AUDITEK

AUGUST 2021

Financial Audit Report

Smart Contract Security Audit

LITTLE DOGE CAKE

PRESENTED BY

Davis Thorne
and Partners

Audit Details

Audited project

LITTLE DOGE CAKE

Deployer address

0x414c7ce1961ae4cede5bfba9735098d1013cddf5

Client contacts:

LITTLE DOGE CAKE TEAM

Blockchain

Binance Smart Chain

Project website:

https://littledogecake.com/

Disclaimer

Background

Auditek was commissioned by Little Doge Cake to perform an audit of smart contracts:
https://bscscan.com/address/0xbc537c876510083f7d2406c12fdd93cbe8da4b8a

The purpose of the audit was to achieve the following:

- **Ensure that the smart contract functions as intended.**
- **Identify potential security issues with the smart contract.**

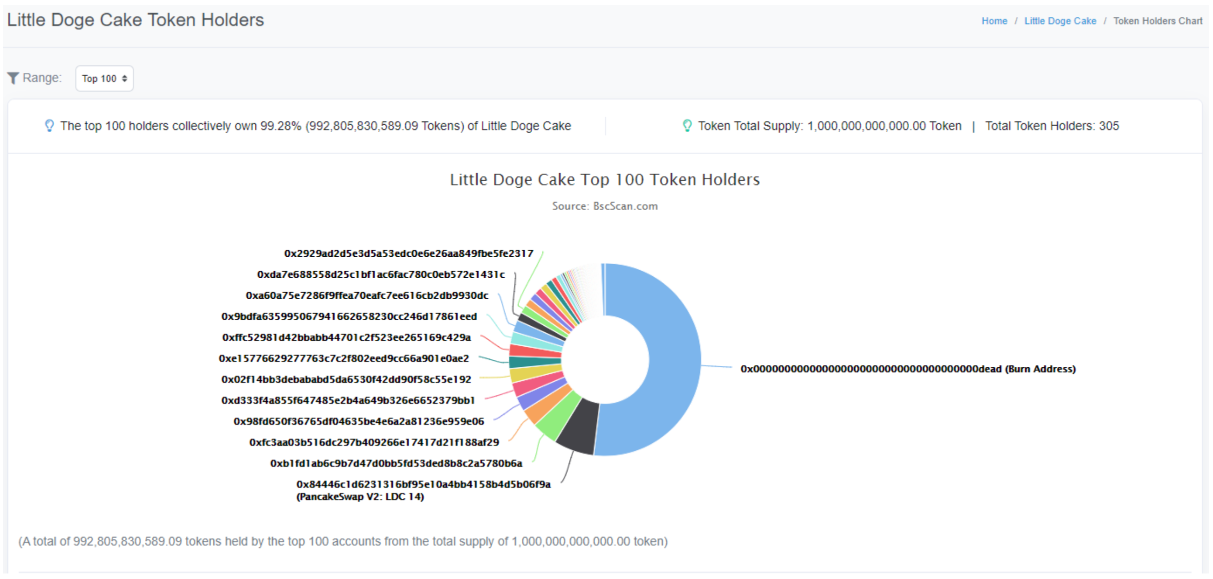
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 22.07.2021

Contract name	Little Doge Cake
Contract address	0xbc537c876510083f7d2406c12fdd93cbe8da4b8a
Total supply	10,000,000,000
Token ticker	LDC
Decimals	9
Token holders	305
Transactions count	6,003
Top 100 holders dominance	100.00%
Liquidity fee	9
Tax fee	2
Total fees	0
Uniswap V2 pair	0x93d94fcb0dcc8a88257b2d2eec7a2615ebedb542
Contract deployer address	0x414c7ce1961ae4cede5bfba9735098d1013cddf5
Contract's current owner address	0x414c7ce1961ae4cede5bfba9735098d1013cddf5

LITTLE DOGE CAKE Token Distribution



Little Doge Cake Token Holders

Home / Little Doge Cake / Token Holders Chart

Range: Top 100

The top 100 holders collectively own 99.28% (992,805,830,589.09 Tokens) of Little Doge Cake

