SOLIDProof

Bring trust into your projects

Blockchain Security | Smart Contract Audits | KYC
Development | Marketing

MADE IN GERMANY

Zyberswap

Audit

Security Assessment
21. January, 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	19. January 2023 - 21. January 2023	• Layout project • Automated- /Manual-Security Testing • Summary
	21. January 2023	• Finishing report

Network

Ethereum (ERC20)
Binance Smart Chain (BEP20)

Website

https://www.zyberswap.io/

Telegram

https://t.me/zyberswap

Twitter

https://twitter.com/zyberswap

Discord

https://discord.gg/NZ2S3ZEYFj

Description

Zyberswap is aiming to become one of the first decentralized exchanges (DEX) with an automated market-maker (AMM) on the Arbitrum blockchain. Compared to its competitors, Zyberswap will allow the swapping of crypto assets with **the lowest fees!** Rewards from Staking and Yield Farming will be among **the most lucrative** in the entire Arbitrum ecosystem. Additionally, Zyberswap aims to fully involve its users in decision-making. All major changes will be decided via **Governance Voting!**

Project Engagement

During the 19th of January 2023, **ZyberSwap 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

- Provided as files

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.

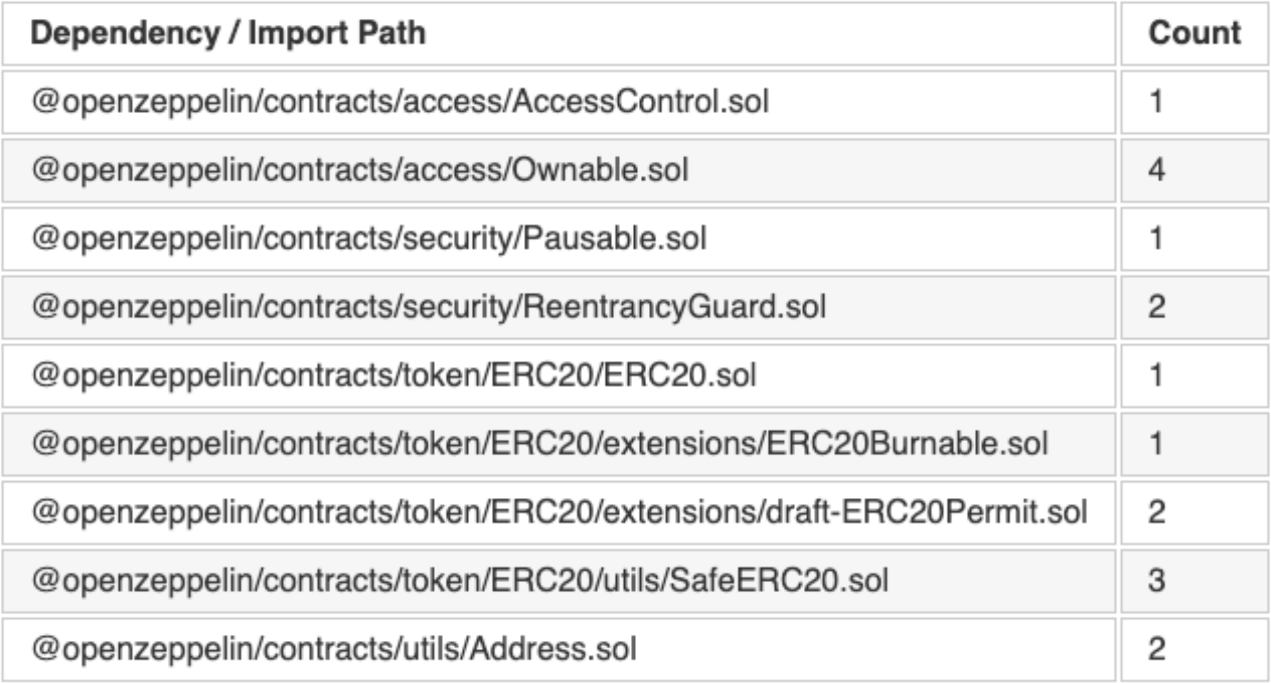
2. Testing and automated analysis that includes the following:
    i) Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    ii) Symbolic execution, which is analysing a program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



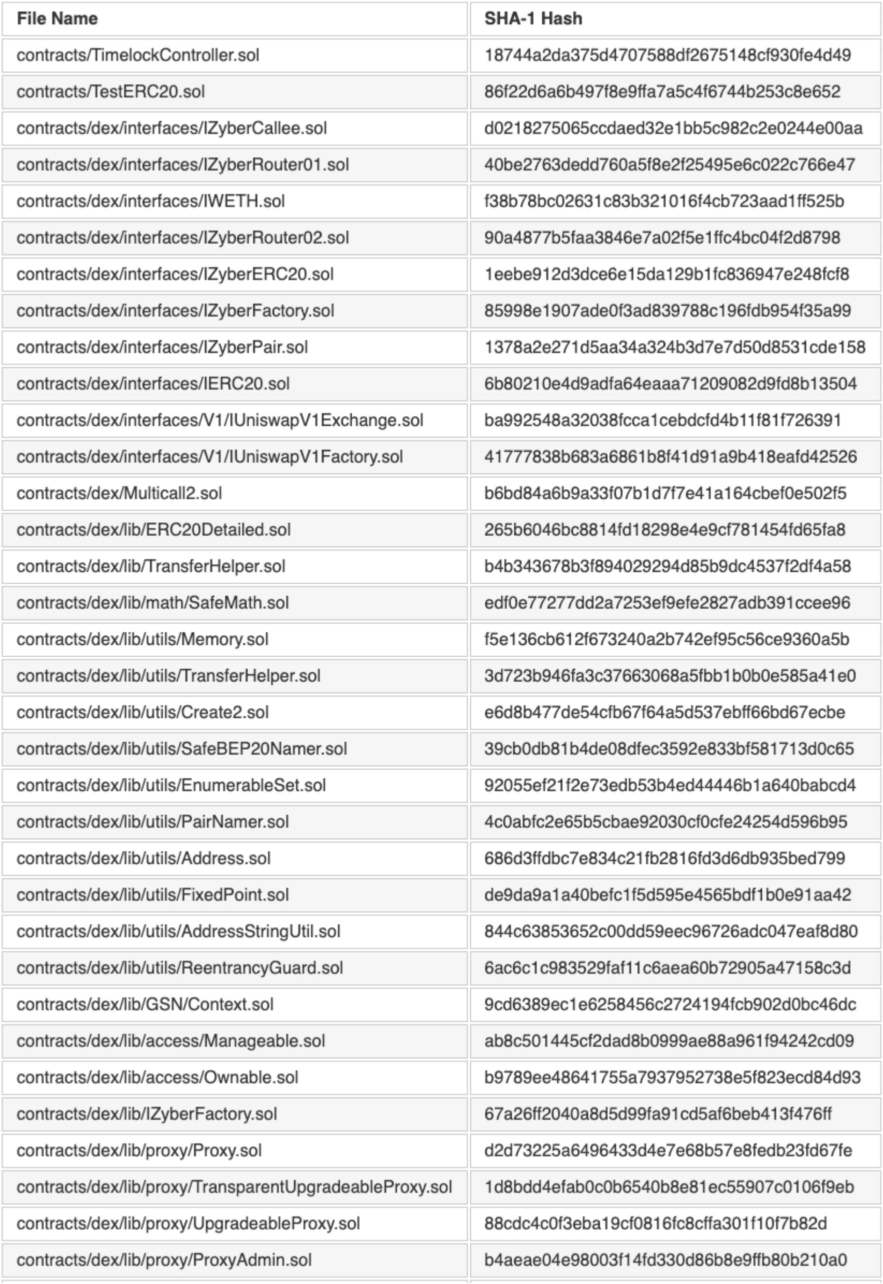
Dependency / Import Path	Count
@openzeppelin/contracts/access/AccessControl.sol	1
@openzeppelin/contracts/access/Ownable.sol	4
@openzeppelin/contracts/security/Pausable.sol	1
@openzeppelin/contracts/security/ReentrancyGuard.sol	2
@openzeppelin/contracts/token/ERC20/ERC20.sol	1
@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol	1
@openzeppelin/contracts/token/ERC20/extensions/draft-ERC20Permit.sol	2
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	3
@openzeppelin/contracts/utils/Address.sol	2

