SOLIDProof

Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC
Development | Marketing

MADE IN GERMANY

Chibi Finance

Audit

Security Assessment
12. May, 2023

For

CHIBI FINANCE

Disclaimer

SolidProof.io 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. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc’...)

SolidProof.io Audits 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 technology proprietors. SolidProof Audits 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.

SolidProof.io 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. SolidProof’s position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of security or functionality of the technology we agree to analyze.

Version	Date	Description
1.0	28. April 2023	• Layout project • Automated- /Manual-Security Testing • Summary
1.1	12. May 2023	• New File Added to the scope

Network
Arbitrum

Website
https://chibi.finance/

Telegram
https://t.me/chibifinance

Twitter
https://twitter.com/chibi_fi

Medium
https://medium.com/@chibifinance

Description

Chibi Finance currently offers a Yield Farming Optimizer, initiated on the Arbitrum Chain. We provide DeFi users with auto-compounded yields at optimal intervals via pooling gas costs through our smart contracts. Users will also receive BLACK CARDs (BCARD tokens) when they stake in our vaults. These BCARD emissions are a form of reward for our early users!

Project Engagement

During the 27 of April 2023, **Chibi Finance Team** engaged Solidproof.io to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. They provided Solidproof.io with access to their code repository and whitepaper.

Logo

Contract Links

v1.0

https://gitlab.com/chibifinance/chibi-finance-contracts

Commit: bbf7d9e5ea8db9e541c93a740fc148d53d291134

v1.1

https://gitlab.com/chibifinance/chibi-finance-contracts

Commit: bbf7d9e5ea8db9e541c93a740fc148d53d291134

Token Contract - https://arbiscan.io/address/0xAC2bB9C87779e14642930203fA902E1C28BA3Bc8#code

Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 - 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon aspossible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
    i) Review of the specifications, sources, and instructions provided to SolidProof to make sure we understand the size, scope, and functionality of the smart contract.
    ii) Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    iii) Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to SolidProof describe.

2. Testing and automated analysis that includes the following:
    i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    ii) Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.

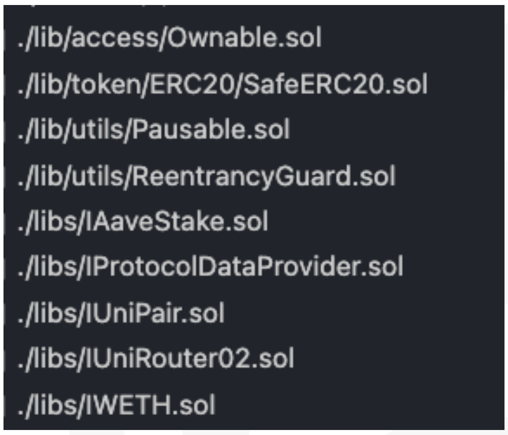
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.

4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

Used Code from other Frameworks/Smart Contracts (direct imports)

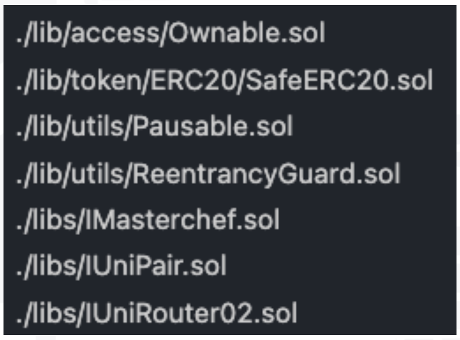
Imported packages:

StrategyAave.sol



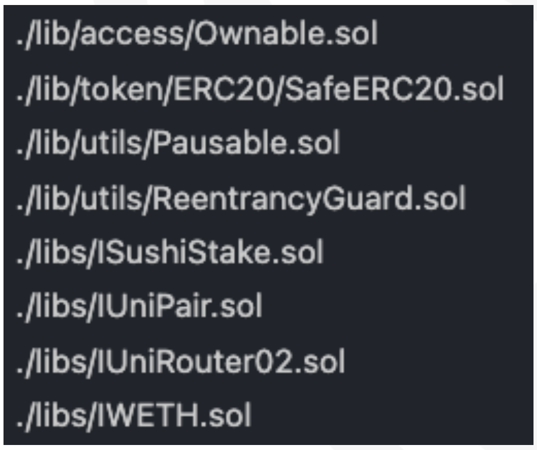
```
./lib/access/Ownable.sol
./lib/token/ERC20/SafeERC20.sol
./lib/utils/Pausable.sol
./lib/utils/ReentrancyGuard.sol
./libs/IAaveStake.sol
./libs/IProtocolDataProvider.sol
./libs/IUniPair.sol
./libs/IUniRouter02.sol
./libs/IWETH.sol
```

StrategyMasterchef.sol



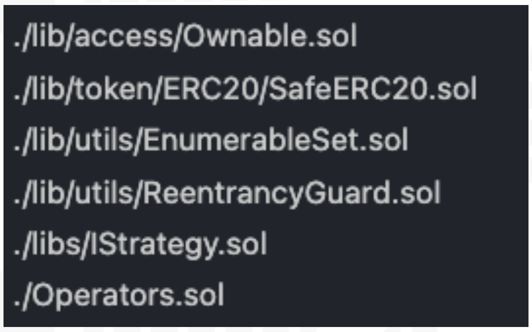
```
./lib/access/Ownable.sol
./lib/token/ERC20/SafeERC20.sol
./lib/utils/Pausable.sol
./lib/utils/ReentrancyGuard.sol
./libs/IMasterchef.sol
./libs/IUniPair.sol
./libs/IUniRouter02.sol
```

StrategySushiSwap.sol



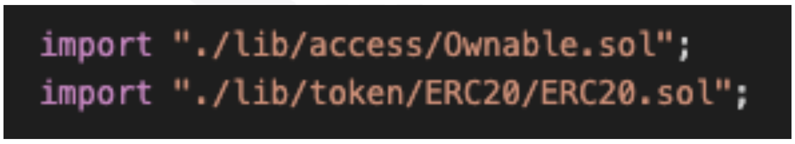
```
./lib/access/Ownable.sol
./lib/token/ERC20/SafeERC20.sol
./lib/utils/Pausable.sol
./lib/utils/ReentrancyGuard.sol
./libs/ISushiStake.sol
./libs/IUniPair.sol
./libs/IUniRouter02.sol
./libs/IWETH.sol
```

VaultChefV2.sol



```
./lib/access/Ownable.sol
./lib/token/ERC20/SafeERC20.sol
./lib/utils/EnumerableSet.sol
./lib/utils/ReentrancyGuard.sol
./libs/IStrategy.sol
./Operators.sol
```

BCARD.sol



```
import "./lib/access/Ownable.sol";
import "./lib/token/ERC20/ERC20.sol";
```

Tested Contract Files

This audit covered the following files listed below with a SHA-1 Hash.

