `totalSupply` and `fee` and are placed inefficiently in a `for` loop that results in increased gas cost.

Recommendation

We advise to make use of local variables to store outside `for` loop to store these values and then utilize them within the loop.

CPM-03 | Usage of literal for arrays' lengths

Category	Severity	Location	Status
Coding Style	● Informational	strategies/curve/Crv3PoolMgr.sol: 16, 18, 23	ⓘ Pending

Description

The aforementioned lines declare fixed length arrays and utilize integer literals to specify their lengths of `3`.

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.

CPM-04 | Unused function

Category	Severity	Location	Status
Coding Style	● Informational	strategies/curve/Crv3PoolMgr.sol: 45	ⓘ Pending

Description

The function on the aforementioned line has `internal` visibility yet it is not used in any of the contracts within the current codebase.

Recommendation

We advise to either remove this function or use it to increase the legibility of codebase.

CPM-05 | Depositing amounts are not validated

Category	Severity	Location	Status
Logical Issue	● Minor	strategies/curve/Crv3PoolMgr.sol: 52	ⓘ Pending

Description

Although, the function on the aforementioned line is not currently utilized in the codebase but it does not validate the amounts it receives for depositing in the Curve pool. As the strategy supports only of the collateral among the three supplied, the function should validate that only the asset corresponding to `collateralId` should have non-zero amount.

Recommendation

We advise to validate the amounts asset amounts such that only the asset corresponding to `collateralId` should have non-zero amount.

CPS-01 | Explicitly returning local variable

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/curve/Crv3PoolStrategy.sol: 138	ⓘ Pending

Description

The aforementioned line explicitly return local variable 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.

CPS-02 | Documentation discrepancy

Category	Severity	Location	Status
Inconsistency	● Informational	strategies/curve/Crv3PoolStrategy.sol: 11	ⓘ Pending

Description

The comment on the aforementioned line has discrepancy as it says the strategy deposits collateral in Compound.

Recommendation

We advise to rectify the comment specifying that the collateral is deposited in Curve.

CPS-03 | Inefficient storage read for state array's length

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/curve/Crv3PoolStrategy.sol: 50, 79, 107	ⓘ Pending

Description

The aforementioned lines redundantly reads length of storage array which results in increased gas cost.

Recommendation

We advise to introduce a local variable for storing arrays' length to save gas cost.

CPS-04 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/curve/Crv3PoolStrategy.sol: 42, 44	ⓘ Pending

Description

The aforementioned lines read storage variable `depositSlippage` 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.

CPS-05 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/curve/Crv3PoolStrategy.sol: 140, 143	ⓘ Pending

Description

The aforementioned lines read storage variable `collIdx` 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.

CPS-06 | Incorrect amount calculation

Category	Severity	Location	Status
Logical Issue	● Medium	strategies/curve/Crv3PoolStrategy.sol: 119	ⓘ Pending

Description

The aforementioned line calculates minimum LP amount to receive after the liquidity is deposited in the Curve pool. The LP amount should have `18` decimals yet the calculated amount has decimals of the collateral currency which can be less than `18`.

Recommendation

We recommend to pass the amount returned from `_minimumLpPrice(_getSafeUsdRate())` to the function `convertFrom18`, so the LP amount calculated has `18` decimals.

CPS-07 | Potential unsafe allocation of allowance

Category	Severity	Location	Status
Volatile Code	● Minor	strategies/curve/Crv3PoolStrategy.sol: 108	ⓘ Pending

Description

The aforementioned lines set maximum allowance to `routers` addresses on Swap Manager contract. As Swap Manager contract is not the core part of the system and if it is compromised then the allowances might be set for malicious router contracts and the funds of the contract will be at risk.

Recommendation

We advise not to set the maximum allowances for the router addresses and only set allowance that is needed for the swaps.

CSI-01 | Usage of `approve` instead of `safeApprove`

Category	Severity	Location	Status
Volatile Code	● Minor	strategies/cream/CreamStrategy.sol: 34~35	ⓘ Pending

Description

The aforementioned lines use ERC20 `approve` function instead of using `safeApprove` function from `SafeERC20` library.

Recommendation

We advise to utilize `safeApprove` function from `SafeERC20` library on the aforementioned lines.

CSV-01 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/compound/CompoundStrategy.sol: 45	ⓘ Pending

Description

The aforementioned line calls `swapManager.N_DEX()` inefficiently which involves storage read of storage read of variable `swapManager` and can optimized by storing the call's result in a local variable and then utilizing it.

