CONTRACT
WOLF

Blockchain Security - Smart Contract Audits

Security Assessment

April 7, 2022

Disclaimer

ContractWolf.io audits and reports should not be considered as a form of project's "advertisement" and does not cover any interaction and assessment from "project's contract" to "external contracts" such as Pancakeswap or similar.

ContractWolf does not provide any warranty on its released reports.

ContractWolf should not be used as a decision to invest into an audited project and is not affiliated nor partners to its audited contract projects.

ContractWolf provides transparent report to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within it’s **SMART CONTRACT**.

ContractWolf presence is to analyze, audit and assess the client's smart contract's code.

Each company or projects should be liable to its security flaws and functionalities.

Network

Binance Smart Chain (BEP20)

Website

https://lotuscapital.xyz

Telegram

https://t.me/lotuscapital

Twitter

https://twitter.com/LotusCapitalVC

Linkedin

https://www.linkedin.com/company/lotus-capital-vc

Medium

https://medium.com/@LotusCapital

E-mail

stefan@lotuscapital.xyz

Description

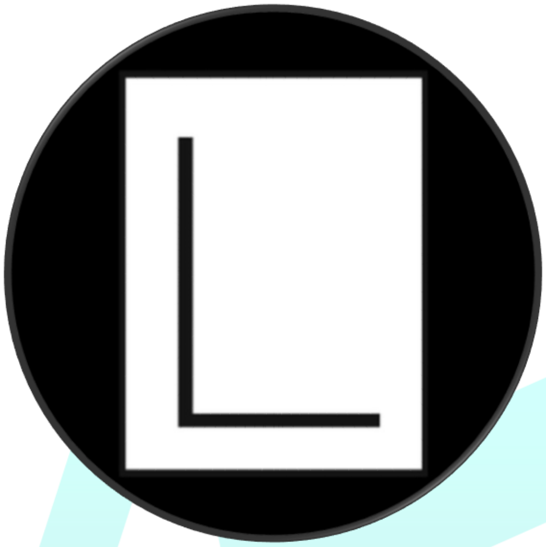
Lotus Capital secures and operates an IDO Launchpad to support early-stage project fundraising in the cryptocurrency-sector of the financial community. We utilize our own venture fund to help established businesses achieve exponential growth, known as the crypto-based Lotus Capital Venture Fund.

ContractWolf Engagement

7th of April 2022, **Lotus Capital** engaged and agrees to audit their smart contract's code by ContractWolf. The goal of this engagement was to identify if there is a possibility of security flaws in the implementation of the contract or system.

ContractWolf will be focusing on contract issues and functionalities along with the projects claims from smart contract to their website, whitepaper and repository which has been provided by **Lotus Capital.**

Logo



Contract link

IFOV2

- https://bscscan.com/address/0xBeE786b2E92C7DCe3aa07B85f37d37491Cb46C64

MasterBuilder

- https://bscscan.com/address/0xC96B0bd79D7fF44eF3Bc8A29561f4D6c83823006

DexTokenVault

- https://bscscan.com/address/0x249e40AB07A9153270857AC1FD4c2B9D8Bdb7959

Risk Level Classification

Risk Level represents the classification or the probability that a certain function or threat that can exploit vulnerability and have an impact within the system or contract.
Risk Level is computed based on CVSS Version 3.0

Level	Value	Vulnerability
Critical	9 - 10	An Exposure that can affect the contract functions in several events that can risk and disrupt the contract
High	7 - 8.9	An Exposure that can affect the outcome when using the contract that can serve as an opening in manipulating the contract in an unwanted manner
Medium	4 - 6.9	An opening that could affect the outcome in executing the contract in a specific situation
Low	0.1 - 3.9	An opening but doesn't have an impact on the functionality of the contract
Informational	0	An opening that consists of information's but will not risk or affect the contract

Auditing Approach

Every line of code along with its functionalities will undergo manual review to check its security issues, quality, and contract scope of inheritance. The manual review will be done by our team that will document any issues that there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:

   - Review of the specifications, sources, and instructions provided to ContractWolf to make sure we understand the size, scope, and functionality of the smart contract.
   - Manual review of code, our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities and security flaws.