A file with a different Hash has been modified, intentionally or otherwise, after the security review. A different Hash could be (but not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of this review.

v1.1

File Name	SHA-1 Hash
contracts/StrategySushiSwap.sol	e8a4beb9cfd8cfd3a1be5720237150fe557c50c5
contracts/lib/presets/ERC20PresetMinterPauser.sol	3ec80cf0570aa5e5c04338485f869ee80a1601e7
contracts/lib/presets/ERC721PresetMinterPauserAutoId.sol	e13dde96c5fa0f1df640bceb88ae2346ae1c9b80
contracts/lib/presets/ERC1155PresetMinterPauser.sol	916c7a0e3fb7d9cab44c2ec2b0864b3f691dee75
contracts/lib/math/Math.sol	d9eac8c88995eef8580b9e0f9a9525312e306a56
contracts/lib/math/SignedSafeMath.sol	04a33e2a9fa761128a45d3b972f68da8183946c3
contracts/lib/math/SafeMath.sol	e78d5ce176b7bbc16a0c5e78e40ab0290d44e9b1
contracts/lib/introspection/IERC165.sol	21852df340c01e66c1efda22d8fa48417f08c814
contracts/lib/introspection/ERC165.sol	29a66d5bc5dbfbebd739267dbd61c24b84919e25
contracts/lib/introspection/ERC1820Implementer.sol	50f33c6d69028ed063600cb5826ef32f23de9fca
contracts/lib/introspection/ERC165Checker.sol	b56e576e4070ec5a48fcefd7a19b9c8b52eebf8c

contracts/lib/introspection/IERC1820Implementer.sol	6508a3c50db01b935fbf9571a19cab9e08f6d246
contracts/lib/introspection/IERC1820Registry.sol	4d7aed1db481bc7d1544b323982ee052e9f4d220
contracts/lib/utils/SafeCast.sol	e72d079589fa74116bd2973af0a73a1515781754
contracts/lib/utils/Strings.sol	444a4b378eeb1b3a7ac601500210f2b09bb1d7b2
contracts/lib/utils/Create2.sol	e6e22aded5130648df30c857271a42dfa4a4fff0
contracts/lib/utils/EnumerableSet.sol	ff28b75e68496eff70710d1dad8e7699a7371540
contracts/lib/utils/Address.sol	18f99241f26986b6f3692628cd5d7045de1e5f68
contracts/lib/utils/Arrays.sol	e20c47541ba83bdd5bd8b27bc8efa6bd4a236cff
contracts/lib/utils/EnumerableMap.sol	c2fa7432211f3b4381688359ed316dcd30767d41
contracts/lib/utils/Counters.sol	b62997e751f76109ee78515ceb4f468b22ce6789
contracts/lib/utils/Pausable.sol	b2fec723ba3e3fc246e6b02c533b7ea0ef61f3a8
contracts/lib/utils/ReentrancyGuard.sol	9843042a0cf844ba3e889c038fadbcaab016eb73
contracts/lib/payment/escrow/ConditionalEscrow.sol	9e8bddd70f0c7d68974af671506ec6a6ed5d6465
contracts/lib/payment/escrow/RefundEscrow.sol	2a0495a2309bef7cf56a430e2e1f6dacd0fd8950
contracts/lib/payment/escrow/Escrow.sol	93b202bbff619d21b81b2b3104c87f561b547f24

contracts/lib/payment/PullPayment.sol	261b4593cd0b80f585c331d5512c5d738948ad65
contracts/lib/payment/PaymentSplitter.sol	53c66addc281a38c794c507ce20b2e79702aa44e
contracts/lib/GSN/Context.sol	3d2622d798014eb7a13a0a356a3a9916beca3a66
contracts/lib/GSN/GSNRecipient.sol	7469f2bab0f0ac84617734d62405112c3588d609
contracts/lib/GSN/IRelayRecipient.sol	159e531db035034aaf216078d4f6945372f25082
contracts/lib/GSN/GSNRecipientERC20Fee.sol	b89b766675894cf684d9d437b394f75dd55af3de
contracts/lib/GSN/IRelayHub.sol	e491c4b3d8c7d981f59ffcf7d266319df9f35b60
contracts/lib/GSN/GSNRecipientSignature.sol	1dd984c67663409c521897374bed79ce660a3453
contracts/lib/access/AccessControl.sol	419d421c4562edddc6fa45170c56ab57a49405d7
contracts/lib/access/Ownable.sol	ac16c3cf74a9e575f8bf20c62e0e01fa784252d0
contracts/lib/token/ERC1155/ERC1155Burnable.sol	acf1be143c2a89ffcc6371214406d3719c8b6d78
contracts/lib/token/ERC1155/ERC1155Holder.sol	43cdec327f12b1024690914a3da5ae6baffdc647
contracts/lib/token/ERC1155/IERC1155MetadataURI.sol	6abdab5daed64d97e85548abed2724796a3723ed
contracts/lib/token/ERC1155/ERC1155Pausable.sol	73ced7a58135ee23cfeb68fa377a45a24fc05ba3
contracts/lib/token/ERC1155/IERC1155.sol	f4a292db3239cfc9d295ba6d00bff56d86365b3c