Recommendation

We advise to make use of local variables to the function call's result to reduce gas cost.

CSV-02 | Potential unsafe allocation of allowance

Category	Severity	Location	Status
Volatile Code	● Minor	strategies/compound/CompoundStrategy.sol: 46	ⓘ Pending

Description

The aforementioned lines set maximum allowance to `routers` addresses on Swap Manager contract. As Swap Manager contract is not the core part of the system and if it is compromised then the allowances might be set for malicious router contracts and the funds of the contract will be at risk.

Recommendation

We advise not to set the maximum allowances for the router addresses and only set allowance that is needed for the swaps.

MSV-01 | Lack of validation for function parameter

Category	Severity	Location	Status
Volatile Code, Logical Issue	● Minor	strategies/maker/MakerStrategy.sol: 21	ⓘ Pending

Description

The function parameter `_cm` on the aforementioned line is not validated against zero address value.

Recommendation

We advise to validate the function parameter `_cm` against zero address value.

MSV-02 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/maker/MakerStrategy.sol: 110	ⓘ Pending

Description

The aforementioned line calls `swapManager.N_DEX()` inefficiently which involves storage read of storage read of variable `swapManager` and can optimized by storing the call's result in a local variable and then utilizing it.

Recommendation

We advise to make use of local variables to the function call's result to reduce gas cost.

MSV-03 | Potential unsafe allocation of allowance

Category	Severity	Location	Status
Volatile Code	● Minor	strategies/maker/MakerStrategy.sol: 111~112	ⓘ Pending

Description

The aforementioned lines set maximum allowance to `routers` addresses on Swap Manager contract. As Swap Manager contract is not the core part of the system and if it is compromised then the allowances might be set for malicious router contracts and the funds of the contract will be at risk.

Recommendation

We advise not to set the maximum allowances for the router addresses and only set allowance that is needed for the swaps.

MSV-04 | Redundant Statements

Category	Severity	Location	Status
Inconsistency	● Informational	strategies/maker/MakerStrategy.sol: 16	ⓘ Pending

Description

The linked statements do not affect the functionality of the codebase and appear to be either leftovers from test code or older functionality.

Recommendation

We advise that they are removed to better prepare the code for production environments.

PEC-01 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	● Informational	pool/PoolERC20Permit.sol: 3	ⓘ Pending

Description

The contract has unlocked compiler version. An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to an ambiguity when debugging as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

We advise that the compiler version is instead locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.8.3` the contract should contain the following line: `pragma solidity 0.8.3;`.

PRV-01 | Lack of validation for function parameter

Category	Severity	Location	Status
Logical Issue	● Minor	pool/PoolRewards.sol: 56	ⓘ Pending

Description

The function parameters `_pool` and `_rewardToken` on the aforementioned line are not validated against zero address values.

Recommendation

We advise to validate the function parameter `_pool` and `_rewardToken` against zero address values.

PRV-02 | `require` statement can be substituted with `modifier`

Category	Severity	Location	Status
Language Specific	● Informational	pool/PoolRewards.sol: 67, 104, 119	ⓘ Pending

Description

The `require` statements on the aforementioned lines can be substituted with `modifier` to increase legibility of codebase.

Recommendation

We recommend to substitute `require` statements with `modifier`.

PRV-03 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	pool/PoolRewards.sol: 68, 79	ⓘ Pending

Description

The aforementioned lines read storage variable `rewardToken` 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.

PRV-04 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	pool/PoolRewards.sol: 72, 76, 80~83	ⓘ Pending

Description

The aforementioned lines read storage variable `rewardDuration` inefficiently which can optimized by utilizing the local variable `_rewardDuration` which contains the same value.

Recommendation

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

PST-01 | Lack of validation for constructor parameter

Category	Severity	Location	Status
Logical Issue	● Minor	pool/PoolShareToken.sol: 28	ⓘ Pending

Description

The constructor parameter `_token` on the aforementioned line is not validated against zero address value.

Recommendation

We advise to validate the constructor parameter `_token` against zero address value.

PST-02 | Lack of validation for function parameter

Category	Severity	Location	Status
Logical Issue	● Minor	pool/PoolShareToken.sol: 37	ⓘ Pending

Description

The function parameter `_token` on the aforementioned line is not validated against zero address value.

Recommendation

We advise to validate the function parameter `_token` against zero address value.