2. Testing and automated analysis that includes:

   - Testing the smart contract functions with common test cases and scenarios, to ensure that it returns the expected results.

3. Best practices review, the team will review the contract with the aim to improve efficiency, effectiveness, clarifications, maintainability, security, and control within the smart contract.

4. Recommendations to help the project take steps to secure the smart contract.

Used Code from other Frameworks/Smart Contracts (Direct Imports)

Imported Packages

IFOV2

- Context
- Ownable
- SafeMath
- ReentrancyGuard
- IBEP20
- Address
- SafeBEP20
- EnumerableSet
- AccessControl
- Counters
- IERC165
- IERC721
- IERC721Receiver
- ERC721Holder
- IIFOV2
- IFOV2

MasterBuilder

- Address
- BEP20
- Context
- DEXToken
- IBEP20
- MasterBuilder
- Ownable
- RewardToken
- SafeBEP20
- SafeMath

DexTokenVault

- Context
- Ownable
- IERC20
- SafeMath
- Address
- SafeERC20
- Pausable
- IMasterBuilder
- DexTokenVault
- VaultOwner

Description

Optimization enabled: Yes

Version: v0.6.12

Capabilities

Components

IFOV2				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	2	5	5	4

MasterBuilder				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	6	3	1	0

DexTokenVault				
Version	Contracts	Libraries	Interfaces	Abstract
1.0	2	3	2	3

Exposed Functions

IFOV2				
Version	Public	Private	External	Internal
1.0	11	9	43	56

MasterBuilder				
Version	Public	Private	External	Internal
1.0	31	1	23	41

DexTokenVault				
Version	Public	Private	External	Internal
1.0	7	2	38	34

State Variables

IFOV2		
Version	Total	Public
1.0	17	9

MasterBuilder		
Version	Total	Public
1.0	26	19

DexTokenVault		
Version	Total	Public
1.0	19	17

Capabilities

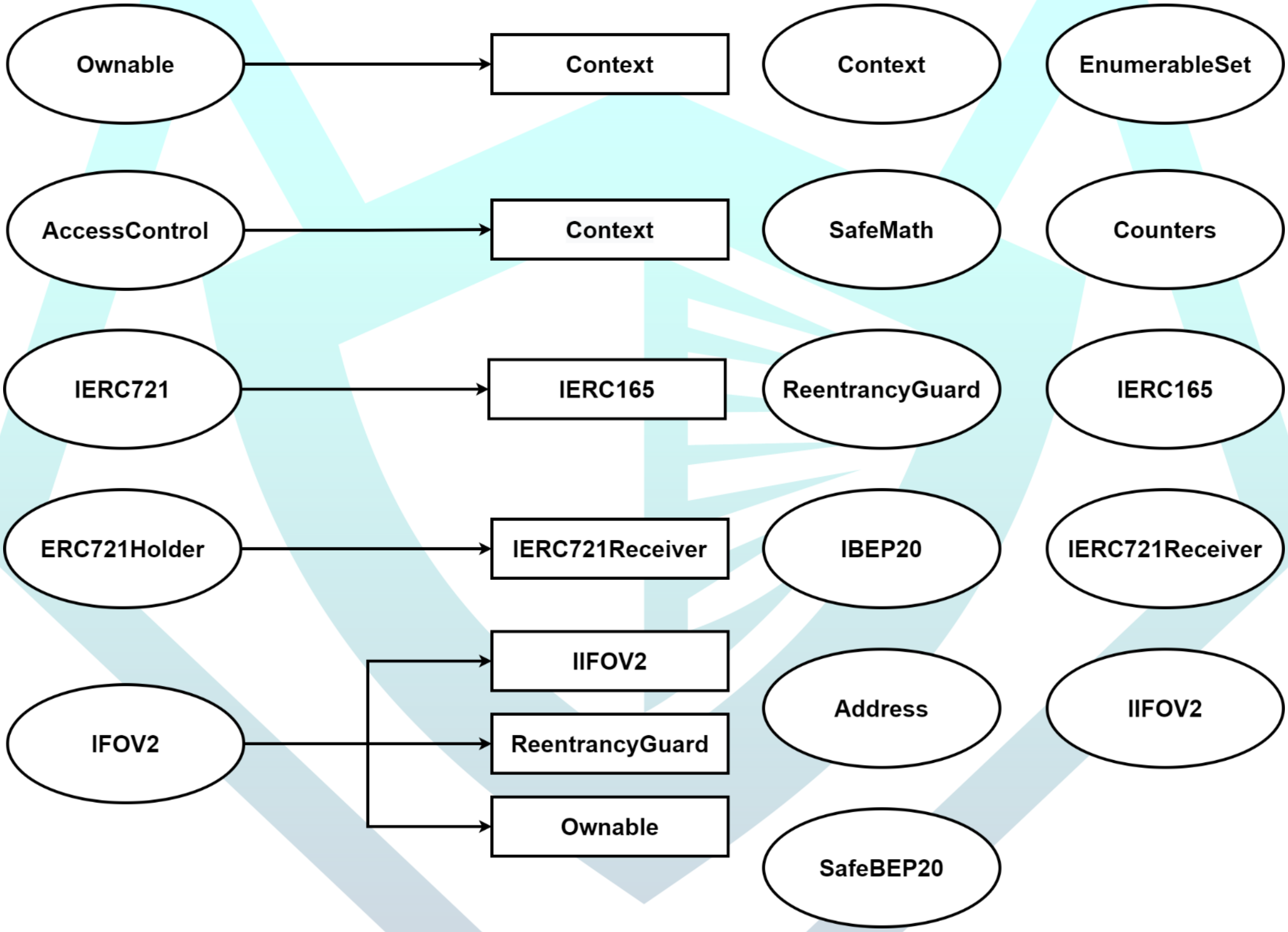
Version	Solidity Versions Observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	v0.6.12		No	Yes	No