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 305

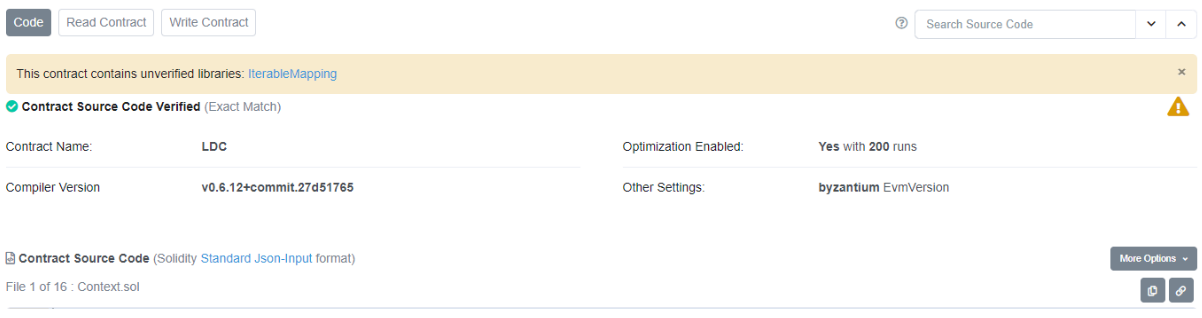
Little Doge Cake Top 100 Token Holders

Source: BscScan.com

0x2929ad2d5e3d5a53edc0e6e26aa849fbe5fe2317
0xda7e688558d25c1bf1ac6fac780c0eb572e1431c
0xa60a75e7286f9ffea70eafc7ee616cb2db9930dc
0x9bdfa635995067941662658230cc246d17861eed
0xffc52981d42bbabb44701c2f523ee265169c429a
0xe15776629277763c7c2f802eed9cc66a901e0ae2
0x02f14bb3debababd5da6530f42dd90f58c55e192
0xd333f4a855f647485e2b4a649b326e6652379bb1
0x98fd650f36765df04635be4e6a2a81236e959e06
0xfc3aa03b516dc297b409266e17417d21f188af29
0xb1fd1ab6c9b7d47d0bb5fd53ded8b8c2a5780b6a
0x84446c1d6231316bf95e10a4bb4158b4d5b06f9a (PancakeSwap V2: LDC 14)
0x000000000000000000000000000000000000dead (Burn Address)

(A total of 992,805,830,589.09 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000.00 token)

LITTLE DOGE CAKE Contract Interaction Details



Code | Read Contract | Write Contract

Search Source Code

This contract contains unverified libraries: IterableMapping

Contract Source Code Verified (Exact Match)

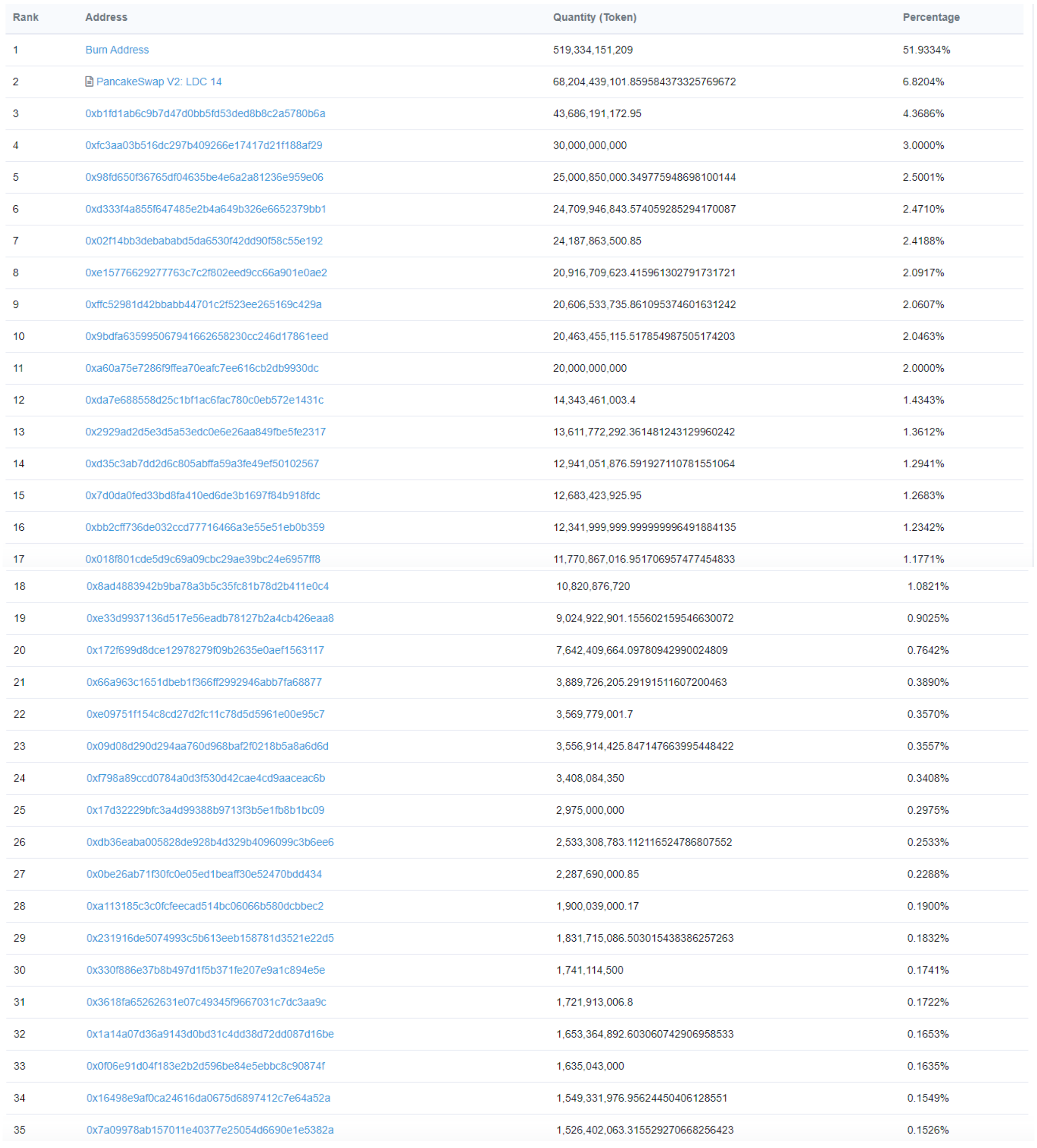
Contract Name:	LDC	Optimization Enabled:	Yes with 200 runs
Compiler Version	v0.6.12+commit.27d51765	Other Settings:	byzantium EvmVersion