Tested Contract Files

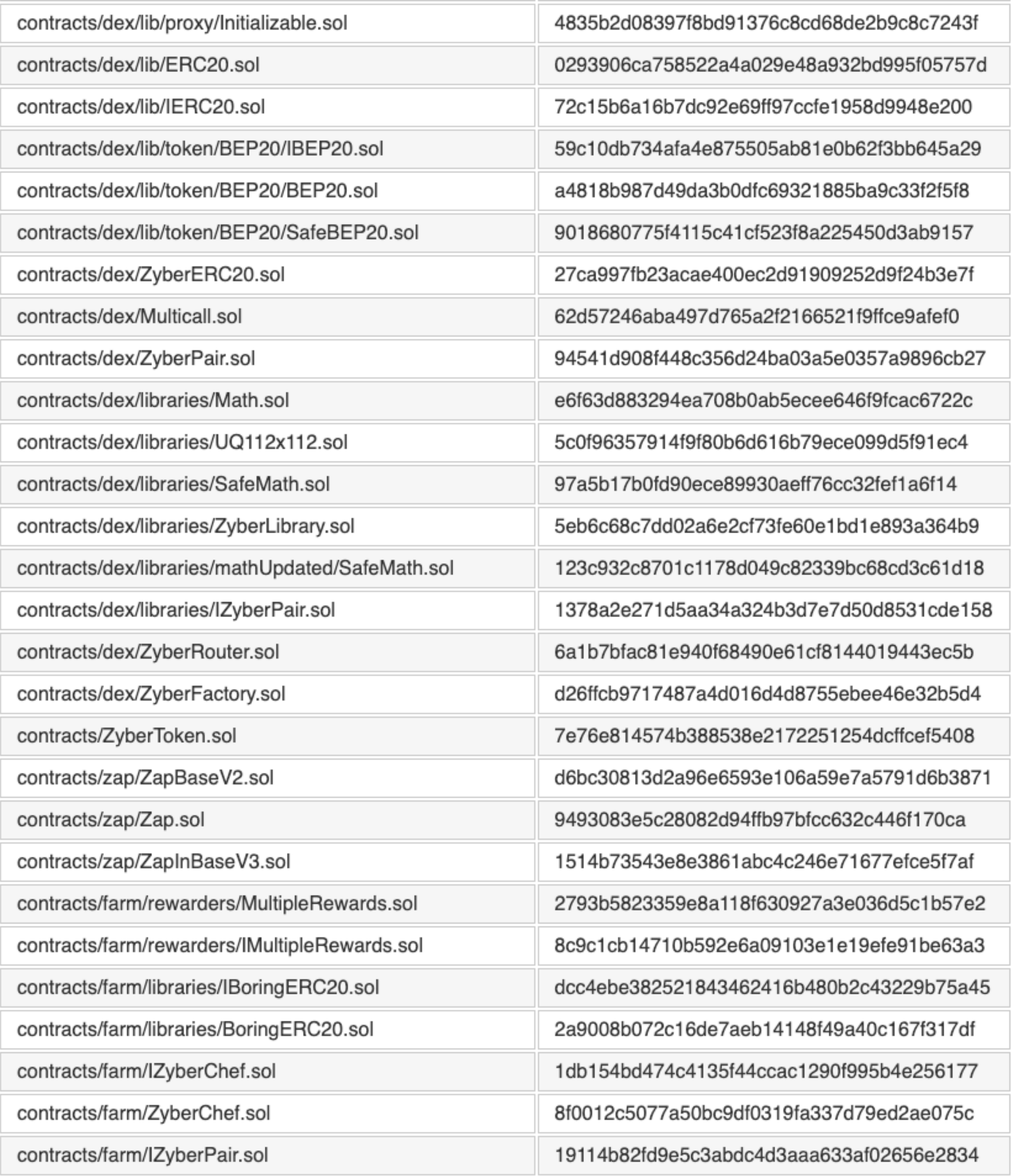
This audit covered the following files listed below with a SHA-1 Hash.

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/TimelockController.sol	18744a2da375d4707588df2675148cf930fe4d49
contracts/TestERC20.sol	86f22d6a6b497f8e9ffa7a5c4f6744b253c8e652
contracts/dex/interfaces/IZyberCallee.sol	d0218275065ccdaed32e1bb5c982c2e0244e00aa
contracts/dex/interfaces/IZyberRouter01.sol	40be2763dedd760a5f8e2f25495e6c022c766e47
contracts/dex/interfaces/IWETH.sol	f38b78bc02631c83b321016f4cb723aad1ff525b
contracts/dex/interfaces/IZyberRouter02.sol	90a4877b5faa3846e7a02f5e1ffc4bc04f2d8798
contracts/dex/interfaces/IZyberERC20.sol	1eebe912d3dce6e15da129b1fc836947e248fcf8
contracts/dex/interfaces/IZyberFactory.sol	85998e1907ade0f3ad839788c196fdb954f35a99
contracts/dex/interfaces/IZyberPair.sol	1378a2e271d5aa34a324b3d7e7d50d8531cde158
contracts/dex/interfaces/IERC20.sol	6b80210e4d9adfa64eaaa71209082d9fd8b13504
contracts/dex/interfaces/V1/IUniswapV1Exchange.sol	ba992548a32038fcca1cebdcfd4b11f81f726391
contracts/dex/interfaces/V1/IUniswapV1Factory.sol	41777838b683a6861b8f41d91a9b418eafd42526
contracts/dex/Multicall2.sol	b6bd84a6b9a33f07b1d7f7e41a164cbef0e502f5
contracts/dex/lib/ERC20Detailed.sol	265b6046bc8814fd18298e4e9cf781454fd65fa8
contracts/dex/lib/TransferHelper.sol	b4b343678b3f894029294d85b9dc4537f2df4a58
contracts/dex/lib/math/SafeMath.sol	edf0e77277dd2a7253ef9efe2827adb391ccee96
contracts/dex/lib/utils/Memory.sol	f5e136cb612f673240a2b742ef95c56ce9360a5b
contracts/dex/lib/utils/TransferHelper.sol	3d723b946fa3c37663068a5fbb1b0b0e585a41e0
contracts/dex/lib/utils/Create2.sol	e6d8b477de54cfb67f64a5d537ebff66bd67ecbe
contracts/dex/lib/utils/SafeBEP20Namer.sol	39cb0db81b4de08dfec3592e833bf581713d0c65
contracts/dex/lib/utils/EnumerableSet.sol	92055ef21f2e73edb53b4ed44446b1a640babcd4
contracts/dex/lib/utils/PairNamer.sol	4c0abfc2e65b5cbae92030cf0cfe24254d596b95
contracts/dex/lib/utils/Address.sol	686d3ffdbc7e834c21fb2816fd3d6db935bed799
contracts/dex/lib/utils/FixedPoint.sol	de9da9a1a40befc1f5d595e4565bdf1b0e91aa42
contracts/dex/lib/utils/AddressStringUtil.sol	844c63853652c00dd59eec96726adc047eaf8d80
contracts/dex/lib/utils/ReentrancyGuard.sol	6ac6c1c983529faf11c6aea60b72905a47158c3d
contracts/dex/lib/GSN/Context.sol	9cd6389ec1e6258456c2724194fcb902d0bc46dc
contracts/dex/lib/access/Manageable.sol	ab8c501445cf2dad8b0999ae88a961f94242cd09
contracts/dex/lib/access/Ownable.sol	b9789ee48641755a7937952738e5f823ecd84d93
contracts/dex/lib/IZyberFactory.sol	67a26ff2040a8d5d99fa91cd5af6beb413f476ff
contracts/dex/lib/proxy/Proxy.sol	d2d73225a6496433d4e7e68b57e8fedb23fd67fe
contracts/dex/lib/proxy/TransparentUpgradeableProxy.sol	1d8bdd4efab0c0b6540b8e81ec55907c0106f9eb
contracts/dex/lib/proxy/UpgradeableProxy.sol	88cdc4c0f3eba19cf0816fc8cffa301f10f7b82d
contracts/dex/lib/proxy/ProxyAdmin.sol	b4aeae04e98003f14fd330d86b8e9ffb80b210a0