Scope of Work

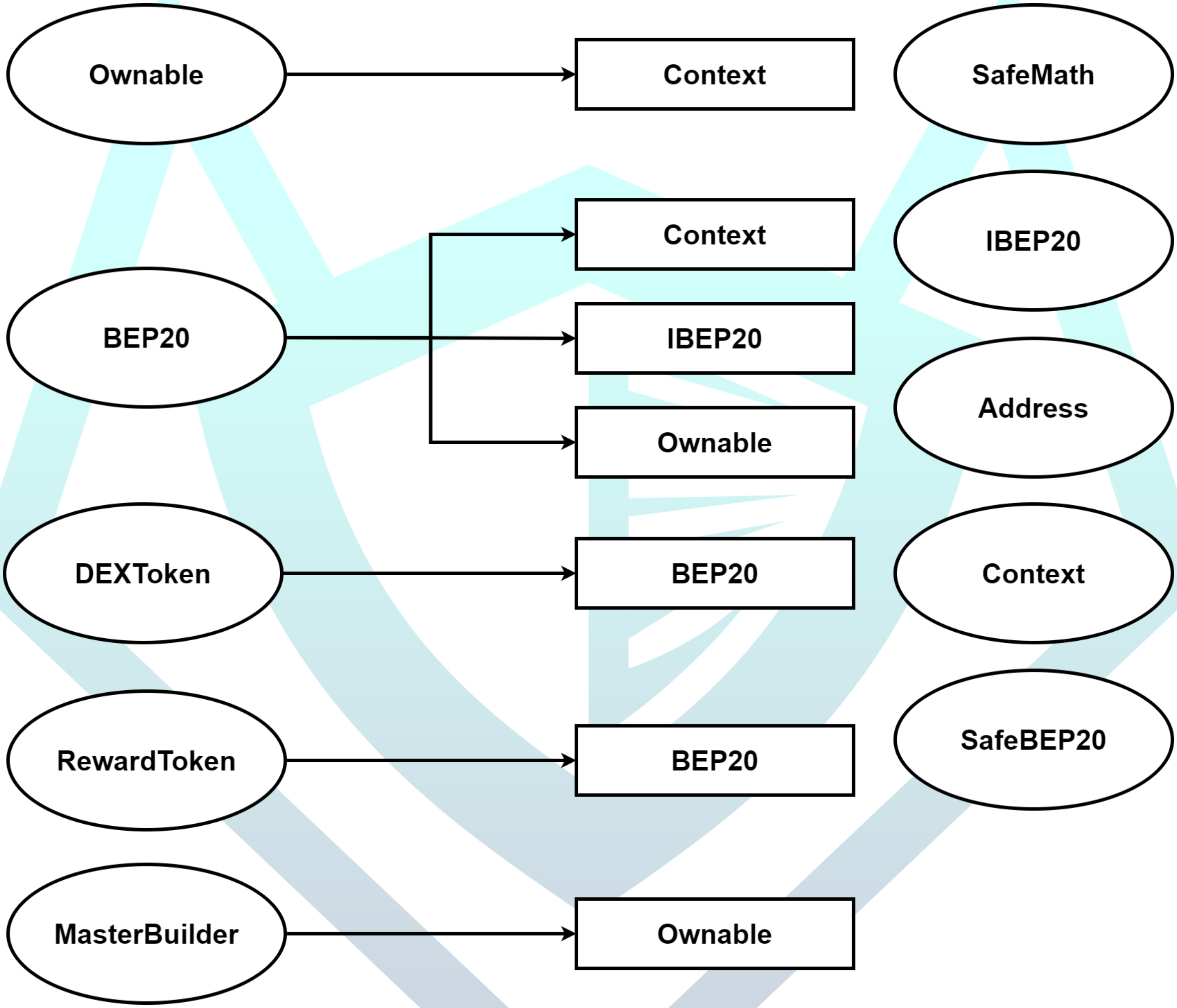
Lotus Capital's team provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract.