contracts/lib/token/ERC1155/IERC1155Receiver.sol	873fd2cfa3510426682ba633db39017915a75e41
contracts/lib/token/ERC1155/ERC1155Receiver.sol	f393fd3ad79e0c42e420ea4f5a0c713f10cc5ee0
contracts/lib/token/ERC1155/ERC1155.sol	c90292dedd9374d4a5adabb1b6ead0981729fb33
contracts/lib/token/ERC721/ERC721Burnable.sol	e986567dfe3d14161909d6b04c15af482dc2726a
contracts/lib/token/ERC721/ERC721Holder.sol	0f14ad05489935ac3356cbe3bf78379603fc4a64
contracts/lib/token/ERC721/IERC721.sol	2bc38530b8027e1100ade4826260e07611b8a13b
contracts/lib/token/ERC721/ERC721.sol	02a9ddac69d67e3fcbb3ca7a37ea28b7adce58d7
contracts/lib/token/ERC721/ERC721Pausable.sol	0ed9f2a85b9036c3aeb51c79d8266c7f17ec96c7
contracts/lib/token/ERC721/IERC721Receiver.sol	fc8c0c0ce63b50adb36e8ac51ec75171db34e61b
contracts/lib/token/ERC721/IERC721Metadata.sol	9ca42611901c231b4acd37d67184048456113955
contracts/lib/token/ERC721/IERC721Enumerable.sol	63b8e526b2a490c58af72314704125d19023f8cc
contracts/lib/token/ERC777/IERC777.sol	b00af989b9abbc61d31ad68c0bfd7e7973c2965d
contracts/lib/token/ERC777/ERC777.sol	63ab3f3539ea40020331eac505d50c1a2f2bde64
contracts/lib/token/ERC777/IERC777Sender.sol	cd98cf9dcd04023591b0752fe9b2afa932d885f9
contracts/lib/token/ERC777/IERC777Recipient.sol	836ae8b87763e7340c149abf098ff9ed29afa2e5

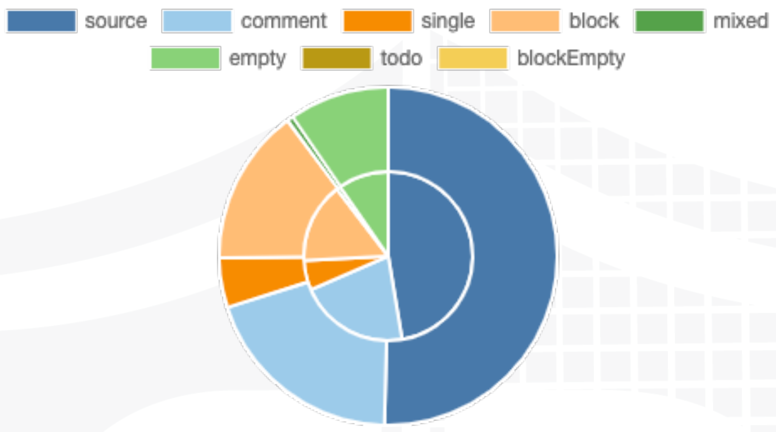
contracts/lib/token/ERC20/ERC20Capped.sol	84472978c63c5af27a39d9172e1c0bb1e8a2035a
contracts/lib/token/ERC20/ERC20Snapshot.sol	fe7b7051e8dd53f333b31a2a48969ff92ceed5cf
contracts/lib/token/ERC20/ERC20Pausable.sol	fd74fa02eb369f9da535632f6edde8fda04a695f
contracts/lib/token/ERC20/SafeERC20.sol	315ebd316565c099582f27f179edcc5b29afbf12
contracts/lib/token/ERC20/ERC20Burnable.sol	1ef08216d36fb531d9d8f72ef4c7ecd5cceff900
contracts/lib/token/ERC20/ERC20.sol	dac745cb1331b446c004f60d67edcc1e0d9edbe4
contracts/lib/token/ERC20/TokenTimelock.sol	2845880bc7753608ab3688a44ff169ed08e50b33
contracts/lib/token/ERC20/IERC20.sol	238d1a6b1bc2d2d39b8ce12da97d76cd43b7e15f
contracts/StrategyMasterchef.sol	553f333a94fe3b93486682f2740e15137a21e30e
contracts/StrategyAave.sol	a92fc7a62f0bf3dc35587f5e1328ad1c7032714e
contracts/libs/IUniRouter01.sol	d264d1932daa9b6a541e489045bb98da04dfd4fc
contracts/libs/IProtocolDataProvider.sol	0b08df29c62a6144c8ee81db7a640ac03d1c1ef7
contracts/libs/IUniRouter02.sol	3a272fa231106f958cd10f3bc2db12269e671d9c
contracts/libs/IStakingRewards.sol	0b00e73fc3d5ef1640b45eb8f026e1f494d4ef46
contracts/libs/ISushiStake.sol	3c77903f01b161c58b4dc200f3da9e52505fd7b7

contracts/libs/IStrategy.sol	b5cf0ddeebaa8d7fa2d8b8f227 2d10545fcb68f2
contracts/libs/IAaveStake.sol	b4e5be23b902bb40b94053a65 68503f4ce557480
contracts/libs/IWETH.sol	e44b165cfde112075fd1649fe1 514a8380573b0e
contracts/libs/IMasterchef.sol	5d20f559f0b9cb391d3ed1025f e9c63bbde3fc7d
contracts/libs/IUniPair.sol	a94ee4d5f0f3628efb2802be31 94ba4a37d3b9a9
contracts/VaultChefV2.sol	66d39be48ba55a9b715dd164a 3ee222a79b4ae5b
contracts/lib/cryptography/ MerkleProof.sol	2fc603f5411d8fe0d803831442 8ef25c827806f1
contracts/lib/cryptography/ ECDSA.sol	389c9c813528316fa9038e17d eb3c8dffd86a1ff
Contracts/BCARD.sol	112075fdc151371640b45eb8f0 26e1dffd86a1ff