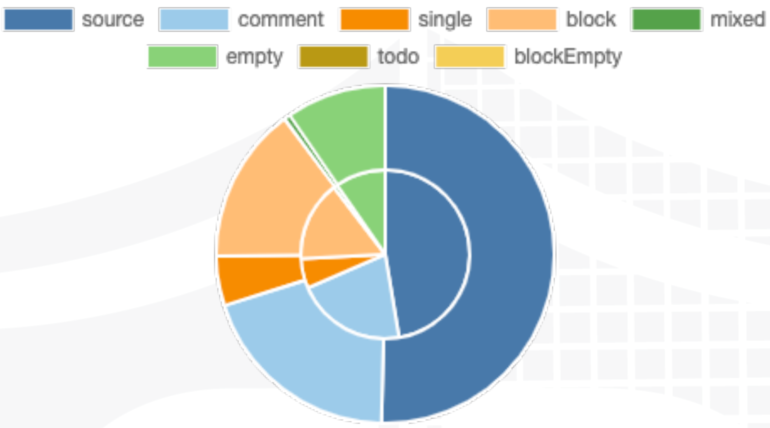


contracts/dex/lib/proxy/Initializable.sol	4835b2d08397f8bd91376c8cd68de2b9c8c7243f
contracts/dex/lib/ERC20.sol	0293906ca758522a4a029e48a932bd995f05757d
contracts/dex/lib/IERC20.sol	72c15b6a16b7dc92e69ff97ccfe1958d9948e200
contracts/dex/lib/token/BEP20/IBEP20.sol	59c10db734afa4e875505ab81e0b62f3bb645a29
contracts/dex/lib/token/BEP20/BEP20.sol	a4818b987d49da3b0dfc69321885ba9c33f2f5f8
contracts/dex/lib/token/BEP20/SafeBEP20.sol	9018680775f4115c41cf523f8a225450d3ab9157
contracts/dex/ZyberERC20.sol	27ca997fb23acae400ec2d91909252d9f24b3e7f
contracts/dex/Multicall.sol	62d57246aba497d765a2f2166521f9ffce9afef0
contracts/dex/ZyberPair.sol	94541d908f448c356d24ba03a5e0357a9896cb27
contracts/dex/libraries/Math.sol	e6f63d883294ea708b0ab5ecee646f9fcac6722c
contracts/dex/libraries/UQ112x112.sol	5c0f96357914f9f80b6d616b79ece099d5f91ec4
contracts/dex/libraries/SafeMath.sol	97a5b17b0fd90ece89930aeff76cc32fef1a6f14
contracts/dex/libraries/ZyberLibrary.sol	5eb6c68c7dd02a6e2cf73fe60e1bd1e893a364b9
contracts/dex/libraries/mathUpdated/SafeMath.sol	123c932c8701c1178d049c82339bc68cd3c61d18
contracts/dex/libraries/IZyberPair.sol	1378a2e271d5aa34a324b3d7e7d50d8531cde158
contracts/dex/ZyberRouter.sol	6a1b7bfac81e940f68490e61cf8144019443ec5b
contracts/dex/ZyberFactory.sol	d26ffcb9717487a4d016d4d8755ebee46e32b5d4
contracts/ZyberToken.sol	7e76e814574b388538e2172251254dcffcef5408
contracts/zap/ZapBaseV2.sol	d6bc30813d2a96e6593e106a59e7a5791d6b3871
contracts/zap/Zap.sol	9493083e5c28082d94ffb97bfcc632c446f170ca
contracts/zap/ZapInBaseV3.sol	1514b73543e8e3861abc4c246e71677efce5f7af
contracts/farm/rewarders/MultipleRewards.sol	2793b5823359e8a118f630927a3e036d5c1b57e2
contracts/farm/rewarders/IMultipleRewards.sol	8c9c1cb14710b592e6a09103e1e19efe91be63a3
contracts/farm/libraries/IBoringERC20.sol	dcc4ebe382521843462416b480b2c43229b75a45
contracts/farm/libraries/BoringERC20.sol	2a9008b072c16de7aeb14148f49a40c167f317df
contracts/farm/IZyberChef.sol	1db154bd474c4135f44ccac1290f995b4e256177
contracts/farm/ZyberChef.sol	8f0012c5077a50bc9df0319fa337d79ed2ae075c
contracts/farm/IZyberPair.sol	19114b82fd9e5c3abdc4d3aaa633af02656e2834

Metrics

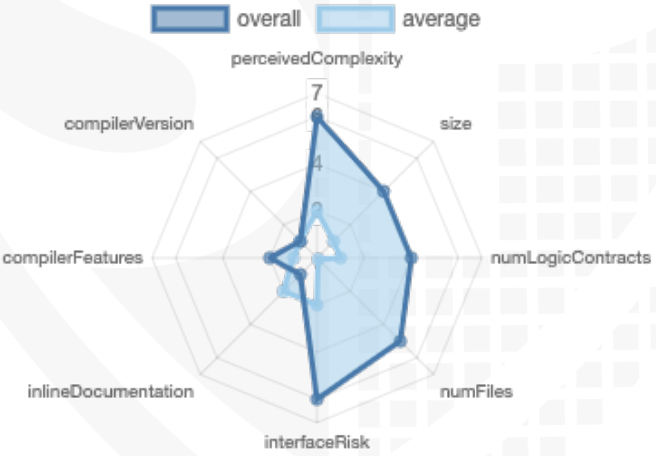
Source Lines

v1.0



Risk Level

v1.0



Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	22	20	24	7

Exposed Functions

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

Version	Public	Payable
1.0	353	24

Version	External	Internal	Private	Pure	View
1.0	235	415	27	85	167

State Variables

Version	Total	Public
1.0	117	63

Capabilities

Version	Solidity Versions observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts

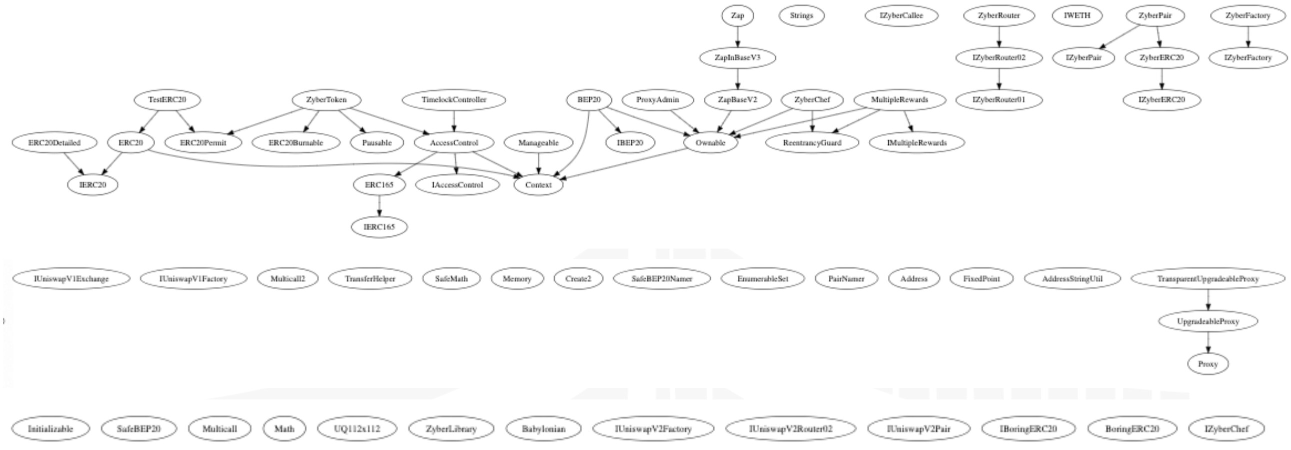
1.0	`^0.8.0`
	`0.8.16`
	`>=0.5.0`
	`>=0.6.2`
	`>=0.5.0`
	`<=0.6.12`
	`>=0.6.0`
	`^0.5.0`
	`=0.5.16`
	`=0.6.6`
		`ABIEncoderV2`	`yes`	`yes`
				(21 asm blocks)	

Version	Transfers ETH	Low-Level Calls	DelegateCall	Uses Hash Functions	ECRecover	New/Create/Create2

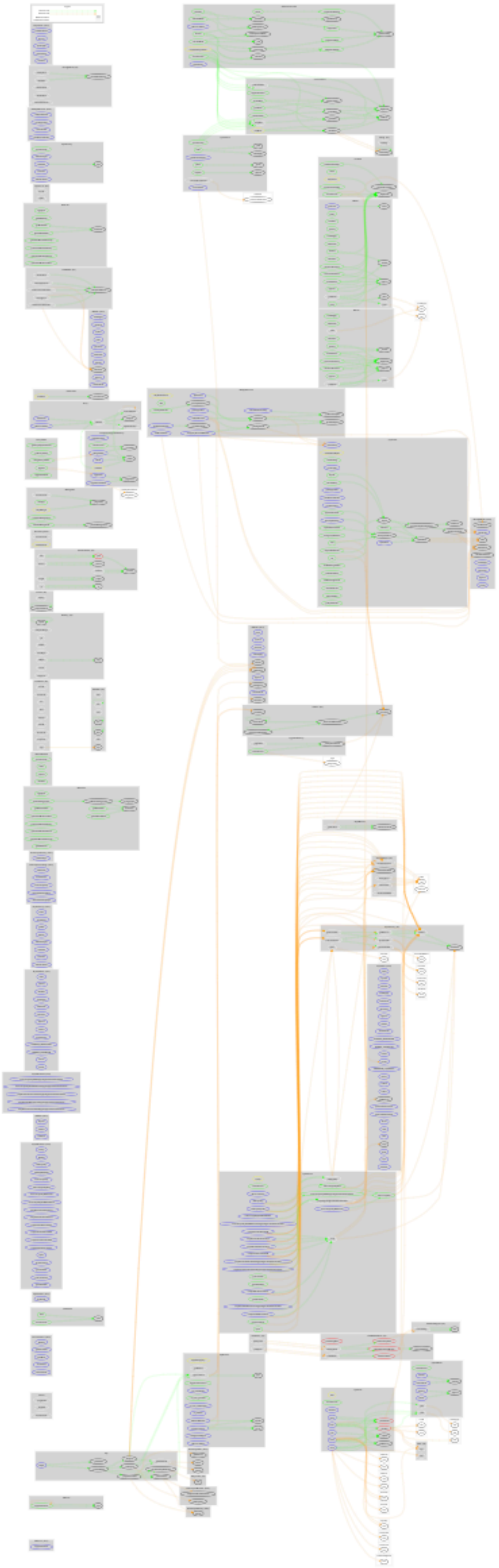
1.0	yes		yes	yes	yes	yes →AssemblyCall:Name:create2

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

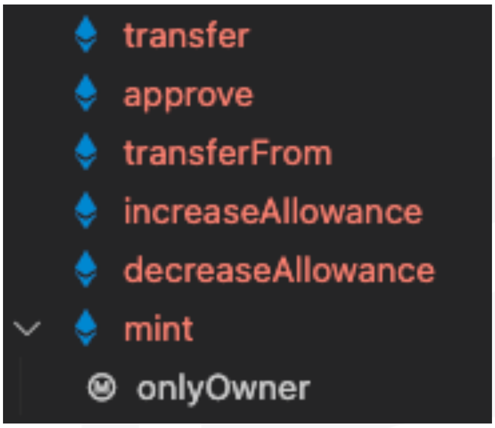
Attribute	Symbol
Verified / Checked	✓
Partly Verified	🚩
Unverified / Not checked	✗
Not available	–

Modifiers and public functions

v1.0

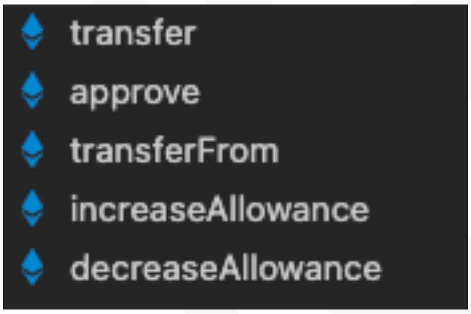
Dex

BEP20.sol



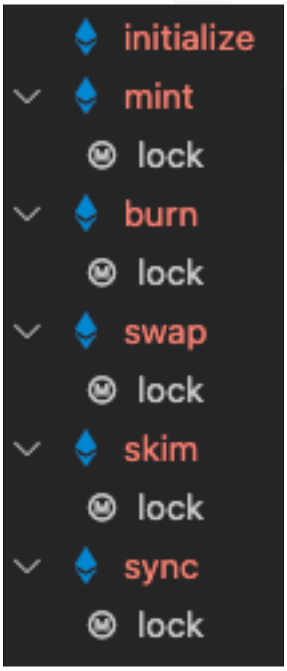
```
transfer
approve
transferFrom
increaseAllowance
decreaseAllowance
mint
  onlyOwner
```

