SOLIDProof

Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC
Development | Marketing

MADE IN GERMANY

Yield Genius

Audit

Security Assessment
21. March, 2023

For

SolidProof_io

@solidproof_io

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	14. March 2023 - 17. March 2023	· Layout project · Automated- /Manual-Security Testing · Summary
1.1	21. March 2023	· Reaudit

Network

Arbitrum

Website

https://www.yieldgenius.io/

Telegram

https://t.me/YieldGenius

Twitter

https://twitter.com/YieldGenius_io

Description

We are a Yield Optimizer Platform, offering an automated process of finding the best yield farming opportunities across multiple protocols, allowing users to earn the highest return on their investment.

Project Engagement

During the 12th of March 2023, **Yield Genius 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 Link

v1.0

- https://github.com/yieldgenius/yieldgenius-contracts
- Commit: 53c6fe6

v1.1

- https://github.com/yieldgenius/yieldgenius-contracts
- Commit: 79bce54

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.

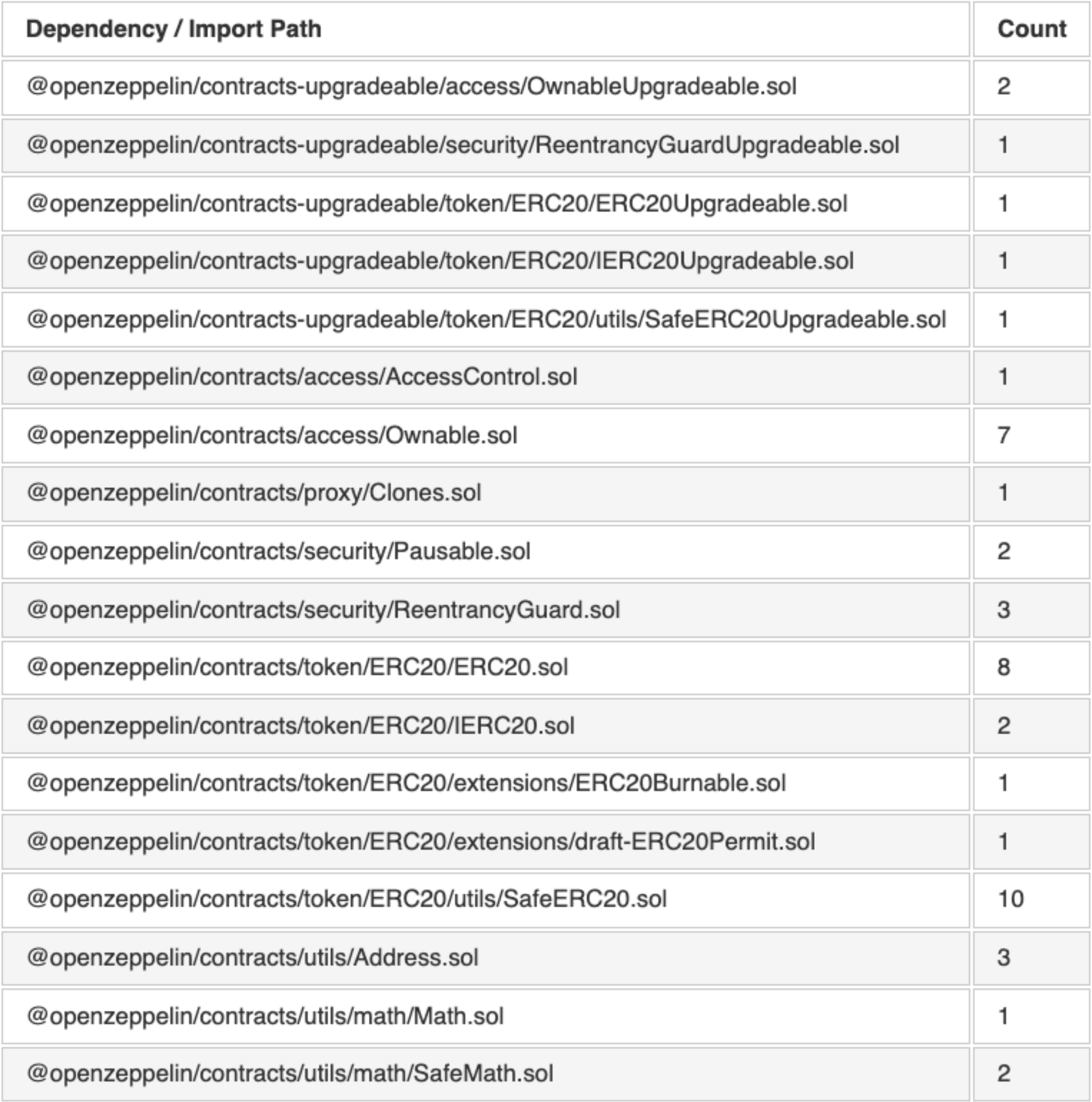
2. Testing and automated analysis that includes the following:
    i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    ii) Symbolic execution, which is analysing a program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



Dependency / Import Path	Count
@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol	2
@openzeppelin/contracts-upgradeable/security/ReentrancyGuardUpgradeable.sol	1
@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol	1
@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol	1
@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol	1
@openzeppelin/contracts/access/AccessControl.sol	1
@openzeppelin/contracts/access/Ownable.sol	7
@openzeppelin/contracts/proxy/Clones.sol	1
@openzeppelin/contracts/security/Pausable.sol	2
@openzeppelin/contracts/security/ReentrancyGuard.sol	3
@openzeppelin/contracts/token/ERC20/ERC20.sol	8
@openzeppelin/contracts/token/ERC20/IERC20.sol	2
@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol	1
@openzeppelin/contracts/token/ERC20/extensions/draft-ERC20Permit.sol	1
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	10
@openzeppelin/contracts/utils/Address.sol	3
@openzeppelin/contracts/utils/math/Math.sol	1
@openzeppelin/contracts/utils/math/SafeMath.sol	2