Contract Source Code (Solidity Standard Json-Input format)

More Options

File 1 of 16 : Context.sol

LITTLE DOGE CAKE Top 30 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	Burn Address	519,334,151,209	51.9334%
2	PancakeSwap V2: LDC 14	68,204,439,101.859584373325769672	6.8204%
3	0xb1fd1ab6c9b7d47d0bb5fd53ded8b8c2a5780b6a	43,686,191,172.95	4.3686%
4	0xfc3aa03b516dc297b409266e17417d21f188af29	30,000,000,000	3.0000%
5	0x98fd650f36765df04635be4e6a2a81236e959e06	25,000,850,000.349775948698100144	2.5001%
6	0xd333f4a855f647485e2b4a649b326e6652379bb1	24,709,946,843.574059285294170087	2.4710%
7	0x02f14bb3debababd5da6530f42dd90f58c55e192	24,187,863,500.85	2.4188%
8	0xe15776629277763c7c2f802eed9cc66a901e0ae2	20,916,709,623.415961302791731721	2.0917%
9	0xffc52981d42bbabb44701c2f523ee265169c429a	20,606,533,735.861095374601631242	2.0607%
10	0x9bdfa635995067941662658230cc246d17861eed	20,463,455,115.517854987505174203	2.0463%
11	0xa60a75e7286f9ffea70eafc7ee616cb2db9930dc	20,000,000,000	2.0000%
12	0xda7e688558d25c1bf1ac6fac780c0eb572e1431c	14,343,461,003.4	1.4343%
13	0x2929ad2d5e3d5a53edc0e6e26aa849fbe5fe2317	13,611,772,292.361481243129960242	1.3612%
14	0xd35c3ab7dd2d6c805abffa59a3fe49ef50102567	12,941,051,876.591927110781551064	1.2941%
15	0x7d0da0fed33bd8fa410ed6de3b1697f84b918fdc	12,683,423,925.95	1.2683%
16	0xbb2cff736de032ccd77716466a3e55e51eb0b359	12,341,999,999.999999996491884135	1.2342%
17	0x018f801cde5d9c69a09cbc29ae39bc24e6957ff8	11,770,867,016.951706957477454833	1.1771%
18	0x8ad4883942b9ba78a3b5c35fc81b78d2b411e0c4	10,820,876,720	1.0821%
19	0xe33d9937136d517e56eadb78127b2a4cb426eaa8	9,024,922,901.155602159546630072	0.9025%
20	0x172f699d8dce12978279f09b2635e0aef1563117	7,642,409,664.09780942990024809	0.7642%
21	0x66a963c1651dbeb1f366ff2992946abb7fa68877	3,889,726,205.29191511607200463	0.3890%
22	0xe09751f154c8cd27d2fc11c78d5d5961e00e95c7	3,569,779,001.7	0.3570%
23	0x09d08d290d294aa760d968baf2f0218b5a8a6d6d	3,556,914,425.847147663995448422	0.3557%
24	0xf798a89ccd0784a0d3f530d42cae4cd9aaceac6b	3,408,084,350	0.3408%
25	0x17d32229bfc3a4d99388b9713f3b5e1fb8b1bc09	2,975,000,000	0.2975%
26	0xdb36eaba005828de928b4d329b4096099c3b6ee6	2,533,308,783.112116524786807552	0.2533%
27	0x0be26ab71f30fc0e05ed1beaff30e52470bdd434	2,287,690,000.85	0.2288%
28	0xa113185c3c0fcfeecad514bc06066b580dcbbec2	1,900,039,000.17	0.1900%
29	0x231916de5074993c5b613eeb158781d3521e22d5	1,831,715,086.503015438386257263	0.1832%
30	0x330f886e37b8b497d1f5b371fe207e9a1c894e5e	1,741,114,500	0.1741%
31	0x3618fa65262631e07c49345f9667031c7dc3aa9c	1,721,913,006.8	0.1722%
32	0x1a14a07d36a9143d0bd31c4dd38d72dd087d16be	1,653,364,892.603060742906958533	0.1653%
33	0x0f06e91d04f183e2b2d596be84e5ebbc8c90874f	1,635,043,000	0.1635%
34	0x16498e9af0ca24616da0675d6897412c7e64a52a	1,549,331,976.95624450406128551	0.1549%
35	0x7a09978ab157011e40377e25054d6690e1e5382a	1,526,402,063.315529270668256423	0.1526%