ERC20.sol



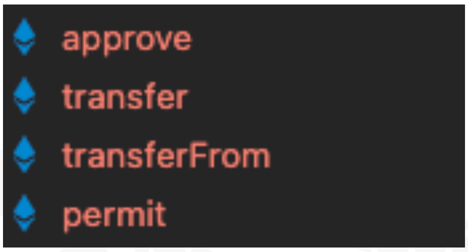
```
transfer
approve
transferFrom
increaseAllowance
decreaseAllowance
```

ZyberPair



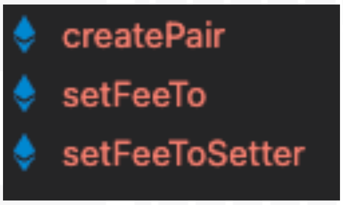
```
initialize
mint
  lock
burn
  lock
swap
  lock
skim
  lock
sync
  lock
```

ZyberERC20.sol



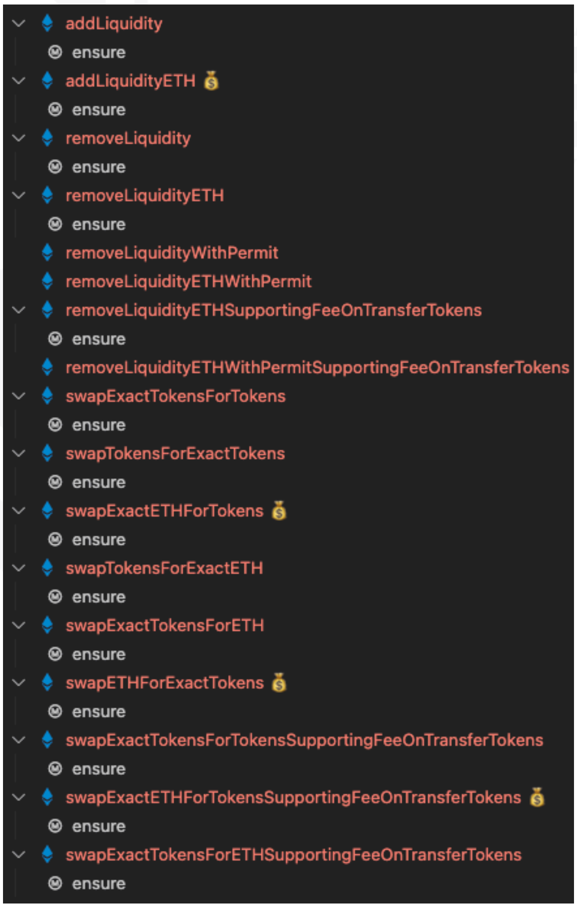
```
approve
transfer
transferFrom
permit
```

ZyberFactory.sol



```
createPair
setFeeTo
setFeeToSetter
```

ZyberRouter.sol



```
addLiquidity
  ensure
addLiquidityETH 💰
  ensure
removeLiquidity
  ensure
removeLiquidityETH
  ensure
removeLiquidityWithPermit
removeLiquidityETHWithPermit
removeLiquidityETHSupportingFeeOnTransferTokens
  ensure
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens
swapExactTokensForTokens
  ensure
swapTokensForExactTokens
  ensure
swapExactETHForTokens 💰
  ensure
swapTokensForExactETH
  ensure
swapExactTokensForETH
  ensure
swapETHForExactTokens 💰
  ensure
swapExactTokensForTokensSupportingFeeOnTransferTokens
  ensure
swapExactETHForTokensSupportingFeeOnTransferTokens 💰
  ensure
swapExactTokensForETHSupportingFeeOnTransferTokens
  ensure
```

Note:

- General fork from uniswap/pancakeswap
- Dex/lib
    - Folders inside are the same as the pancake-swap-lib
        - https://github.com/pancakeswap/pancake-swap-lib
        - Differences are changed pragma versions

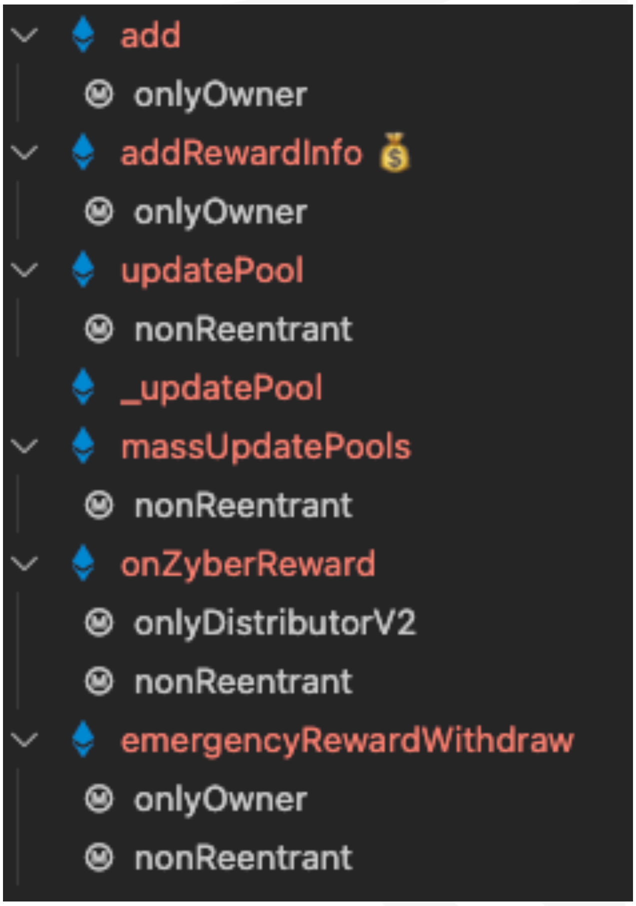
Ownership Privileges:

- The owner can mint tokens in the ‘BEP20.sol’ and ZyberPair contracts.
- The “feeToSetter” address is able to set the fees receiving address.
- The owner is also able to burn tokens if the “unlocked” value is set to 1
    - Be aware of this

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

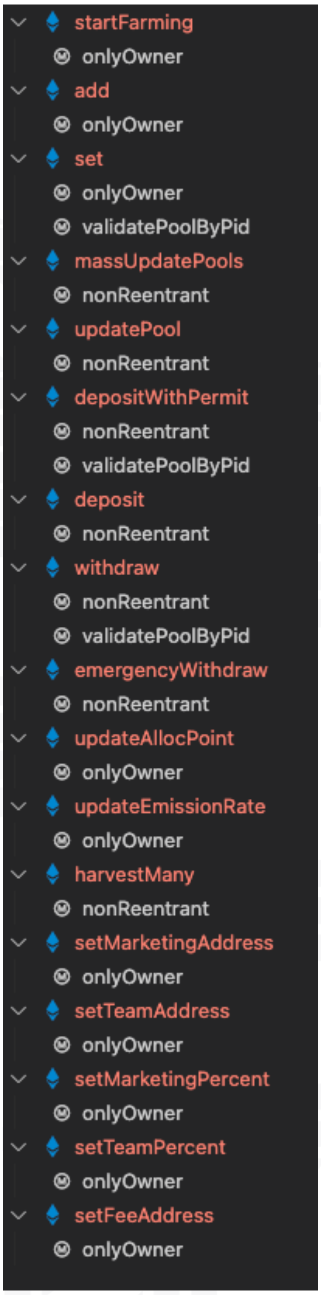
Farm

MultipleRewards.sol



```
add
    onlyOwner
addRewardInfo 💰
    onlyOwner
updatePool
    nonReentrant
_updatePool
massUpdatePools
    nonReentrant
onZyberReward
    onlyDistributorV2
    nonReentrant
emergencyRewardWithdraw
    onlyOwner
    nonReentrant
```