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.

v1.0

File Name	SHA-1 Hash
contracts/interfaces/zyber/IZyberChef.sol	2a61fcab47cd48a8d8342b9cf3fe2fbd0442a52f
contracts/interfaces/sushi/IMiniChefV2.sol	97d69bf12c6c63bcd6d928a7c5268b7a52aba177
contracts/interfaces/sushi/IRewarder.sol	25d0189372f1af482a7b051691c9ba42f21d76cb
contracts/interfaces/yieldgenius/IStrategyV7.sol	8831fb978b49443f74881d28b2805a59159e53a1
contracts/interfaces/common/IUniswapV2Pair.sol	6bb05b091725be6f6934ba13548c938093f0792d
contracts/interfaces/common/IUniswapRouterV3WithDeadline.sol	89e97c8b295e3d6a1cf92b882ab1cfb447495aa1
contracts/interfaces/common/IUniswapRouterETH.sol	6daf4855bde9b9964d29b401b665a4a3c4f3d442
contracts/interfaces/common/IKyberElastic.sol	c64e1d471ab6ae5acfe1afc57aed35360cd02f9c
contracts/interfaces/common/IMasterChef.sol	e591c497353eaa63472c64dadfee4249a41b3a4f
contracts/interfaces/common/IMultiRewardPool.sol	6b252a88c43e0958d9304c11c23cb78ca3282802
contracts/interfaces/common/IUniswapRouterV3.sol	a5755a09ac567ebef025f5a5a8d90a1f6e88e395

contracts/interfaces/common/IWrappedNative.sol	56f51368d3d7696baacbd152195eb12b42fa513a
contracts/interfaces/common/IFeeConfig.sol	2e38cb0a2a01e581e9913dc1cd91d87a41a13e1d
contracts/interfaces/common/ISolidlyRouter.sol	548eaa31708f769192828ca3ded0b3d22ea394aa
contracts/interfaces/common/ISolidlyPair.sol	528cae7eb78d666975e6cf935aae2cbbcee68a59
contracts/interfaces/convex/IStakedCvxCrv.sol	385a46e72d2845b21e84f8addcd5d7e5a05e4d61
contracts/interfaces/convex/IConvex.sol	cf9663e3ed0f9f326eda2215fb107a946907988c
contracts/interfaces/curve/ICurveSwap.sol	9cb1b57778ee7b7f548cb61ff8a62c3e4f2fb68f
contracts/interfaces/curve/IRewardsGauge.sol	e96037f74a0b43bbcb05876a0089f3d112af9651
contracts/interfaces/curve/IHelper.sol	1bac227da5b99d9575f2c176a91f4196c5ba6e6f
contracts/interfaces/curve/IGaugeFactory.sol	3333402e1eb405add6606761ee10ea425d2e19e0
contracts/interfaces/curve/ICurveRouter.sol	9098b201d914d7451489cac7dcd32877318e6e2e
contracts/interfaces/curve/IStreamer.sol	795bce35606726da35541002703acf08bc65dbc7
contracts/interfaces/gmx/IBeefyVault.sol	b0d3a5da41672092de9c8e93b52ce7a93f5f28c2
contracts/interfaces/gmx/IGMXTracker.sol	f195279927e04c4021be9c97aaea741acb5be49d
contracts/interfaces/gmx/IGMXRouter.sol	581527da28fd6fa6c25768ba6f3a1ed3bfc268fc

contracts/interfaces/gmx/IGLPManager.sol	3441b7bee16144a8ea008e035515f0390cd97a3d
contracts/interfaces/gmx/IGMXVault.sol	2906a3ae98fabcbc22d4db52455dbe9bff2c609a
contracts/interfaces/gmx/IFeeStakedOLP.sol	581f8ab9809085b98e5dfab0d1fbbcc4c688a38c
contracts/interfaces/gmx/IGMXStrategy.sol	5ba3fca94640b16d58752765dd7db896e35441ca
contracts/distributor/IEscrowMaster.sol	29b368815066913bc47d6ac23b62710056374bfb
contracts/distributor/rewarders/MultipleRewards.sol	a36786914727174abf2a27073cbd0543857d25e9
contracts/distributor/rewarders/IMultipleRewards.sol	9b2f1ce8cb4052f53a8490f942935bf1e3987a27
contracts/distributor/EscrowMaster.sol	4023de29ea429b6050181544f18ede97e07965a0
contracts/distributor/libraries/IBoringERC20.sol	042423caee824b0904a82a4c30ea24136f697a43
contracts/distributor/libraries/BoringERC20.sol	edffbd6a0afe166e64494d1c97e620f93dc52867
contracts/distributor/IYieldGeniusDistributor.sol	9158e14c9a3f83b956d618bb5f2dc144d250e6a5
contracts/distributor/YieldGeniusDistributor.sol	3c6c5e841179136d76428e4fd7ccf34b474e359d
contracts/distributor/IZyberPair.sol	48677fd53e7adb2571ddbf79c800d52e2cecfce6
contracts/vaults/YieldGeniusVaultFactory.sol	b7f50334e22a0a535cfecaf480d25dbc0a783373
contracts/vaults/YieldGeniusVault.sol	79af9d876658b7ef1b4a2dc8c3bfa2508e00004c