Metrics

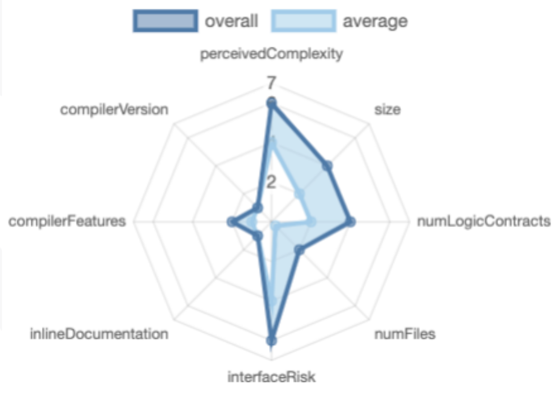
Source Lines

v1.0



Risk Level

v1.0



Capabilities

Components

🌐Public	💰Payable
351	17

External	Internal	Private	Pure	View
196	480	34	46	172

StateVariables

Total	🌐Public
176	103

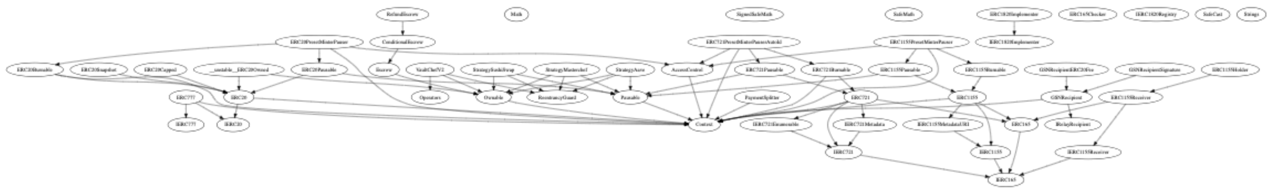
Capabilities

Solidity Versions observed	🧪 Experimental Features	💰 Can Receive Funds	🖥 Uses Assembly	💣 Has Destroyable Contracts
`0.6.12` `^0.6.0` `^0.6.2` `>=0.5.0`	`ABIEncoderV2`	`yes`	`yes` (5 asm blocks)	

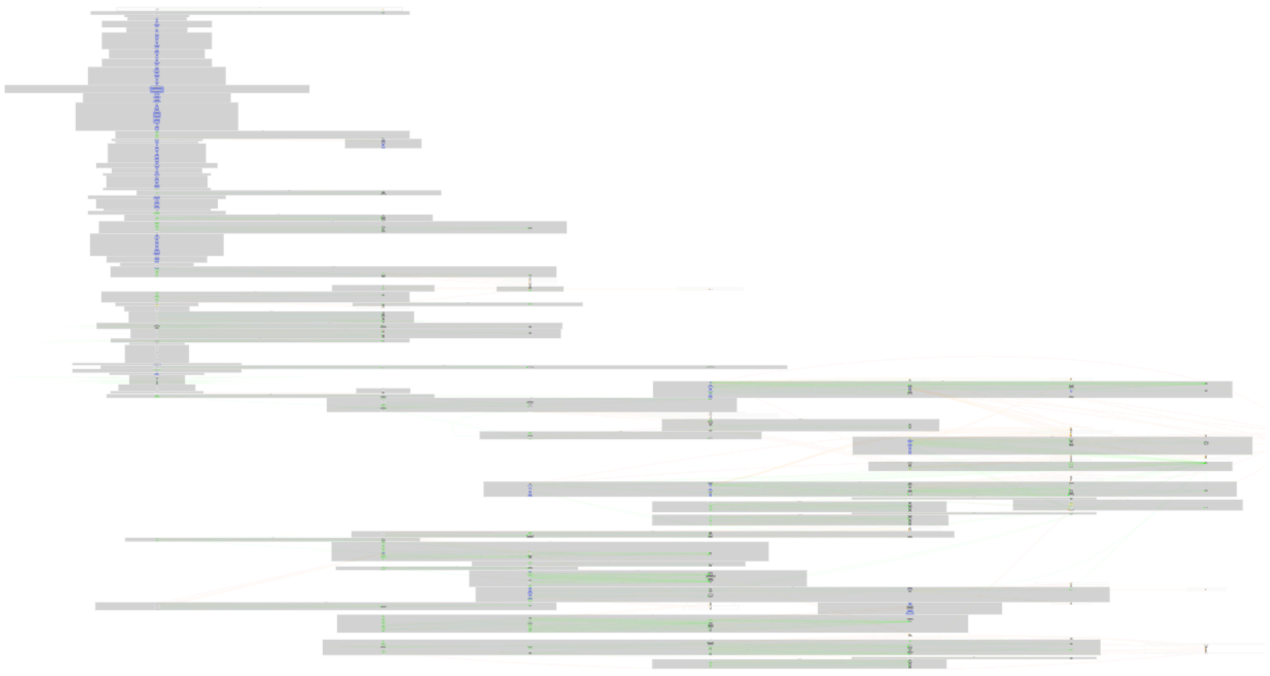
📤 Transfers ETH	⚡ Low-Level Calls	👥 DelegateCall	🧮 Uses Hash Functions	🔖 ECRecover	🌀 New/Create/Create2
`yes`			`yes`	`yes`	`yes` → `AssemblyCall:Name:create2` → `NewContract:Escrow` → `NewContract:__unstable__ERC20Owned`

♻️ TryCatch	Σ Unchecked
`yes`	

Inheritance Graph
v1.0



CallGraph
v1.0



Scope of Work/Verify Claims

The above token Team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract (usual the same name as team appended with .sol).