ZyberChef



```
startFarming
    onlyOwner
add
    onlyOwner
set
    onlyOwner
    validatePoolByPid
massUpdatePools
    nonReentrant
updatePool
    nonReentrant
depositWithPermit
    nonReentrant
    validatePoolByPid
deposit
    nonReentrant
withdraw
    nonReentrant
    validatePoolByPid
emergencyWithdraw
    nonReentrant
updateAllocPoint
    onlyOwner
updateEmissionRate
    onlyOwner
harvestMany
    nonReentrant
setMarketingAddress
    onlyOwner
setTeamAddress
    onlyOwner
setMarketingPercent
    onlyOwner
setTeamPercent
    onlyOwner
setFeeAddress
    onlyOwner
```

Note:

- General fork from Trader Joe/Sushi
- Farm/libraries
    - Files inside are the same as the Joe-core
        - https://github.com/traderjoe-xyz/joe-core/tree/main/contracts/libraries
            - Differences are changed pragma versions

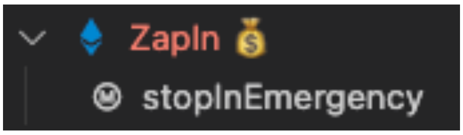
Ownership Privileges:

- Add a new pool
- Start farming in the farm contract, add new LP to the pool
- Set deposit fees, harvest interval, and allocation point
- Update the reward info for a pool including the end time and reward per second.
- Update allocation points, emission rate to any arbitrary value.
- Set marketing address, team address, and fee address.
- Set marketing and team percentage in fees but not more than 20%
- Update pool/pools
- Owner can also withdraw rewards of the users from the contract
  - Be aware of this

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

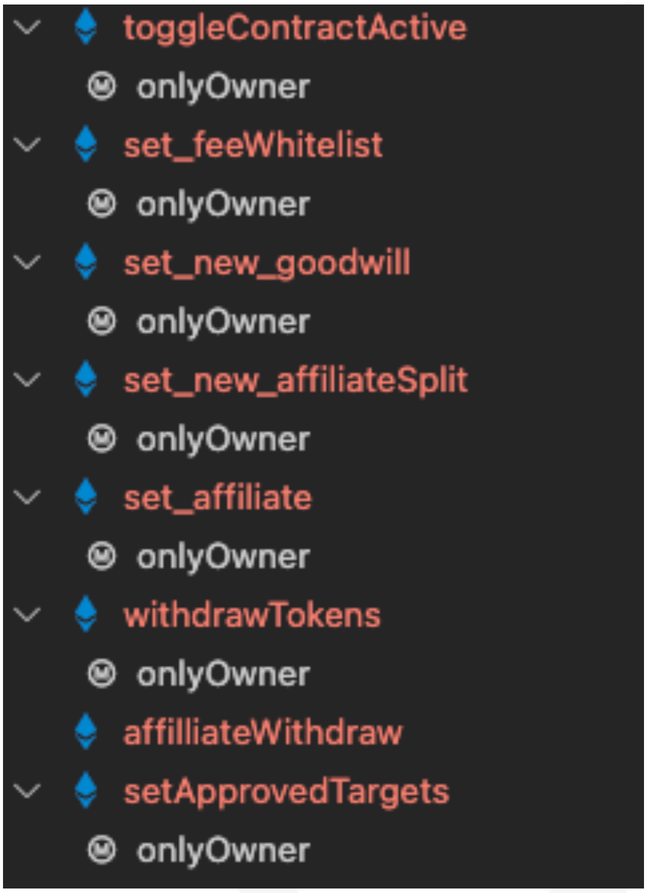
Zap

Zap.sol



ZapIn 💰
stopInEmergency

ZapBaseV2.sol



toggleContractActive
onlyOwner
set_feeWhitelist
onlyOwner
set_new_goodwill
onlyOwner
set_new_affiliateSplit
onlyOwner
set_affiliate
onlyOwner
withdrawTokens
onlyOwner
affilliateWithdraw
setApprovedTargets
onlyOwner

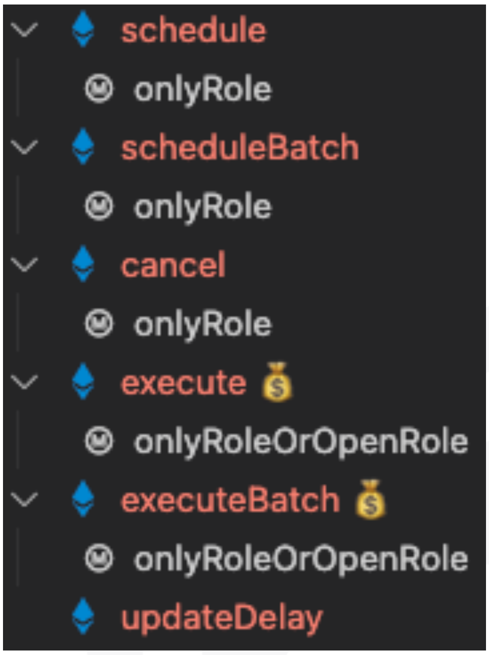
Ownership Privileges:

- Pause/Unpause the contract
- Set fees whitelist, and new goodwill within the range of '0-100'
- Set new affiliate addresses and Split up to a maximum of 100
- Withdraw tokens from the "ZapBaseV2Contract".
- Set Approved targets.

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

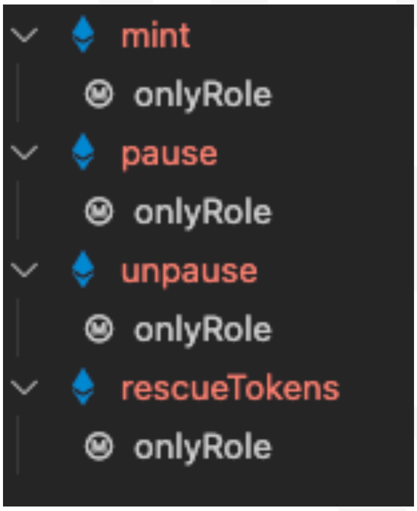
Miscellaneous

TimelockController.sol



```
schedule
    onlyRole
scheduleBatch
    onlyRole
cancel
    onlyRole
execute 💰
    onlyRoleOrOpenRole
executeBatch 💰
    onlyRoleOrOpenRole
updateDelay
```

ZyberToken.sol



```
mint
    onlyRole
pause
    onlyRole
unpause
    onlyRole
rescueTokens
    onlyRole
```

Ownership Privileges:

- *The wallet/account with the “MINTER_ROLE”, granted by the owner can mint new tokens*
- *The wallet/account with the “PAUSER_ROLE”, granted by the owner can pause/unpause the contract*
- *The wallet/account with the “RESCUER_ROLE”, granted by the owner can transfer any tokens from the contract*

- *The wallet/account with the “PROPOSER_ROLE”, granted by the owner can schedule an operation containing a single transaction or a batch of transactions.*

- *The wallet/account with the “PROPOSER_ROLE”, granted by the owner can cancel an operation, and execute transactions.*

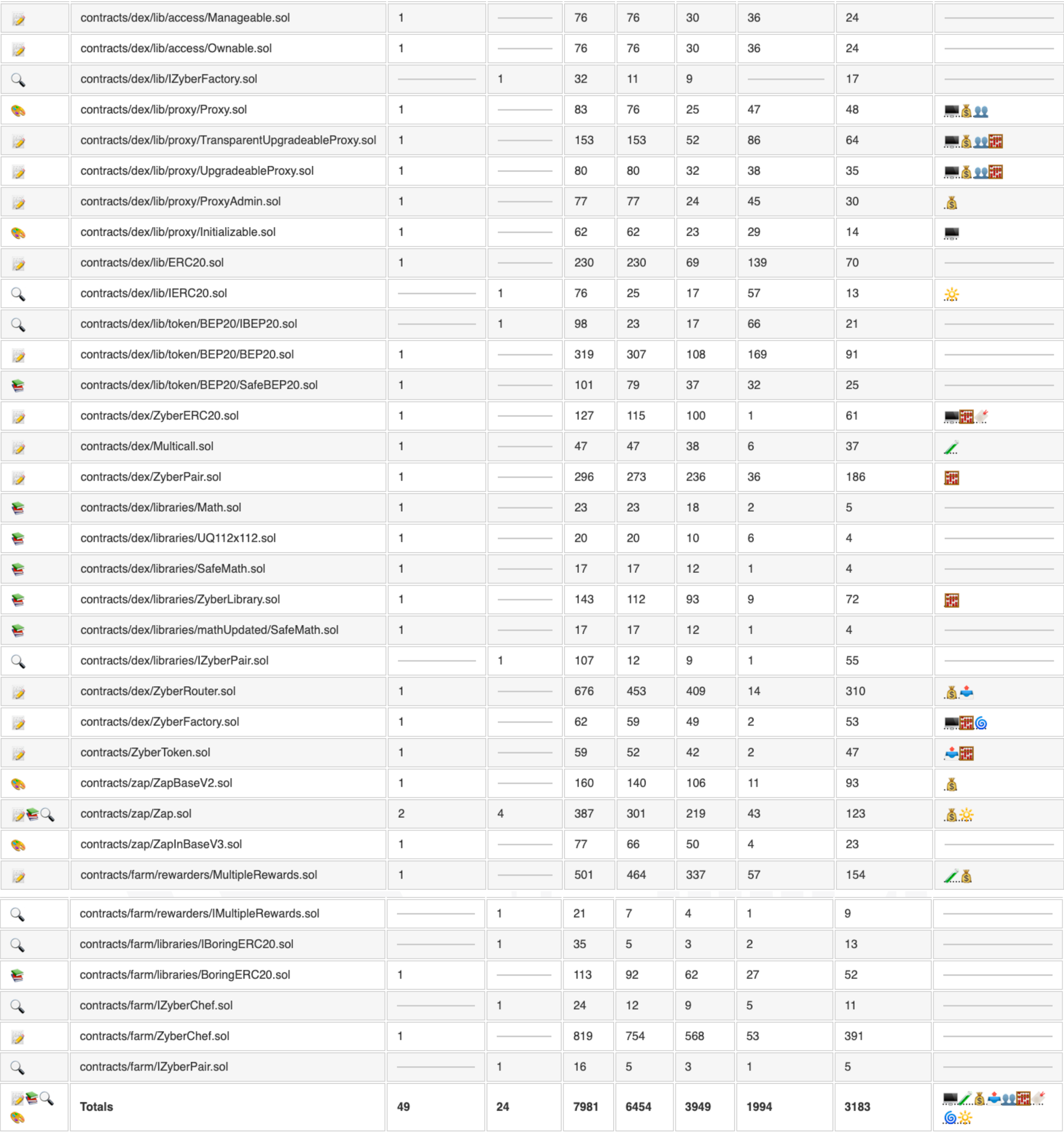
- There are several authorities which are authorized to call some functions, that means, if the owner is renounced, another address is still authorized to call functions
    - Be aware of this

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

Source Units in Scope

v1.0

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
📝📚🔍🎨	contracts/TimelockController.sol	5	2	927	782	413	391	258	💰🧮🔆
📝	contracts/TestERC20.sol	1		22	22	17	1	11	
🔍	contracts/dex/interfaces/IZyberCallee.sol		1	11	5	3	1	3	
🔍	contracts/dex/interfaces/IZyberRouter01.sol		1	150	4	3		48	💰
🔍	contracts/dex/interfaces/IWETH.sol		1	9	4	3		10	💰
🔍	contracts/dex/interfaces/IZyberRouter02.sol		1	50	6	4		16	💰
🔍	contracts/dex/interfaces/IZyberERC20.sol		1	52	12	9	1	27	
🔍	contracts/dex/interfaces/IZyberFactory.sol		1	33	12	9	1	17	
🔍	contracts/dex/interfaces/IZyberPair.sol		1	107	12	9	1	55	
🔍	contracts/dex/interfaces/IERC20.sol		1	19	9	5	1	19	
🔍	contracts/dex/interfaces/V1/IUniswapV1Exchange.sol		1	9	4	3		14	💰
🔍	contracts/dex/interfaces/V1/IUniswapV1Factory.sol		1	5	4	3		3	
📝	contracts/dex/Multicall2.sol	1		123	91	70	5	68	🧪
📝	contracts/dex/lib/ERC20Detailed.sol	1		54	54	21	27	14	🔆
📚	contracts/dex/lib/TransferHelper.sol	1		51	38	28	5	26	
📚	contracts/dex/lib/math/SafeMath.sol	1		189	177	54	107	14	
📚	contracts/dex/lib/utils/Memory.sol	1		108	108	71	28	123	🖥🧮
📚	contracts/dex/lib/utils/TransferHelper.sol	1		41	28	19	5	26	
📚	contracts/dex/lib/utils/Create2.sol	1		65	57	20	33	29	🖥🧮🌀
📚	contracts/dex/lib/utils/SafeBEP20Namer.sol	1		94	94	65	18	76	
📚	contracts/dex/lib/utils/EnumerableSet.sol	1		242	242	77	136	29	
📚	contracts/dex/lib/utils/PairNamer.sol	1		48	39	30	4	11	
📚	contracts/dex/lib/utils/Address.sol	1		161	128	57	87	37	🖥
📚	contracts/dex/lib/utils/FixedPoint.sol	1		76	76	45	17	29	
📚	contracts/dex/lib/utils/AddressStringUtil.sol	1		35	35	23	8	26	
📝	contracts/dex/lib/utils/ReentrancyGuard.sol	1		62	62	15	38	5	🔆
📝	contracts/dex/lib/GSN/Context.sol	1		28	28	11	14	1	🔆