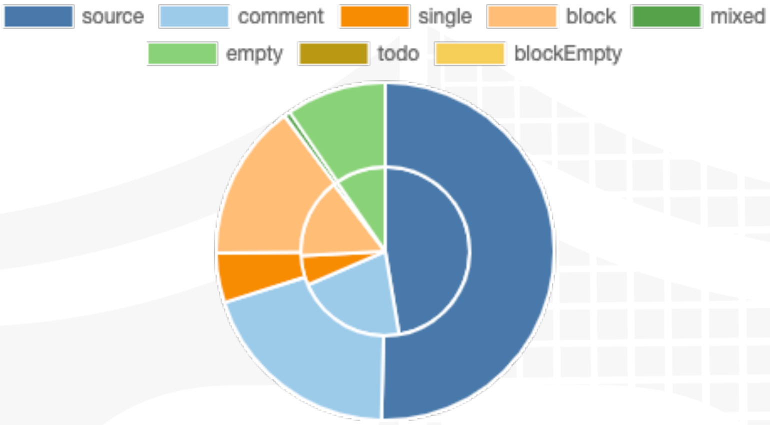
contracts/TimeLock.sol	09143d7b480b32164991adf654c23924648a28db
contracts/utils/BytesLib.sol	c699f3f2470e099cc677c81d60cbd1ea15f365eb
contracts/utils/GasThrottler.sol	e15fe7839e667526659ac1a5d7aa3c1893b7fa9e
contracts/utils/IGasPrice.sol	df056fa77ef39cda19ab34c79e62e52032ec96f2
contracts/utils/StringUtils.sol	bed83b5bc1507201b0f8a40de26b3390cdaef20d
contracts/utils/FeeConfigurator.sol	e56bb4fb630ab6b9a0b8eaea2e95d0042e066728
contracts/utils/Path.sol	7da088871d5dc49469bf08671dabdea3e085a714
contracts/utils/UniV3Actions.sol	65ff912dc2022e594408737da5d8c7d5d9518fd5
contracts/utils/GasPrice.sol	229fc2c5ad0b21fd86ed6757ba6e0960c75e3ff6
contracts/zaps/YieldGeniusZapOneInch.sol	09750767745066881d3c250b05b578720d5f631f
contracts/zaps/YieldGeniusUniswapv2Zap.sol	a8e13e3bfb2f56bc1bd1d9d5d0d3ad5a62e972a5
contracts/zaps/zapInterfaces/IYieldGeniusVault.sol	8475a6fcc7dcbb4362d4d915b9ccfe6f3f8c6688
contracts/zaps/zapInterfaces/IStrategy.sol	9322b0e358e7c729df7dd66bdf07593b1c601434
contracts/zaps/zapInterfaces/IWETH.sol	fbe18f8946c0b5388b0e959a0896e73cba23c3a3
contracts/zaps/zapInterfaces/IERC20Extended.sol	cc22d3c0b06c3b0fdaaa846f681bd01eac27a877

contracts/zaps/zapInterfaces/IBeefyDataSource.sol	93cf467d5c3e01b0901435765b40f0933b9de02d
contracts/YieldGeniusToken.sol	16fc682fe31d237b4a51974beba77552f464653f
contracts/strategies/zyber/StrategyZyberMultiRewardsLP.sol	948120f2d0719882df28ef7be482ff597b396a76
contracts/strategies/sushi/StrategyArbSushiDualLP.sol	c348762f44a6298f7c272b45ab1926512fef0d52
contracts/strategies/common/StratFeeManager.sol	39d129edbc4522b513dac5f8f6c4715eb831ba48
contracts/strategies/common/StrategyCommonChefLP.sol	f400dce221d918c8ce77a03d4ff46b6354151774
contracts/strategies/common/StrategyCommonMultiRewardPoolLP.sol	a9f0789df2b9b427d2648cbd2e0b8be7ae96dcfa
contracts/strategies/common/FeeManager.sol	fe0446395b8bcf985e47e0e3df495a77d1e601bf
contracts/strategies/common/StratManager.sol	c86b28d73af723386633247f753e52864cd9c9cf
contracts/strategies/gmx/StrategyGLP.sol	d5b12c097cae7665802d7436a5fa9573fb57a8e6
contracts/strategies/curve/StrategyCurveLPUniV3Router.sol	0926d66c7d70fdf32b0cdf6b004969694c1c41b1
contracts/strategies/curve/StrategyConvexL2.sol	574512c113648f11c3b95cde943b56e2b74bf1d2