We will verify the following claims:

1. Overall checkup (Smart Contract Security)

Overall checkup (Smart Contract Security)

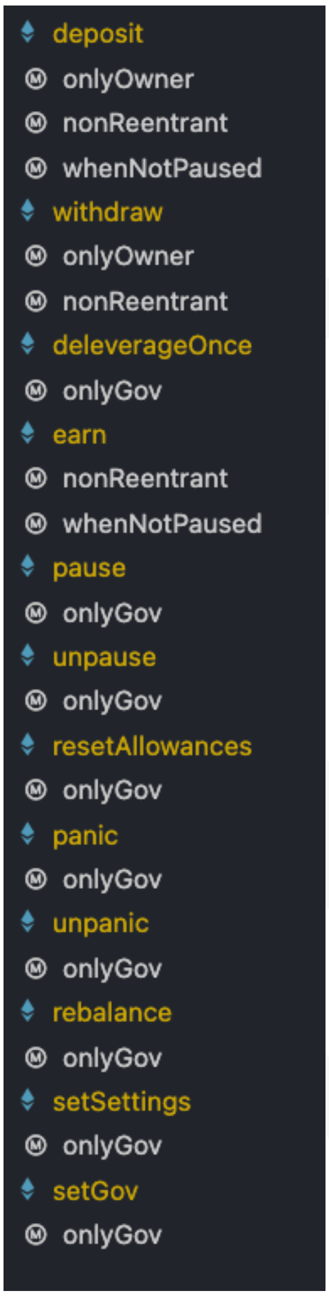
Tested	Verified
✓	✓

Legend

Attribute	Symbol
Verified / Checked	✓
Partly Verified	⚑
Unverified / Not checked	✗
Not available	–

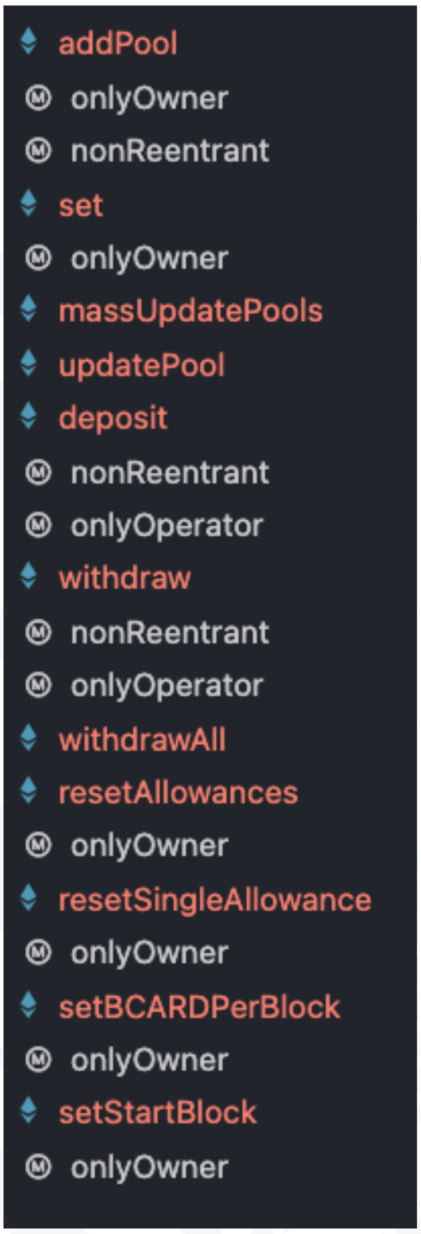
Modifiers and public functions
v1.1

StrategyAave.sol



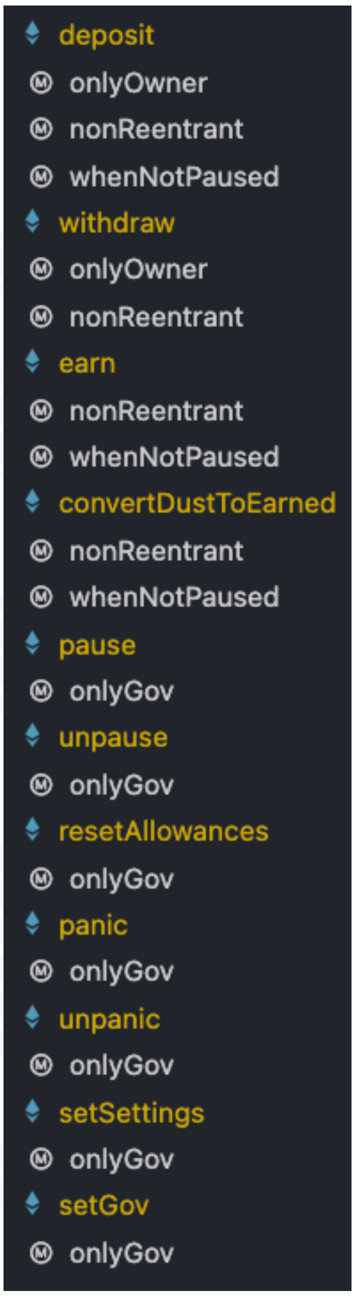
- deposit
  - onlyOwner
  - nonReentrant
  - whenNotPaused
- withdraw
  - onlyOwner
  - nonReentrant
- deleverageOnce
  - onlyGov
- earn
  - nonReentrant
  - whenNotPaused
- pause
  - onlyGov
- unpause
  - onlyGov
- resetAllowances
  - onlyGov
- panic
  - onlyGov
- unpanic
  - onlyGov
- rebalance
  - onlyGov
- setSettings
  - onlyGov
- setGov
  - onlyGov

VaultChefV2



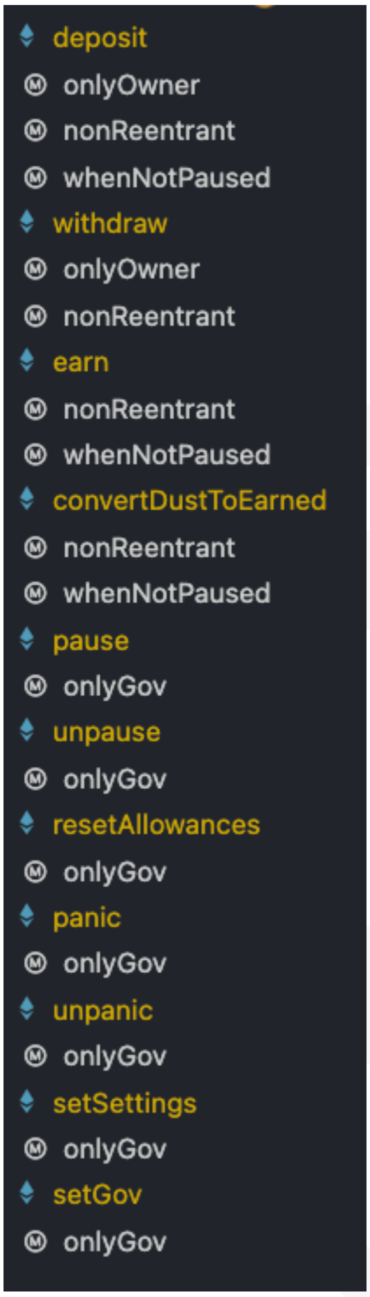
- addPool
  - onlyOwner
  - nonReentrant
- set
  - onlyOwner
- massUpdatePools
- updatePool
- deposit
  - nonReentrant
  - onlyOperator
- withdraw
  - nonReentrant
  - onlyOperator
- withdrawAll
- resetAllowances
  - onlyOwner
- resetSingleAllowance
  - onlyOwner
- setBCARDPerBlock
  - onlyOwner
- setStartBlock
  - onlyOwner