Contract functions details

```
+  Context
   - [Int] _msgSender
   - [Int] _msgData

 + [Int] IERC20
   - [Ext] totalSupply
   - [Ext] balanceOf
   - [Ext] transfer #
   - [Ext] allowance
   - [Ext] approve #
   - [Ext] transferFrom #

 + [Lib] SafeMath
   - [Int] add
   - [Int] sub
   - [Int] sub
   - [Int] mul
   - [Int] div
   - [Int] div
   - [Int] mod
   - [Int] mod

 + [Lib] Address
   - [Int] isContract
   - [Int] sendValue #
   - [Int] functionCall #
   - [Int] functionCall #
   - [Int] functionCallWithValue #
   - [Int] functionCallWithValue #
   - [Prv] _functionCallWithValue #

 +  Ownable (Context)
   - [Pub] <Constructor> #
   - [Pub] owner
   - [Pub] renounceOwnership #
      - modifiers: onlyOwner
   - [Pub] transferOwnership #
      - modifiers: onlyOwner
   - [Pub] getUnlockTime
   - [Pub] getTime
   - [Pub] lock #
      - modifiers: onlyOwner
   - [Pub] unlock #

 + [Int] IUniswapV2Factory
   - [Ext] feeTo
   - [Ext] feeToSetter
   - [Ext] getPair
   - [Ext] allPairs
   - [Ext] allPairsLength
   - [Ext] createPair #
```

  - [Ext] setFeeTo #
  - [Ext] setFeeToSetter #

+ [Int] IUniswapV2Pair
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transfer #
  - [Ext] transferFrom #
  - [Ext] DOMAIN_SEPARATOR
  - [Ext] PERMIT_TYPEHASH
  - [Ext] nonces
  - [Ext] permit #
  - [Ext] MINIMUM_LIQUIDITY
  - [Ext] factory
  - [Ext] token0
  - [Ext] token1
  - [Ext] getReserves
  - [Ext] price0CumulativeLast
  - [Ext] price1CumulativeLast
  - [Ext] kLast
  - [Ext] burn #
  - [Ext] swap #
  - [Ext] skim #
  - [Ext] sync #
  - [Ext] initialize #

+ [Int] IUniswapV2Router01
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidity #
  - [Ext] addLiquidityETH ($)
  - [Ext] removeLiquidity #
  - [Ext] removeLiquidityETH #
  - [Ext] removeLiquidityWithPermit #
  - [Ext] removeLiquidityETHWithPermit #
  - [Ext] swapExactTokensForTokens #
  - [Ext] swapTokensForExactTokens #
  - [Ext] swapExactETHForTokens ($)
  - [Ext] swapTokensForExactETH #
  - [Ext] swapExactTokensForETH #
  - [Ext] swapETHForExactTokens ($)
  - [Ext] quote
  - [Ext] getAmountOut
  - [Ext] getAmountIn
  - [Ext] getAmountsOut
  - [Ext] getAmountsIn

+ [Int] IUniswapV2Router02 (IUniswapV2Router01)
  - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
  - [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #

- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
- [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
- [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ Kingwarrior (Context, IERC20, Ownable)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Pub] isExcludedFromReward
  - [Pub] totalFees
  - [Pub] minimumTokensBeforeSwapAmount
  - [Pub] buyBackUpperLimitAmount
  - [Pub] deliver #
  - [Pub] reflectionFromToken
  - [Pub] tokenFromReflection
  - [Pub] excludeFromReward #
    - modifiers: onlyOwner
  - [Ext] includeInReward #
    - modifiers: onlyOwner
  - [Prv] _approve #
  - [Prv] _transfer #
  - [Prv] swapTokens #
    - modifiers: lockTheSwap
  - [Prv] buyBackTokens #
    - modifiers: lockTheSwap
  - [Prv] swapTokensForEth #
  - [Prv] swapETHForTokens #
  - [Prv] addLiquidity #
  - [Prv] _tokenTransfer #
  - [Prv] _transferStandard #
  - [Prv] _transferToExcluded #
  - [Prv] _transferFromExcluded #
  - [Prv] _transferBothExcluded #
  - [Prv] _reflectFee #
  - [Prv] _getValues
  - [Prv] _getTValues
  - [Prv] _getRValues
  - [Prv] _getRate
  - [Prv] _getCurrentSupply
  - [Prv] _takeLiquidity #
  - [Prv] calculateTaxFee
  - [Prv] calculateLiquidityFee
  - [Prv] removeAllFee #
  - [Prv] restoreAllFee #
  - [Pub] isExcludedFromFee
  - [Pub] excludeFromFee #

  - modifiers: onlyOwner
 - [Pub] includeInFee #
  - modifiers: onlyOwner
 - [Ext] setTaxFeePercent #
  - modifiers: onlyOwner
 - [Ext] setLiquidityFeePercent #
  - modifiers: onlyOwner
 - [Ext] setMaxTxAmount #
  - modifiers: onlyOwner
 - [Ext] setMarketingDivisor #
  - modifiers: onlyOwner
 - [Ext] setNumTokensSellToAddToLiquidity #
  - modifiers: onlyOwner
 - [Ext] setBuybackUpperLimit #
  - modifiers: onlyOwner
 - [Ext] setMarketingAddress #
  - modifiers: onlyOwner
 - [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
 - [Pub] setBuyBackEnabled #
  - modifiers: onlyOwner
 - [Ext] prepareForPreSale #
  - modifiers: onlyOwner
 - [Ext] afterPreSale #
  - modifiers: onlyOwner
 - [Prv] transferToAddressETH #
 - [Ext] <Fallback> ($)

($) = payable function
# = non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Low issues
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user data leaks.	Passed
14.	Malicious Event log.	Passed
15.	Scoping and Declarations.	Passed
16.	Uninitialized storage pointers.	Passed
17.	Arithmetic accuracy.	Passed
18.	Design Logic.	Passed
19.	Cross-function race conditions.	Passed
20.	Safe Open Zeppelin contracts implementation and usage.	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

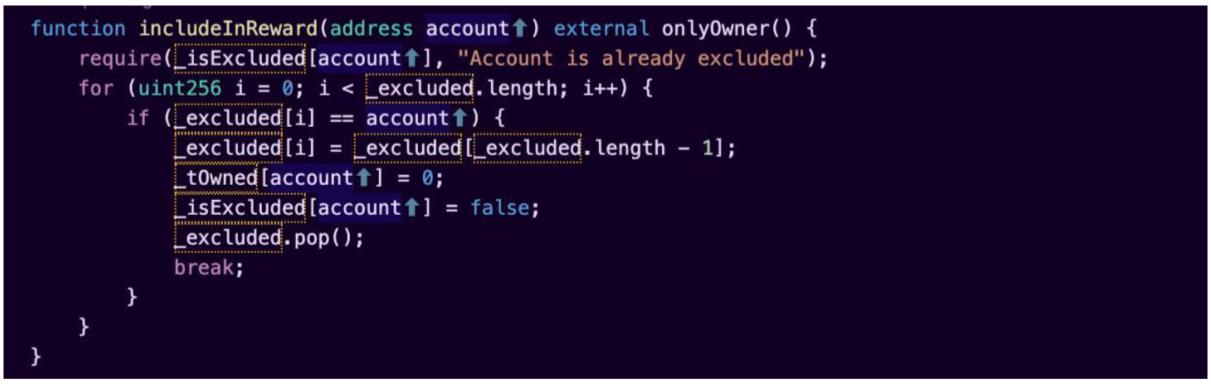
No medium severity issues found.