Metrics

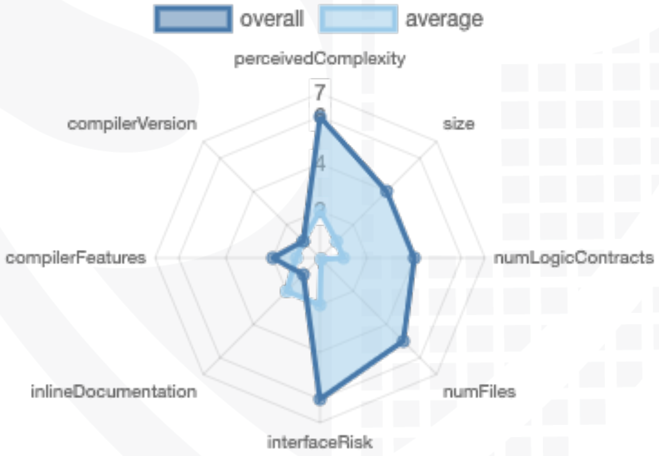
Source Lines

v1.0



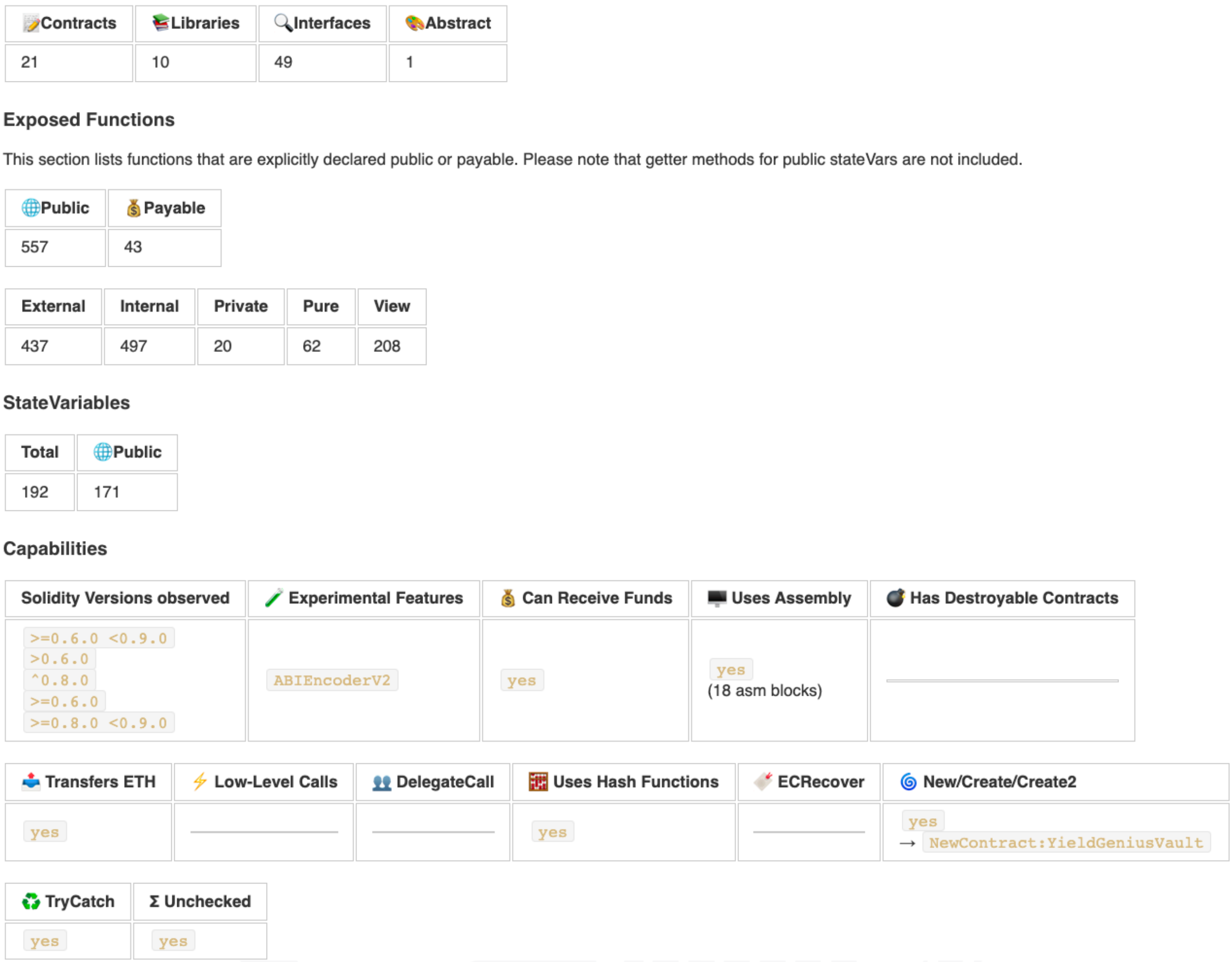
Risk Level

v1.0



Capabilities

Components



📝Contracts	📚Libraries	🔍Interfaces	🎨Abstract
21	10	49	1

Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

🌐Public	💰Payable
557	43

External	Internal	Private	Pure	View
437	497	20	62	208

StateVariables

Total	🌐Public
192	171

Capabilities

Solidity Versions observed	🧪 Experimental Features	💰 Can Receive Funds	🖥 Uses Assembly	💣 Has Destroyable Contracts

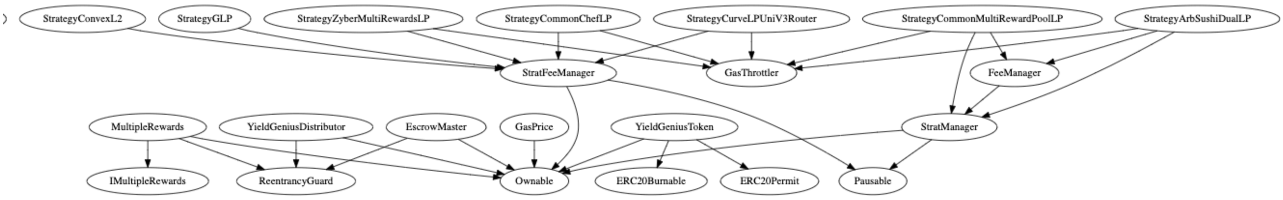
`>=0.6.0 <0.9.0`
`>0.6.0`
`^0.8.0`
`>=0.6.0`
`>=0.8.0 <0.9.0`
	`ABIEncoderV2`	`yes`	`yes`
			(18 asm blocks)	