StrategyMasterChef



- deposit
  - onlyOwner
  - nonReentrant
  - whenNotPaused
- withdraw
  - onlyOwner
  - nonReentrant
- earn
  - nonReentrant
  - whenNotPaused
- convertDustToEarned
  - nonReentrant
  - whenNotPaused
- pause
  - onlyGov
- unpause
  - onlyGov
- resetAllowances
  - onlyGov
- panic
  - onlyGov
- unpanic
  - onlyGov
- setSettings
  - onlyGov
- setGov
  - onlyGov

StrategySushiSwap


deposit
onlyOwner
nonReentrant
whenNotPaused
withdraw
onlyOwner
nonReentrant
earn
nonReentrant
whenNotPaused
convertDustToEarned
nonReentrant
whenNotPaused
pause
onlyGov
unpause
onlyGov
resetAllowances
onlyGov
panic
onlyGov
unpanic
onlyGov
setSettings
onlyGov
setGov
onlyGov

Note:

❖ General fork from Polycat and AutoFramenetwork

❖ Contracts inside are the same as the polycat-contracts/Vault2, and autofarm-v2-contracts directories

- https://github.com/polycatfi/polycat-contracts/tree/master/Vault2
- https://github.com/autofarmnetwork/autofarm-v2-contracts/blob/master/AutoFarmV2.sol

- Differences between ChibiFinance, Polycat contracts are the following:
    - VaultChefV2 have the same logic as AutoFarmV2.sol with the following differences:
        - Changed token name
        - Added constructor
        - Removal of mint in 'stakedWant' function
        - Deposit/Withdraw can only be done by the operator address
        - Removal of Emergency Withdraw function
        - Added resetting allowance
    - StrategyAave have the same logic as StrategyAave.sol(PolyCat) with the following differences:
        - Removed USDC, fish, and reward address variables
        - Removed buyback rate
        - Added single deleverage functionality
        - Removed distribution of reward, and buyback
        - Removed referral code from the settings
    - StrategyMasterChef have the same logic as StrategyMasterChef.sol(PolyCat) with the following differences:
        - Removed USDC, fish, withdraw fee and reward address variables
        - Removed buyback and rewards
        - Removed distribution of reward, and buyback
    - StrategySushiSwap have the same logic as StrategySushiSwap.sol(PolyCat) with the following differences:
        - Removed USDC, fish, withdraw fee and reward address variables
        - Removed buyback and rewards
        - Removed distribution of reward, and buyback

Ownership/Authority Privileges

- ❖ *VaultChefv2.sol* -
  - Only the operator address can deposit/withdraw tokens
  - Owner can reset multiple or single allowance
  - Set BCARD tokens created per block to any arbitrary value including zero, if done so it may affect users' rewards drastically.
  - Set start block number for adding new pools.
  - Set/Update a given pool's allocation point to any arbitrary value
  - Owner can add new pools

- ❖ *StrategyAave.sol* -
  - Only owner can deposit and withdraw tokens
  - Only Gov address can manually deleverage one step
  - Gov address can pause/unpause the contract
  - Gov address can reset allowances
  - Gov address can set borrow rate and borrow depth but not more than 1% and 0.1% respectively
  - Gov address can set controller fee, but not more than 10%. Uniswap router address

- ❖ *StrategyMasterChef.sol* -
  - Only owner can deposit and withdraw tokens
  - Only Gov address can manually deleverage one step
  - Gov address can pause/unpause the contract
  - Gov address can reset allowances
  - Gov address can set borrow rate and borrow depth but not more than 1% and 0.1% respectively
  - Gov address can set controller fee, but not more than 10%. Uniswap router address

- ❖ *StrategySushiSwap.sol* -
  - Only owner can deposit and withdraw tokens
  - Only Gov address can manually deleverage one step
  - Gov address can pause/unpause the contract
  - Gov address can reset allowances
  - Gov address can set borrow rate and borrow depth but not more than 1% and 0.1% respectively
  - Gov address can set controller fee, but not more than 10%. Uniswap router address

Please check if an OnlyOwner or similar restrictive modifier has been forgotten.

Source Units in Scope

v1.0

File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score
contracts/StrategySushiSwap.sol	1		403	390	315	1	293
contracts/lib/presets/ERC20PresetMinterPauser.sol	1		87	87	30	48	43
contracts/lib/presets/ERC721PresetMinterPauserAutoId.sol	1		102	102	35	54	47
contracts/lib/presets/ERC1155PresetMinterPauser.sol	1		104	95	34	49	49
contracts/lib/math/Math.sol	1		31	31	12	15	3
contracts/lib/math/SignedSafeMath.sol	1		92	92	29	50	9
contracts/lib/math/SafeMath.sol	1		159	159	39	106	10
contracts/lib/introspection/IERC165.sol		1	24	23	3	18	3
contracts/lib/introspection/ERC165.sol	1		54	54	16	31	9