Low Severity Issues

1. Out of gas

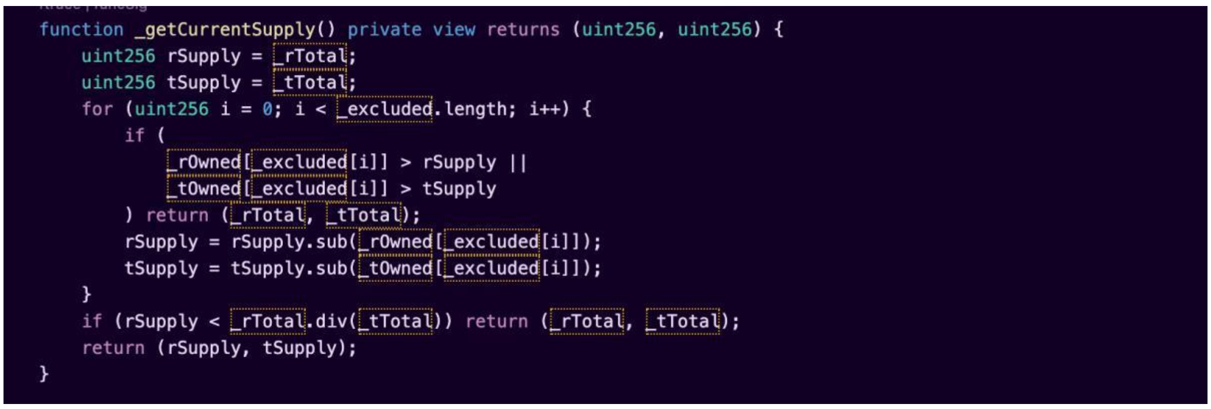
Issue:

- The function includeInReward() uses the loop to find and remove addresses from the _excluded list. Function will be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.



```
function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The function _getCurrentSupply also uses the loop for evaluating total supply. It also could be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.



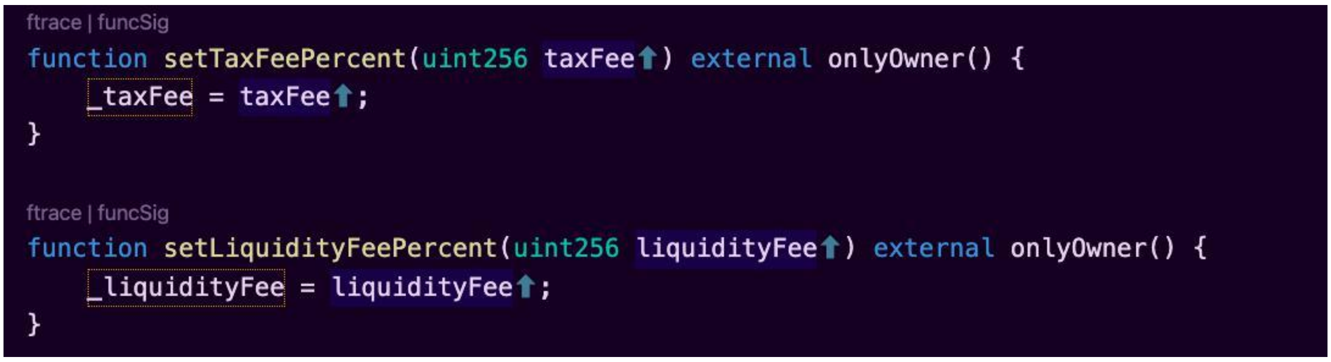
```
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _rOwned[_excluded[i]] > rSupply ||
            _tOwned[_excluded[i]] > tSupply
        ) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

Recommendation:

Check that the excluded array length is not too big.

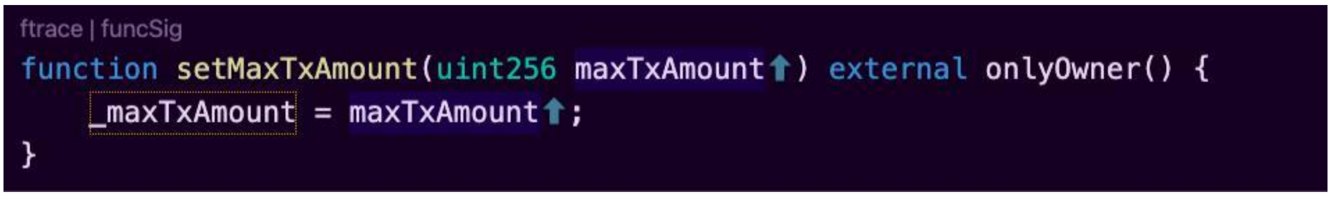
Owner privileges (In the period when the owner is not renounced)

- **Owner can change tax and liquidity fees.**



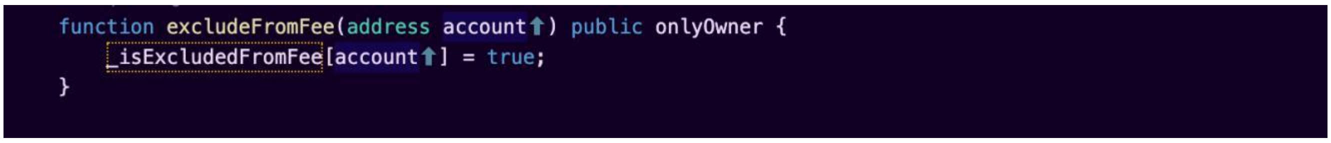
```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}

ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
}
```