Inheritance Graph

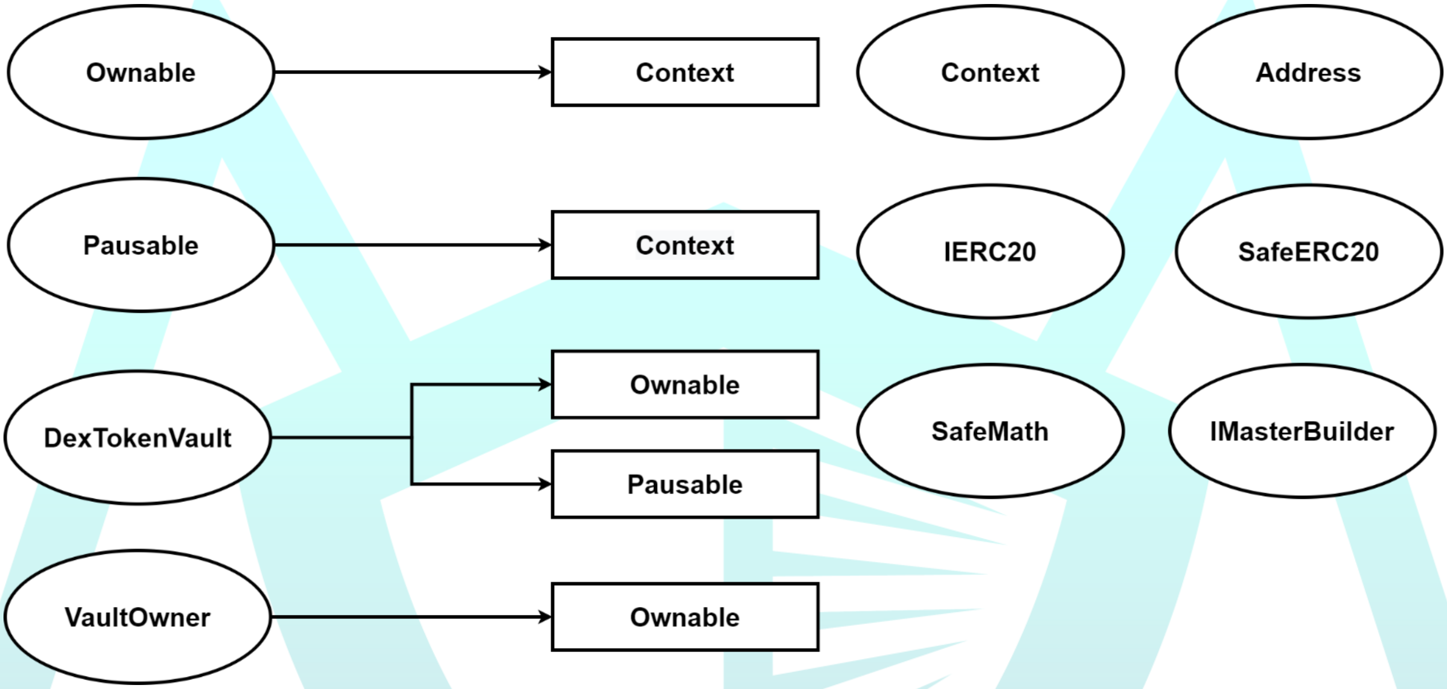
IFOV2



MasterBuilder



DexTokenVault



Verify Claims

Correct implementation of Token Standard

Tested	Verified
✓	✗

Function	Description	Exist	Tested	Verified
TotalSupply	Information about the total coin or token supply	✓	✓	✓
BalanceOf	Details on the account balance from a specified address	✓	✓	✓
Transfer	An action that transfers a specified amount of coin or token to a specified address	✓	✓	✓
TransferFrom	An action that transfers a specified amount of coin or token from a specified address	✓	✓	✓
Approve	Provides permission to withdraw specified number of coin or token from a specified address	✓	✓	✓

Function	IFOV2	MasterBuilder	DexTokenVault
Deployer can renounce ownership	✓	✓	✓

Statement	IFOV2	MasterBuilder	DexTokenVault
Deployer can mint after deployment	–	✓	–