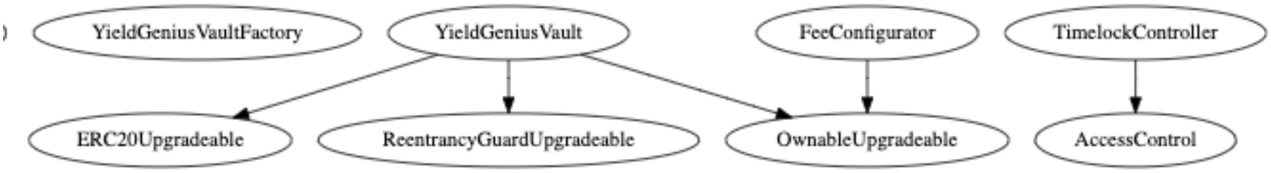
📤 Transfers ETH	⚡ Low-Level Calls	👥 DelegateCall	🧮 Uses Hash Functions	🔖 ECRecover	🌀 New/Create/Create2
`yes`			`yes`		`yes` → `NewContract:YieldGeniusVault`

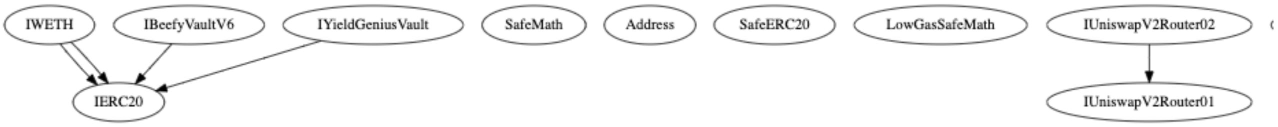
♻️ TryCatch	Σ Unchecked
`yes`	`yes`

Inheritance Graph

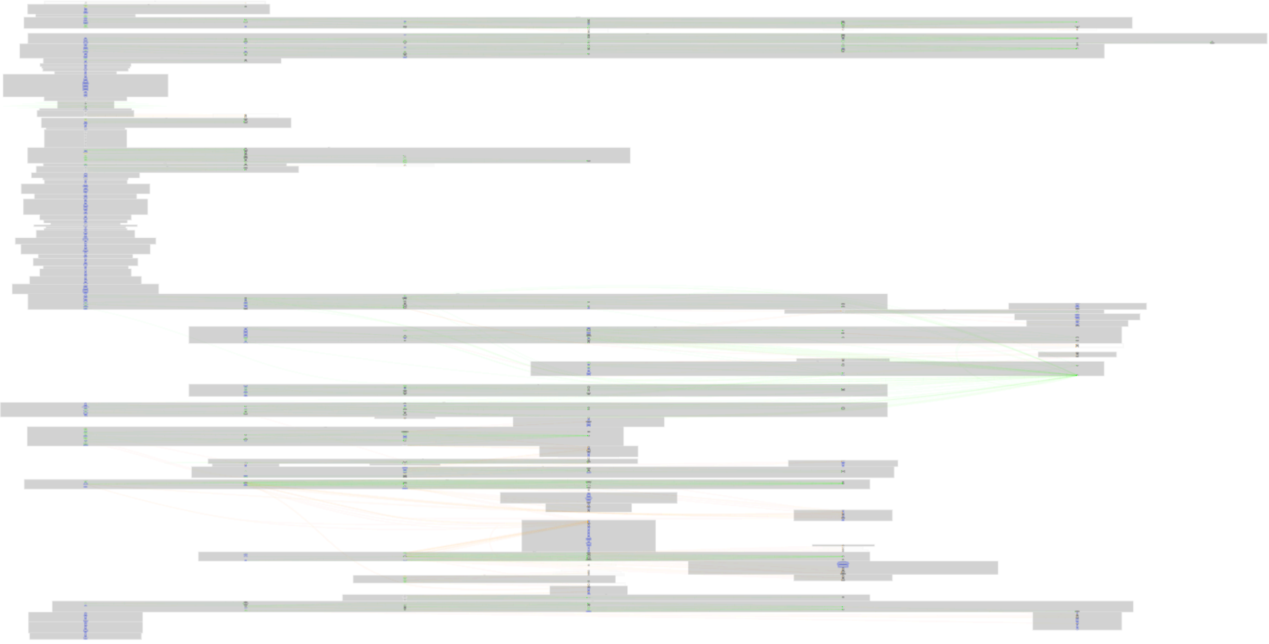
v1.0







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)