- **Owner can change maximum transaction amount.**



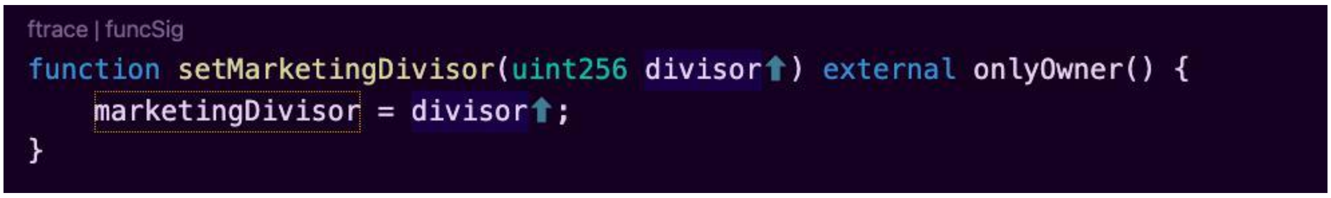
```
ftrace | funcSig
function setMaxTxAmount(uint256 maxTxAmount) external onlyOwner() {
    _maxTxAmount = maxTxAmount;
}
```

- **Owner can exclude from the fee.**



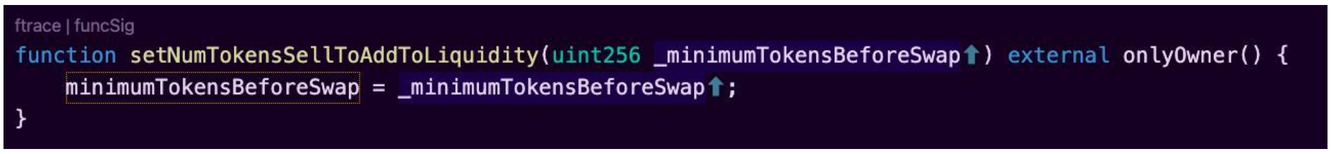
```
    function excludeFromFee(address account) public onlyOwner {
        _isExcludedFromFee[account] = true;
    }
```

- **Owner can change marketingDivisor.**



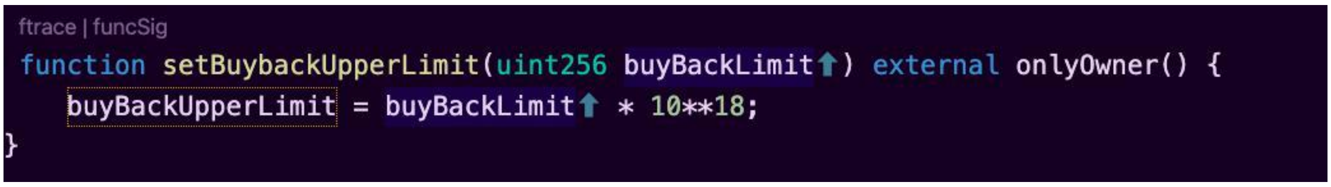
```
ftrace | funcSig
function setMarketingDivisor(uint256 divisor) external onlyOwner() {
    marketingDivisor = divisor;
}
```

- **Owner can change minimum number of tokens to add to liquidity.**



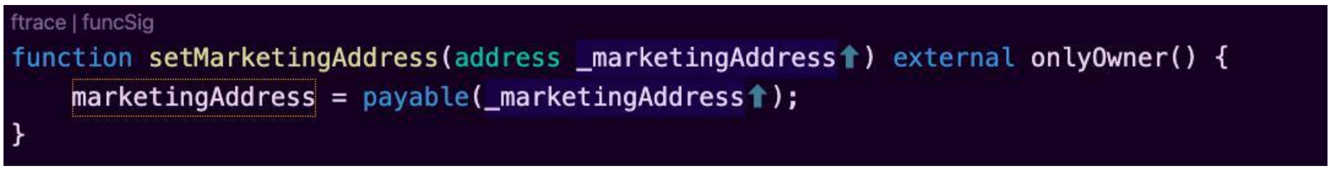
```
ftrace | funcSig
function setNumTokensSellToAddToLiquidity(uint256 _minimumTokensBeforeSwap) external onlyOwner() {
    minimumTokensBeforeSwap = _minimumTokensBeforeSwap;
}
```

- Owner can change buyBackUpperLimit.