Statement	IFOV2	MasterBuilder	DexTokenVault
Deployer cannot block user	–	–	–

Statement	IFOV2	MasterBuilder	DexTokenVault
Deployer can burn	–	✓	–

Statement	IFOV2	MasterBuilder	DexTokenVault
Deployer can pause	–	–	✓

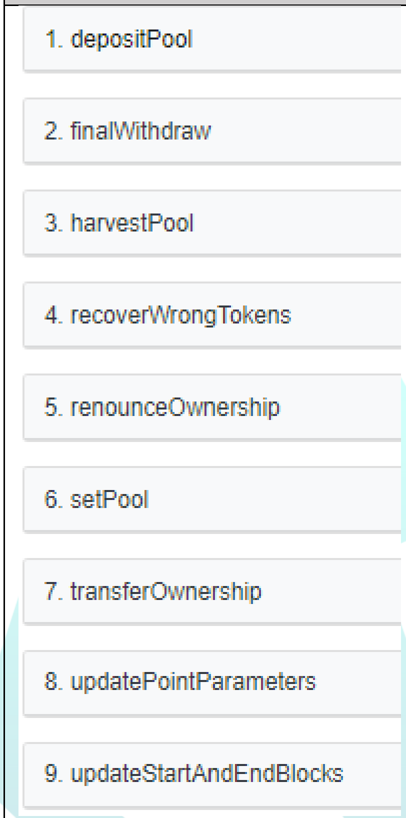
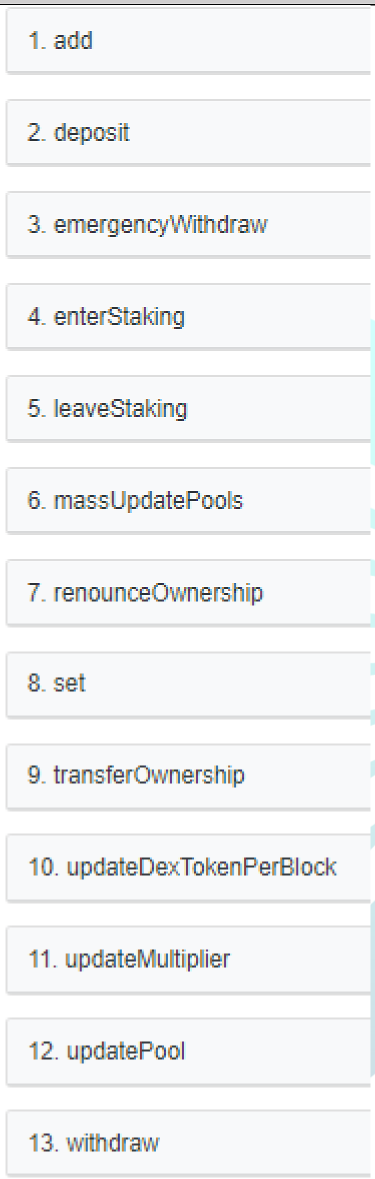
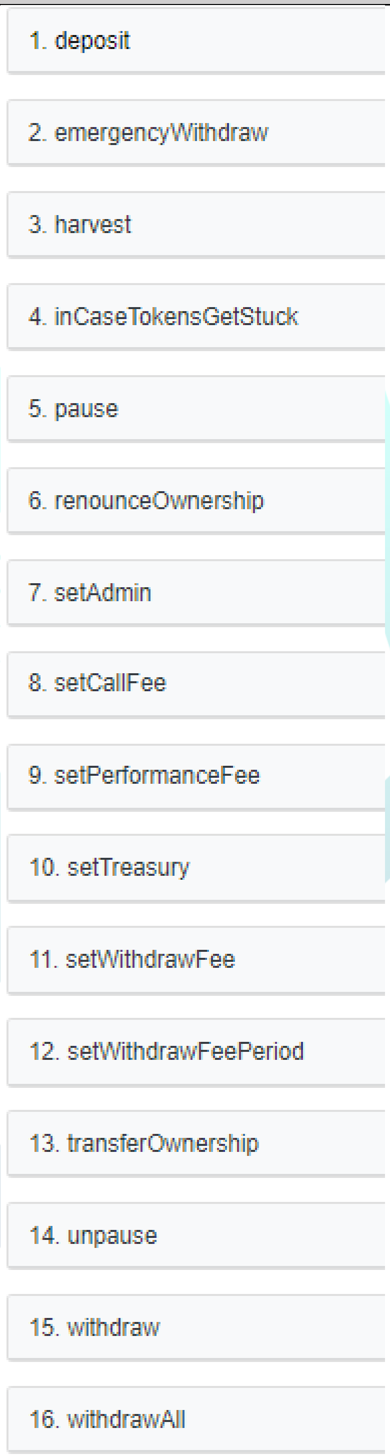
Overall Checkup (Smart Contract Security)