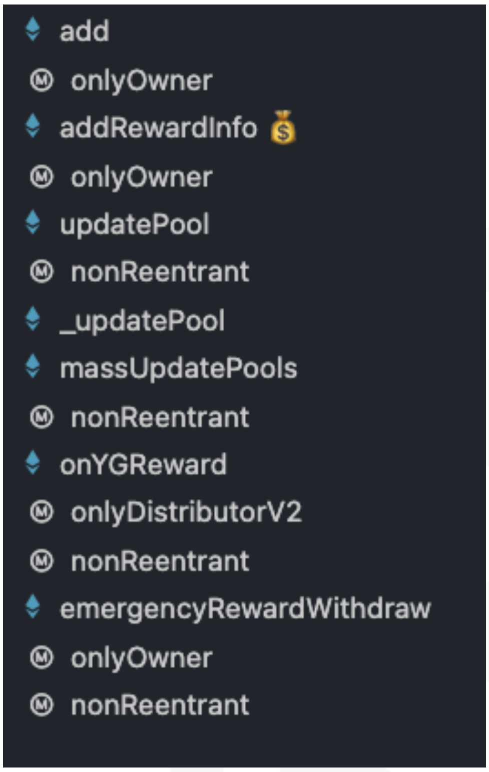
Tested	Verified
✓	✓

Legend

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

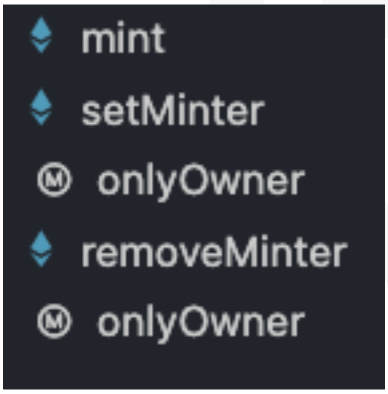
Modifiers and public functions
v1.1

MultipleRewards.sol



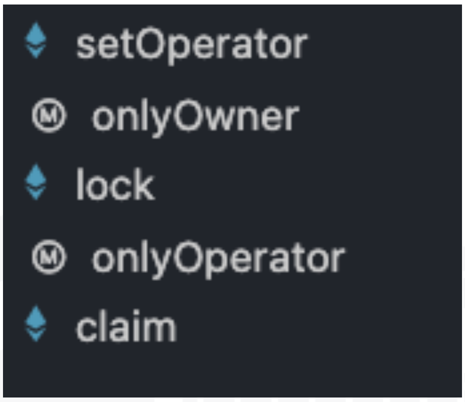
♦ add
Ⓜ onlyOwner
♦ addRewardInfo 💰
Ⓜ onlyOwner
♦ updatePool
Ⓜ nonReentrant
♦ _updatePool
♦ massUpdatePools
Ⓜ nonReentrant
♦ onYGReward
Ⓜ onlyDistributorV2
Ⓜ nonReentrant
♦ emergencyRewardWithdraw
Ⓜ onlyOwner
Ⓜ nonReentrant

YieldGeniusToken.sol



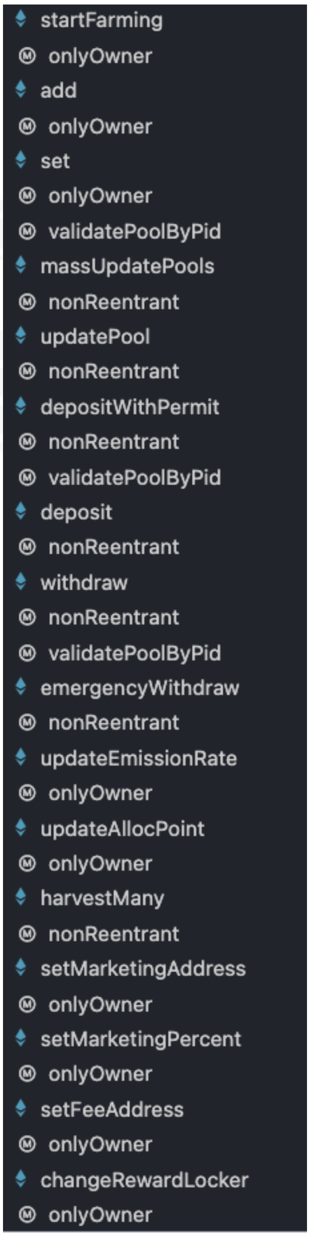
♦ mint
♦ setMinter
Ⓜ onlyOwner
♦ removeMinter
Ⓜ onlyOwner

EscrowMaster.sol



♦ setOperator
Ⓜ onlyOwner
♦ lock
Ⓜ onlyOperator
♦ claim

YieldGeniusDistributor.sol



♦ startFarming
Ⓜ onlyOwner
♦ add
Ⓜ onlyOwner
♦ set
Ⓜ onlyOwner
Ⓜ validatePoolByPid
♦ massUpdatePools
Ⓜ nonReentrant
♦ updatePool
Ⓜ nonReentrant
♦ depositWithPermit
Ⓜ nonReentrant
Ⓜ validatePoolByPid
♦ deposit
Ⓜ nonReentrant
♦ withdraw
Ⓜ nonReentrant
Ⓜ validatePoolByPid
♦ emergencyWithdraw
Ⓜ nonReentrant
♦ updateEmissionRate
Ⓜ onlyOwner
♦ updateAllocPoint
Ⓜ onlyOwner
♦ harvestMany
Ⓜ nonReentrant
♦ setMarketingAddress
Ⓜ onlyOwner
♦ setMarketingPercent
Ⓜ onlyOwner
♦ setFeeAddress
Ⓜ onlyOwner
♦ changeRewardLocker
Ⓜ onlyOwner

Note:

- General fork from Beefy Finance
- BIFI
    - Folders inside are the same as the BIFI directory
        - https://github.com/beefyfinance/beefy-contracts/tree/master/contracts/BIFI
        - Differences are changed pragma versions

Ownership Privileges

- The owner/manager can pause the strategy contracts
- *MultipleRewards.so*l -
    - Add new pool, and reward info
- *EscrowMaster.so*l -
    - Set operator address and the operator address can lock tokens
- *YieldGeniusDistributor.sol* -
    - Start Farming, add new lp to the pool
    - Set allocation point, and harvest interval for a given pool but not more than the maximum values
    - Update allocation points for a given '*_pid*' without to any arbitrary values.
    - Change Reward locker address which may result in changes in the reward system
    - The owner is also able to set marketing fee percentage up to 10%

Alleviation

- *YieldGeniusToken.sol* -
    - The owner can set minter addresses and those addresses can mint tokens without any restrictions in the "*YieldGeniusToken*" contract
        - Be aware of this

YieldGenius Team's response- "*The minting will be protected by a timelock or even renounced further on. Please modify the text to include timelock protection.*"