PST-03 | Unnecessary use of conditional

Category	Severity	Location	Status
Coding Style	● Informational	pool/PoolShareToken.sol: 49	ⓘ Pending

Description

The ternary conditional on the aforementioned line is not needed as the `else` part of the conditional already returns `1` for `18` decimals.

Recommendation

We advise to substitute the ternary conditional with the `else` part to increase the legibility of codebase.

PST-04 | Data location can be changed from `memory` to `calldata`

Category	Severity	Location	Status
Gas Optimization	● Informational	pool/PoolShareToken.sol: 108	ⓘ Pending

Description

The `external` function on the aforementioned line has data location of its array type parameters specified as `memory` which can be substituted with `calldata` to save gas cost associated with copying of these parameters to `memory`.

Recommendation

We advise to substitute the data location array type parameters on the aforementioned lines, from `memory` to `calldata`.

PST-05 | Inheritance order does not allow expanding of `PoolStorageV1` contract with additional storage structures

Category	Severity	Location	Status
Logical Issue, Language Specific	● Minor	pool/PoolShareToken.sol: 18	ⓘ Pending

Description

The `PoolShareToken` contract inherits from several contracts containing state variables. The inheritance order does not allow expanding of `PoolStorageV1` with additional state variables if the need arises as any additional state variables introduced in `PoolStorageV1` will overwrite the storage of the contracts that are in the inheritance order following `PoolStorageV1`.

Recommendation

We advise to place the `PoolStorageV1` contract at the end of inheritance order, so later it can be expanded with additional state variables.

STR-01 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/Strategy.sol: 109~110	ⓘ Pending

Description

The aforementioned lines read storage variable `feeCollector` 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.

STR-02 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/Strategy.sol: 120~121	ⓘ Pending

Description

The aforementioned lines read storage variable `swapManager` 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.

STR-03 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	strategies/Strategy.sol: 153, 157, 160	ⓘ Pending

Description

The aforementioned lines read storage variable `feeCollector` 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.

VPB-01 | Lack of validation for function parameter

Category	Severity	Location	Status
Logical Issue	● Minor	pool/VPoolBase.sol: 40	ⓘ Pending

Description

The function parameter `_addressListFactory` on the aforementioned line is not validated against zero address value.

Recommendation

We advise to validate the function parameter `_addressListFactory` against zero address value.

VPB-02 | Return value of function call is ignored

Category	Severity	Location	Status
Logical Issue	● Minor	pool/VPoolBase.sol: 66~67	ⓘ Pending

Description

The function calls of `add` on the aforementioned lines return `bool` value that returns the successful status of if an address is successfully in the address list or not. This return value is ignored for both of the function calls.

Recommendation

We advise to validate the returns values of the function calls against `true`.

VPB-03 | Inefficient storage read for state array's length

Category	Severity	Location	Status
Gas Optimization	● Informational	pool/VPoolBase.sol: 144, 150, 182, 423	ⓘ Pending

Description

The aforementioned lines redundantly reads length of storage array which results in increased gas cost.

Recommendation

We advise to introduce a local variable for storing arrays' length to save gas cost.

VPB-04 | Explicitly returning local variable

Category	Severity	Location	Status
Gas Optimization	● Informational	pool/VPoolBase.sol: 488	ⓘ Pending

Description

The aforementioned line explicitly return local variable 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.

VPB-05 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	pool/VPoolBase.sol: 499, 503	ⓘ Pending

Description

The aforementioned lines read storage variable `totalDebt` 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.

VPB-06 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	pool/VPoolBase.sol: 383~384	ⓘ Pending

Description

The aforementioned lines read storage variable `feeCollector` 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.

VPB-07 | Inefficient storage read

Category	Severity	Location	Status
Gas Optimization	● Informational	pool/VPoolBase.sol: 424, 434, 440	ⓘ Pending

Description

The aforementioned lines read storage variable `withdrawQueue[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.

VPB-08 | Governor can change withdraw fee

Category	Severity	Location	Status
Centralization / Privilege	● **Informational**	pool/VPoolBase.sol: 228	ⓘ **Pending**

Description

The Governor has the ability to change withdraw fee.

YSV-01 | Unlocked Compiler Version

Category	Severity	Location	Status
Language Specific	● Informational	strategies/yearn/YearnStrategy.sol: 3	ⓘ Pending

Description

The contract has unlocked compiler version. An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to an ambiguity when debugging as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

We advise that the compiler version is instead locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.8.3` the contract should contain the following line: `pragma solidity 0.8.3;`.

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.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

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 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 without CertiK's prior written consent.

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.

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.

CERTIK