Tested	Verified
✓	✓

Legend

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

Write Functions of Contract

IFOV2	MasterBuilder	DexTokenVault
 1. depositPool 2. finalWithdraw 3. harvestPool 4. recoverWrongTokens 5. renounceOwnership 6. setPool 7. transferOwnership 8. updatePointParameters 9. updateStartAndEndBlocks	 1. add 2. deposit 3. emergencyWithdraw 4. enterStaking 5. leaveStaking 6. massUpdatePools 7. renounceOwnership 8. set 9. transferOwnership 10. updateDexTokenPerBlock 11. updateMultiplier 12. updatePool 13. withdraw	 1. deposit 2. emergencyWithdraw 3. harvest 4. inCaseTokensGetStuck 5. pause 6. renounceOwnership 7. setAdmin 8. setCallFee 9. setPerformanceFee 10. setTreasury 11. setWithdrawFee 12. setWithdrawFeePeriod 13. transferOwnership 14. unpause 15. withdraw 16. withdrawAll

AUDIT PASSED

Low Issues

IFOV2	
A floating pragma is set (SWC-103)	L: 9, L: 34, L: 102, L: 329, L: 329, L: 491, L: 708, L: 804, L: 1101, L: 1315, L: 1354, L: 1379, L: 1520, L: 1547, L: 1574, L: 1685
Use of “tx.origin” as part of authorization control (SWC-115)	L: 1770 C: 30
Potential use of “block.number” as source of randomness (SWC-120)	L: 1818 C: 16, L: 1821 C: 16, L: 1853 C: 16, L: 1949 C: 16, L: 1972 C: 16, L: 1987 C: 16, L: 1989 C: 16,

MasterBuilder	
Read of persistent state following external call / Write to persistent state following external call (SWC - 107)	L: 1670 C: 26, L: 1670 C: 12, L: 1672 C: 42, L: 1672 C: 26, L: 1672 C: 8, L: 1693 C: 42, L: 1693 C: 26, L: 1693 C: 8, L: 1743 C: 49, L: 1744 C: 8, L: 1745 C: 8
Potential use of "block.number" as source of randonmness (SWC - 120)	L: 1101 C: 30, L: 1174 C: 36, L: 1366 C: 30, L: 1439 C: 36, L: 1567 C: 34, L: 1567 C: 62, L: 1618 C: 12, L: 1619 C: 69, L: 1638 C: 12, L: 1643 C: 35, L: 1646 C: 65, L: 1651 C: 31,
Requirement violation (SWC - 123)	L: 429 C: 50, L: 1472

DexTokenVault	
A floating pragma is set (SWC-103)	L: 9, L: 34, L: 102, L: 329, L: 329, L: 491, L: 708, L: 804, L: 1101, L: 1315, L: 1354, L: 1379, L: 1520, L: 1547, L: 1574, L: 1685
Use of “tx.origin” as part of authorization control (SWC-115)	L: 1770 C: 30
Potential use of “block.number” as source of randomness (SWC-120)	L: 1818 C: 16, L: 1821 C: 16, L: 1853 C: 16, L: 1949 C: 16, L: 1972 C: 16, L: 1987 C: 16, L: 1989 C: 16,

Audit Comments

IFOV2

- Deployer cannot mint after initial deployment
- Deployer cannot burn
- Deployer cannot block user
- Deployer cannot pause contract
- Deployer can renounce ownership
- Deployer can transfer ownership
- Deployer can take tokens from contract
- Deployer can modify pool setting
- Deployer can update start/end blocks
- Deployer can withdraw liquidity pool and offering token

MasterBuilder

- Deployer can renounce ownership
- Deployer can transfer ownership
- Deployer can mint tokens
- Deployer can burn
- Deployer can transfer dex token
- Deployer can update dex token per block
- Deployer can add liquidity pool
- Deployer can set allocation on liquidty pool

DexTokenVault

- Deployer can renounce ownership
- Deployer can renounce ownership
- Deployer can transfer ownership
- Deployer can set admin
- Deployer can set treasury
- Deployer/Admin can pause/unpause contract
- Deployer/Admin can set fees with an indefinite amount
- Deployer/Admin can collect tokens from contract
- Admin can collect fees
- Admin can withdraw from MasterBuilder contract

CONTRACTWOLF

Blockchain Security - Smart Contract Audits