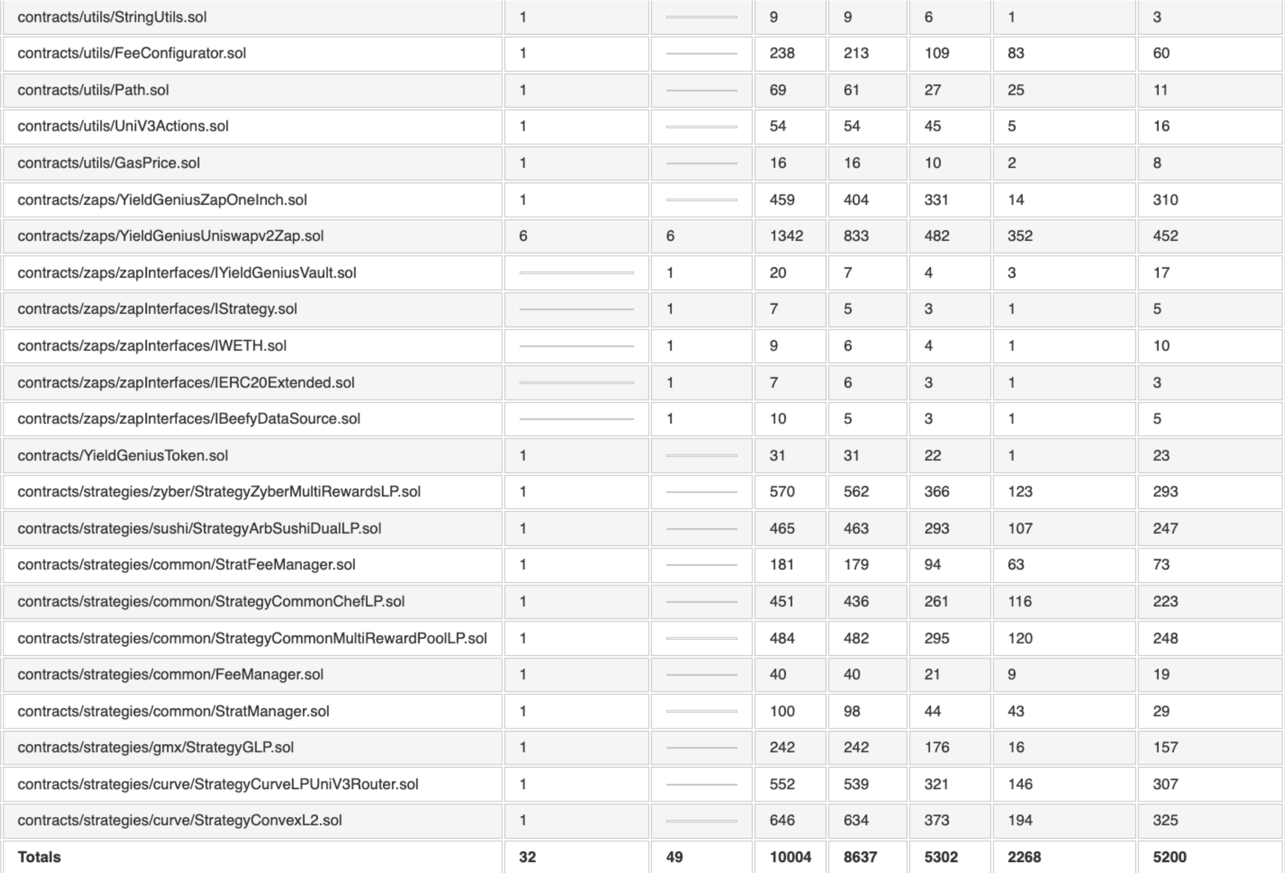
According to the YieldGenius team, time lock protection will prevent unnecessary minting of tokens by the addresses with the minter role.

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/interfaces/zyber/IZyberChef.sol		1	30	7	4	1	11
contracts/interfaces/sushi/IMiniChefV2.sol		1	15	6	3	1	19
contracts/interfaces/sushi/IRewarder.sol		1	14	6	3	1	17
contracts/interfaces/yieldgenius/IStrategyV7.sol		1	23	8	4	1	31
contracts/interfaces/common/IUniswapV2Pair.sol		1	13	6	3	1	15
contracts/interfaces/common/IUniswapRouterV3WithDeadline.sol		1	64	57	38	13	21
contracts/interfaces/common/IUniswapRouterETH.sol		1	63	6	3	1	23
contracts/interfaces/common/IKyberElastic.sol		1	63	56	37	13	21
contracts/interfaces/common/IMasterChef.sol		1	12	6	3	1	13
contracts/interfaces/common/IMultiRewardPool.sol		1	17	6	3	1	13
contracts/interfaces/common/IUniswapRouterV3.sol		1	59	52	33	13	21
contracts/interfaces/common/IWrappedNative.sol		1	9	6	3	1	8
contracts/interfaces/common/IFeeConfig.sol		1	22	19	16	1	7
contracts/interfaces/common/ISolidlyRouter.sol		1	86	14	9	2	22
contracts/interfaces/common/ISolidlyPair.sol		1	13	6	3	1	15
contracts/interfaces/convex/IStakedCvxCrv.sol		1	12	6	3	1	13
contracts/interfaces/convex/IConvex.sol		3	42	8	5	7	33
contracts/interfaces/curve/ICurveSwap.sol		1	28	6	3	1	63
contracts/interfaces/curve/IRewardsGauge.sol		1	12	6	3	1	13
contracts/interfaces/curve/IHelper.sol		1	7	6	3	1	3
contracts/interfaces/curve/IGaugeFactory.sol		1	7	6	3	1	3
contracts/interfaces/curve/ICurveRouter.sol		1	13	7	3	1	3
contracts/interfaces/curve/IStreamer.sol		1	7	6	3	1	3
contracts/interfaces/gmx/IBeefyVault.sol		1	12	10	7	1	5
contracts/interfaces/gmx/IGMXTracker.sol		1	10	6	3	1	9
contracts/interfaces/gmx/IGMXRouter.sol		1	29	6	3	1	27
contracts/interfaces/gmx/IGLPManager.sol		1	10	6	3	1	9
contracts/interfaces/gmx/IGMXVault.sol		1	24	6	3	1	17
contracts/interfaces/gmx/IFeeStakedOLP.sol		1	9	6	3	1	7
contracts/interfaces/gmx/IGMXStrategy.sol		1	7	6	3	1	3
contracts/distributor/IEscrowMaster.sol		1	33	11	5	4	17
contracts/distributor/rewarders/MultipleRewards.sol	1		501	464	337	58	154
contracts/distributor/rewarders/IMultipleRewards.sol		1	21	7	4	1	9
contracts/distributor/EscrowMaster.sol	1		152	148	127	3	70
contracts/distributor/libraries/IBoringERC20.sol		1	35	5	3	2	13
contracts/distributor/libraries/BoringERC20.sol	1		107	92	62	27	52
contracts/distributor/IYieldGeniusDistributor.sol		1	24	12	9	5	11
contracts/distributor/YieldGeniusDistributor.sol	1		1000	935	572	223	386
contracts/distributor/IZyberPair.sol		1	16	5	3	1	5
contracts/vaults/YieldGeniusVaultFactory.sol	1		39	39	23	8	25
contracts/vaults/YieldGeniusVault.sol	1		219	214	119	67	119
contracts/TimeLock.sol	1		409	355	191	128	142
contracts/utils/BytesLib.sol	1		684	638	318	239	872
contracts/utils/GasThrottler.sol	1		23	23	17	1	10
contracts/utils/IGasPrice.sol		1	7	6	3	1	3
contracts/utils/StringUtils.sol	1		9	9	6	1	3
contracts/utils/FeeConfigurator.sol	1		238	213	109	83	60
contracts/utils/Path.sol	1		69	61	27	25	11
contracts/utils/UniV3Actions.sol	1		54	54	45	5	16
contracts/utils/GasPrice.sol	1		16	16	10	2	8