📝	contracts/dex/lib/access/Manageable.sol	1		76	76	30	36	24	
📝	contracts/dex/lib/access/Ownable.sol	1		76	76	30	36	24	
🔍	contracts/dex/lib/IZyberFactory.sol		1	32	11	9		17	
🎨	contracts/dex/lib/proxy/Proxy.sol	1		83	76	25	47	48	🖥💰👥
📝	contracts/dex/lib/proxy/TransparentUpgradeableProxy.sol	1		153	153	52	86	64	🖥💰👥🧮
📝	contracts/dex/lib/proxy/UpgradeableProxy.sol	1		80	80	32	38	35	🖥💰👥🧮
📝	contracts/dex/lib/proxy/ProxyAdmin.sol	1		77	77	24	45	30	💰
🎨	contracts/dex/lib/proxy/Initializable.sol	1		62	62	23	29	14	🖥
📝	contracts/dex/lib/ERC20.sol	1		230	230	69	139	70	
🔍	contracts/dex/lib/IERC20.sol		1	76	25	17	57	13	🔆
🔍	contracts/dex/lib/token/BEP20/IBEP20.sol		1	98	23	17	66	21	
📝	contracts/dex/lib/token/BEP20/BEP20.sol	1		319	307	108	169	91	
📚	contracts/dex/lib/token/BEP20/SafeBEP20.sol	1		101	79	37	32	25	
📝	contracts/dex/ZyberERC20.sol	1		127	115	100	1	61	🖥🧮🔖
📝	contracts/dex/Multicall.sol	1		47	47	38	6	37	🧪
📝	contracts/dex/ZyberPair.sol	1		296	273	236	36	186	🧮
📚	contracts/dex/libraries/Math.sol	1		23	23	18	2	5	
📚	contracts/dex/libraries/UQ112x112.sol	1		20	20	10	6	4	
📚	contracts/dex/libraries/SafeMath.sol	1		17	17	12	1	4	
📚	contracts/dex/libraries/ZyberLibrary.sol	1		143	112	93	9	72	🧮
📚	contracts/dex/libraries/mathUpdated/SafeMath.sol	1		17	17	12	1	4	
🔍	contracts/dex/libraries/IZyberPair.sol		1	107	12	9	1	55	
📝	contracts/dex/ZyberRouter.sol	1		676	453	409	14	310	💰📤
📝	contracts/dex/ZyberFactory.sol	1		62	59	49	2	53	🖥🧮🌀
📝	contracts/ZyberToken.sol	1		59	52	42	2	47	📤🧮
🎨	contracts/zap/ZapBaseV2.sol	1		160	140	106	11	93	💰
📝📚🔍	contracts/zap/Zap.sol	2	4	387	301	219	43	123	💰🔆
🎨	contracts/zap/ZapInBaseV3.sol	1		77	66	50	4	23	
📝	contracts/farm/rewarders/MultipleRewards.sol	1		501	464	337	57	154	🧪💰
🔍	contracts/farm/rewarders/IMultipleRewards.sol		1	21	7	4	1	9	
🔍	contracts/farm/libraries/IBoringERC20.sol		1	35	5	3	2	13	
📚	contracts/farm/libraries/BoringERC20.sol	1		113	92	62	27	52	
🔍	contracts/farm/IZyberChef.sol		1	24	12	9	5	11	
📝	contracts/farm/ZyberChef.sol	1		819	754	568	53	391	
🔍	contracts/farm/IZyberPair.sol		1	16	5	3	1	5	
📝📚🔍 🎨	**Totals**	**49**	**24**	**7981**	**6454**	**3949**	**1994**	**3183**	🖥🧪💰📤👥🧮🔖 🌀🔆

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		Contract doesn't import npm packages from source (like OpenZeppelin etc.)	-	We recommend to import all packages from npm directly without flatten the contract. Functions could be modified or can be susceptible to vulnerabilities
#2	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.
#3	ZyberFactory.sol	Missing Zero Address Validation (missing-zero-check)	5.358	Check that the address is not zero
#4	MultipleRewards.sol	Missing Zero Address Validation (missing-zero-check)	481	Check that the address is not zero otherwise the amount will be lost
#5	ZyberPair.sol	Missing Zero Address Validation (missing-zero-check)	88	Check that the address is not zero otherwise the amount will be lost

#6	ZapBaseV2.sol	Missing Zero Address Validation (missing-zero-check)	77.103.146	Check that the address is not zero
#7	ZyberChef.sol	Missing Events Arithmetic	182	Emit an event for critical parameter changes
#8	ZapBaseV2.sol	Missing Events Arithmetic	73-111, 146	Emit an event for critical parameter changes
#9	ZyberRouter.sol	Missing Events Arithmetic	All	Emit an event for critical parameter changes
#10	ZyberPair.sol	Raw mathematical operations	136	Since the contract is below pragma version 0.8.x the contract must use SafeMath library functions because of the under-/overflow vulnerability. We recommend you to replace raw mathematical operations with SafeMath library operations.
#11	ZapBaseV2.sol	Tautology	86	Fix the incorrect comparison by changing the value type or the comparison
#12	ZapBaseV2-.sol	State variable visibility	16	The visibility of the state variable is not set. We recommend you to specify the visibility.

Informational issues

Issue	File	Type	Line	Description
#1	ZapBaseV2.sol	Missing Existence check	77	We recommend to check the existence of an address in the whitelist of the fees before directly adding it.
#2	MultipleRewards.sol	Wrong comment	277	Check the comment. It says that the start timestamp will be used but you can see that the end timestamp of the the pool id will be checked instead. Either justify the comment or the logic.

#3	MultipleRewards.sol	Check division by 0	316	We recommend also to check for totalAllocPoint is zero or not because the contract is dividing with it in L316
#4	MultipleRewards.sol	Naming convention	266	Start private/internal functions with an underscore and public/externals without one.
#5	MultipleRewards.sol	Misspelling	See description	Change following words: - vairables L334 L339 Make sure to change it everywhere else as well.
#6	Zap.sol	Misspelling	See description	Change following words: - Excecution L116 - entrire L120 Make sure to change it everywhere else as well.
#7	ZapBaseV2.sol	Misspelling	See description	Change following words: - Affilliate L110 L128 Make sure to change it everywhere else as well.
#8	Multicall.sol	Error message is missing	21	Provide an error message for require statement
#9	ProxyAdmin.sol	Error message is missing	25, 40	Provide an error message for require statement
#10	TransparentUpgradeableProxy.sol	Error message is missing	120	Provide an error message for require statement
#11	UpgradeableProxy.sol	Error message is missing	30	Provide an error message for require statement
#12	Memory.sol	Error message is missing	26	Provide an error message for require statement
#13	Memory.sol	Unused state variables	8, 10	Remove unused state variables or use it.

#14	Zap.sol	Value can be 0	270	If the "_fromTokenAddress" is not zero address the "valueToSend" will be 0. We recommend you to check for 0 before calling the _swapTarget call" in L284

Information from the team

#1 Transfer own tokens from contract
Description: Rescuer is able to transfer contract tokens to the caller. That means that the rescuer can drain out the contract.

From the team: The contract can rescue any ERC20 token that gets mistakenly sent to the contract itself. Since the contract will not hold native tokens, we do not need to check the submitted address with our own contract address

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. January 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
  - uniswap
  - Trader Joe
  - zapper.fi
- Read whole report and modifiers section for more information
- 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	NOT 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	NOT 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