contracts/lib/introspection/ERC1820Implementer.sol	1		37	37	12	19	13
contracts/lib/introspection/ERC165Checker.sol	1		106	102	34	57	21
contracts/lib/introspection/IERC1820Implementer.sol		1	19	18	3	13	3
contracts/lib/introspection/IERC1820Registry.sol		1	111	58	28	87	17
contracts/lib/utils/SafeCast.sol	1		211	211	51	145	25
contracts/lib/utils/Strings.sol	1		34	34	22	9	18
contracts/lib/utils/Create2.sol	1		59	59	22	33	29
contracts/lib/utils/EnumerableSet.sol	1		243	243	77	136	29
contracts/lib/utils/Address.sol	1		141	126	55	87	37
contracts/lib/utils/Arrays.sol	1		47	47	24	16	6
contracts/lib/utils/EnumerableMap.sol	1		237	237	81	130	30
contracts/lib/utils/Counters.sol	1		40	40	17	17	2
contracts/lib/utils/Pausable.sol	1		90	90	29	50	14

contracts/lib/utils/ReentrancyGuard.sol	1		62	62	15	38	5
contracts/lib/payment/escrow/ConditionalEscrow.sol	1		24	22	8	11	10
contracts/lib/payment/escrow/RefundEscrow.sol	1		93	93	41	39	37
contracts/lib/payment/escrow/Escrow.sol	1		65	65	25	28	19
contracts/lib/payment/PullPayment.sol	1		69	69	17	45	20
contracts/lib/payment/PaymentSplitter.sol	1		134	134	58	55	58
contracts/lib/GSN/Context.sol	1		24	24	10	12	1
contracts/lib/GSN/GSNRecipient.sol	1		230	213	78	119	62
contracts/lib/GSN/IRelayRecipient.sol		1	76	53	42	51	9
contracts/lib/GSN/GSNRecipientERC20Fee.sol	2		152	136	69	46	65
contracts/lib/GSN/IRelayHub.sol		1	269	145	59	182	37
contracts/lib/GSN/GSNRecipientSignature.sol	1		72	56	34	16	20
contracts/lib/access/AccessControl.sol	1		217	217	58	136	43
contracts/lib/access/Ownable.sol	1		68	68	27	33	23

contracts/lib/token/ERC1155/ERC1155Burnable.sol	1		31	31	18	7	17
contracts/lib/token/ERC1155/ERC1155Holder.sol	1		18	18	10	4	7
contracts/lib/token/ERC1155/IERC1155MetadataURI.sol		1	21	20	4	13	5
contracts/lib/token/ERC1155/ERC1155Pausable.sol	1		39	30	9	17	8
contracts/lib/token/ERC1155/IERC1155.sol		1	103	81	42	77	15
contracts/lib/token/ERC1155/IERC1155Receiver.sol		1	57	25	4	30	7
contracts/lib/token/ERC1155/ERC1155Receiver.sol	1		18	18	11	4	10
contracts/lib/token/ERC1155/ERC1155.sol	1		413	358	157	141	172
contracts/lib/token/ERC721/ERC721Burnable.sol	1		25	25	9	13	11
contracts/lib/token/ERC721/ERC721Holder.sol	1		23	23	7	12	5
contracts/lib/token/ERC721/IERC721.sol		1	129	62	40	99	21
contracts/lib/token/ERC721/ERC721.sol	1		473	460	170	222	158

contracts/lib/token/ERC721/ERC721Pausable.sol	1		28	28	9	15	8
contracts/lib/token/ERC721/IERC721Receiver.sol		1	22	20	3	15	3
contracts/lib/token/ERC721/IERC721Metadata.sol		1	27	16	4	14	9
contracts/lib/token/ERC721/IERC721Enumerable.sol		1	29	20	8	16	9
contracts/lib/token/ERC777/IERC777.sol		1	188	84	69	130	27
contracts/lib/token/ERC777/ERC777.sol	1		503	441	181	186	171
contracts/lib/token/ERC777/IERC777Sender.sol		1	34	26	3	21	3
contracts/lib/token/ERC777/IERC777Recipient.sol		1	34	26	3	21	3
contracts/lib/token/ERC20/ERC20Capped.sol	1		43	43	18	19	15
contracts/lib/token/ERC20/ERC20Snapshot.sol	1		184	182	77	77	43
contracts/lib/token/ERC20/ERC20Pausable.sol	1		28	28	9	15	8
contracts/lib/token/ERC20/SafeERC20.sol	1		75	74	33	32	25

contracts/lib/token/ERC20/ERC20Burnable.sol	1		40	40	13	22	17
contracts/lib/token/ERC20/ERC20.sol	1		307	307	91	184	81
contracts/lib/token/ERC20/TokenTimelock.sol	1		67	67	29	25	20
contracts/lib/token/ERC20/IERC20.sol		1	77	26	17	57	13
contracts/StrategyMasterchef.sol	1		390	377	288	23	243
contracts/StrategyAave.sol	1		452	443	327	35	334
contracts/libs/IUniRouter01.sol		1	162	6	3	1	48
contracts/libs/IProtocolDataProvider.sol		1	17	11	8	1	13
contracts/libs/IUniRouter02.sol		1	52	8	4	1	16
contracts/libs/IStakingRewards.sol		1	29	6	3	1	25
contracts/libs/ISushiStake.sol		1	15	6	3	1	11
contracts/libs/IStrategy.sol		1	24	8	3	8	13
contracts/libs/IAaveStake.sol		1	27	6	3	1	35
contracts/libs/IWETH.sol		1	9	6	3	1	10

contracts/libs/IMasterchef.sol		1	13	6	3	1	9
contracts/libs/IUniPair.sol		1	8	6	3	1	5
contracts/VaultChefV2.sol	1		310	293	232	29	195
contracts/lib/cryptography/MerkleProof.sol	1		33	33	15	13	11
contracts/lib/cryptography/ECDSA.sol	1		83	83	28	46	35
Totals	**54**	**26**	**8646**	**7569**	**3483**	**3633**	**3013**

Legend

Attribute	Description
Lines	total lines of the source unit
nLines	normalised lines of the source unit (e.g. normalises functions spanning multiple lines)
nSLOC	normalised source lines of code (only source-code lines; no comments, no blank lines)
Comment Lines	lines containing single or block comments
Complexity Score	a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