contracts/utils/StringUtils.sol	1		9	9	6	1	3
contracts/utils/FeeConfigurator.sol	1		238	213	109	83	60
contracts/utils/Path.sol	1		69	61	27	25	11
contracts/utils/UniV3Actions.sol	1		54	54	45	5	16
contracts/utils/GasPrice.sol	1		16	16	10	2	8
contracts/zaps/YieldGeniusZapOneInch.sol	1		459	404	331	14	310
contracts/zaps/YieldGeniusUniswapv2Zap.sol	6	6	1342	833	482	352	452
contracts/zaps/zapInterfaces/IYieldGeniusVault.sol		1	20	7	4	3	17
contracts/zaps/zapInterfaces/IStrategy.sol		1	7	5	3	1	5
contracts/zaps/zapInterfaces/IWETH.sol		1	9	6	4	1	10
contracts/zaps/zapInterfaces/IERC20Extended.sol		1	7	6	3	1	3
contracts/zaps/zapInterfaces/IBeefyDataSource.sol		1	10	5	3	1	5
contracts/YieldGeniusToken.sol	1		31	31	22	1	23
contracts/strategies/zyber/StrategyZyberMultiRewardsLP.sol	1		570	562	366	123	293
contracts/strategies/sushi/StrategyArbSushiDualLP.sol	1		465	463	293	107	247
contracts/strategies/common/StratFeeManager.sol	1		181	179	94	63	73
contracts/strategies/common/StrategyCommonChefLP.sol	1		451	436	261	116	223
contracts/strategies/common/StrategyCommonMultiRewardPoolLP.sol	1		484	482	295	120	248
contracts/strategies/common/FeeManager.sol	1		40	40	21	9	19
contracts/strategies/common/StratManager.sol	1		100	98	44	43	29
contracts/strategies/gmx/StrategyGLP.sol	1		242	242	176	16	157
contracts/strategies/curve/StrategyCurveLPUniV3Router.sol	1		552	539	321	146	307
contracts/strategies/curve/StrategyConvexL2.sol	1		646	634	373	194	325
Totals	**32**	**49**	**10004**	**8637**	**5302**	**2268**	**5200**

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	EscrowMaster.sol	Missing Zero Address Validation (missing-zero-check)	55	Check that the address is not zero
#3	MultipleRewards.sol	Missing Zero Address Validation (missing-zero-check)	481	Check that the address is not zero otherwise the amount will be lost
#4	YieldGeniusToken.sol	Missing Zero Address Validation (missing-zero-check)	24, 28	Check that the address is not zero otherwise the amount will be lost
#5	YieldGeniusDistributor.sol	Missing Events Arithmetic	237, 894, 911, 997	Emit an event for critical parameter changes
#6	YieldGeniusToken.sol	Missing Events Arithmetic	24, 28	Emit an event for critical parameter changes

#7	Escrow Master.sol	Missing Events Arithmetic	All	Emit an event for critical parameter changes

Informational issues

Issue	File	Type	Line	Description
#1	Escrow Master.sol	Uninitialised Local Variables	59, 115	Make sure to initialise all local variables
#2	YieldGeniusToken.sol	Missing Inheritence	9	The contract should inherit from IBoringER20
#3	BoringERC20.sol	Dead Code	49	The function is never used and should be removed.

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.

21. March 2023:

- Owner can deploy a new version of the contract which can change any limit and give owner new privileges
- This project consists of the following forks
  - Beefy
  - Marsecosystem
- Read whole report and modifiers section for more information
- The low issues that exist in the Beefy finance codebase still exist in the forked code.
- 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	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

SolidProof_io @solidproof_io

Solid Proofed

Blockchain Security | Smart Contract Audits | KYC
Development | Marketing

MADE IN GERMANY