Audit Results

Critical issues

No critical issues

High issues

No high issues

Medium issues

No medium issues

Low issues

Issue	File	Type	Line	Description
#1	All	Multiple pragma is set	—	Some of the contracts contain different pragma versions which is not recommended for deployment. We recommend to have the same pragma in all contracts and also to update the old pragma versions to the new ones.
#2	StrategySushiSwap.sol	Missing Zero Address Validation (missing-zero-check)	347	Check that the address is not zero
#3	StrategySushiSwap.sol	Missing Events Arithmetic	All	Emit an event for critical parameter changes
#4	StrategySushiSwap.sol	Old Compiler Version	3	The contract uses a very old compiler version which is not recommended for deployment as it is susceptible to known vulnerabilities

#5	Strategy SushiSwap.sol	Old Compiler Version	5	The contract uses a very old compiler version which is not recommended for deployment as it is susceptible to known vulnerabilities
#6	Strategy Aave.sol	Missing Zero Address Validation (missing-zero-check)	406	Check that the address is not zero
#7	BCARD. sol	Old Compiler Version	—	The contract uses a very old compiler version which is not recommended for deployment as it is susceptible to known vulnerabilities

Informational issues

Issue	File	Type	Line	Description
#1	All	Contract doesn't import npm packages from source (like OpenZeppelin etc.)	—	We recommend importing all packages from npm directly without flattening the contract. Functions could be modified or can be susceptible to vulnerabilities

Audit Comments

We recommend you to use the special form of comments (NatSpec Format, Follow link for more information https://docs.soliditylang.org/en/latest/natspec-format.html) for your contracts to provide rich documentation for functions, return variables and more. This helps investors to make clear what that variables, functions etc. do.

12. May 2023:

- This project consists of the following forks
    - AutoFarm
    - PolyCat
- Read whole report and modifiers section for more information
- The low issues that exist in the PolyCat and AutoFarm codebase still exist in the forked code.
- Unit tests with 100% code coverage was not provided to SolidProof so we cannot ensure complete functional correctness of the code's logic.
- We recommend **Chibi Finance** team to conduct unit and fuzz tests thoroughly to rule out possibilities of an unwanted logical and calculation errors.

- We recommend using a multisig wallet for the owner address to prevent any risk of the loss of private key
- Do your own research here

SWC Attacks

ID	Title	Relationships	Status
SWC-136	Unencrypted Private Data On-Chain	CWE-767: Access to Critical Private Variable via Public Method	PASSED
SWC-135	Code With No Effects	CWE-1164: Irrelevant Code	PASSED
SWC-134	Message call with hardcoded gas amount	CWE-655: Improper Initialization	PASSED
SWC-133	Hash Collisions With Multiple Variable Length Arguments	CWE-294: Authentication Bypass by Capture-replay	PASSED
SWC-132	Unexpected Ether balance	CWE-667: Improper Locking	PASSED
SWC-131	Presence of unused variables	CWE-1164: Irrelevant Code	PASSED
SWC-130	Right-To-Left-Override control character (U+202E)	CWE-451: User Interface (UI) Misrepresentation of Critical Information	PASSED
SWC-129	Typographical Error	CWE-480: Use of Incorrect Operator	PASSED
SWC-128	DoS With Block Gas Limit	CWE-400: Uncontrolled Resource Consumption	PASSED

SWC-127	Arbitrary Jump with Function Type Variable	CWE-695: Use of Low-Level Functionality	**PASSED**
SWC-125	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	**PASSED**
SWC-124	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	**PASSED**
SWC-123	Requirement Violation	CWE-573: Improper Following of Specification by Caller	**PASSED**
SWC-122	Lack of Proper Signature Verification	CWE-345: Insufficient Verification of Data Authenticity	**PASSED**
SWC-121	Missing Protection against Signature Replay Attacks	CWE-347: Improper Verification of Cryptographic Signature	**PASSED**
SWC-120	Weak Sources of Randomness from Chain Attributes	CWE-330: Use of Insufficiently Random Values	**PASSED**
SWC-119	Shadowing State Variables	CWE-710: Improper Adherence to Coding Standards	**PASSED**
SWC-118	Incorrect Constructor Name	CWE-665: Improper Initialization	**PASSED**
SWC-117	Signature Malleability	CWE-347: Improper Verification of Cryptographic Signature	**PASSED**

SWC-116	Timestamp Dependence	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	PASSED
SWC-115	Authorization through tx.origin	CWE-477: Use of Obsolete Function	PASSED
SWC-114	Transaction Order Dependence	CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	PASSED
SWC-113	DoS with Failed Call	CWE-703: Improper Check or Handling of Exceptional Conditions	PASSED
SWC-112	Delegatecall to Untrusted Callee	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	PASSED
SWC-111	Use of Deprecated Solidity Functions	CWE-477: Use of Obsolete Function	PASSED
SWC-110	Assert Violation	CWE-670: Always-Incorrect Control Flow Implementation	PASSED
SWC-109	Uninitialized Storage Pointer	CWE-824: Access of Uninitialized Pointer	PASSED
SWC-108	State Variable Default Visibility	CWE-710: Improper Adherence to Coding Standards	PASSED
SWC-107	Reentrancy	CWE-841: Improper Enforcement of Behavioral Workflow	PASSED
SWC-106	Unprotected SELFDESTRUCT Instruction	CWE-284: Improper Access Control	PASSED

SWC-105	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	**PASSED**
SWC-104	Unchecked Call Return Value	CWE-252: Unchecked Return Value	**PASSED**
SWC-103	Floating Pragma	CWE-664: Improper Control of a Resource Through its Lifetime	**NOT PASSED**
SWC-102	Outdated Compiler Version	CWE-937: Using Components with Known Vulnerabilities	**NOT PASSED**
SWC-101	Integer Overflow and Underflow	CWE-682: Incorrect Calculation	**PASSED**
SWC-100	Function Default Visibility	CWE-710: Improper Adherence to Coding Standards	**PASSED**