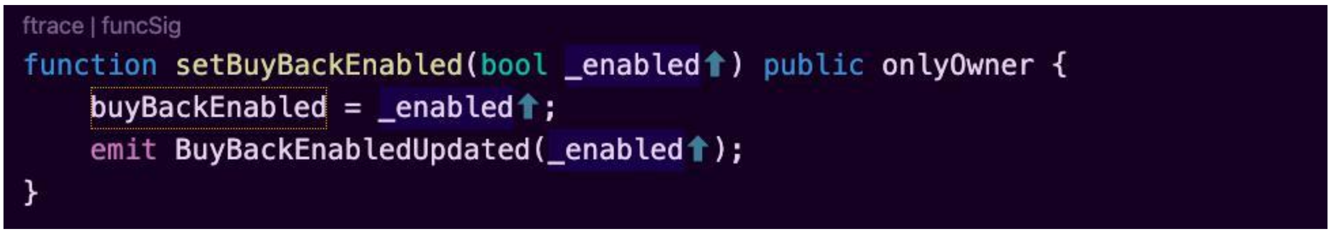
```
ftrace | funcSig
function setBuybackUpperLimit(uint256 buyBackLimit) external onlyOwner() {
    buyBackUpperLimit = buyBackLimit * 10**18;
}
```

- Owner can change marketing address.



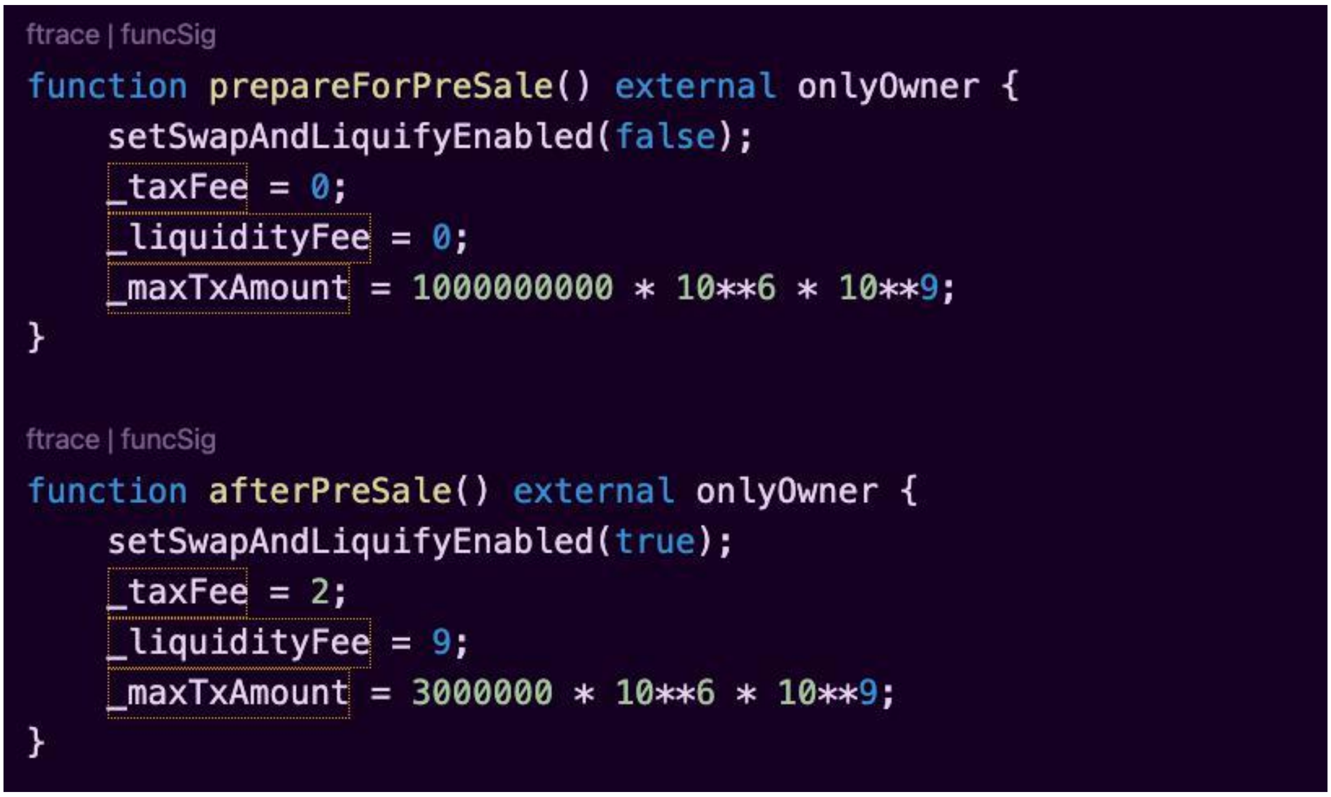
```
ftrace | funcSig
function setMarketingAddress(address _marketingAddress) external onlyOwner() {
    marketingAddress = payable(_marketingAddress);
}
```

- Owner can enable and disable buyBack.



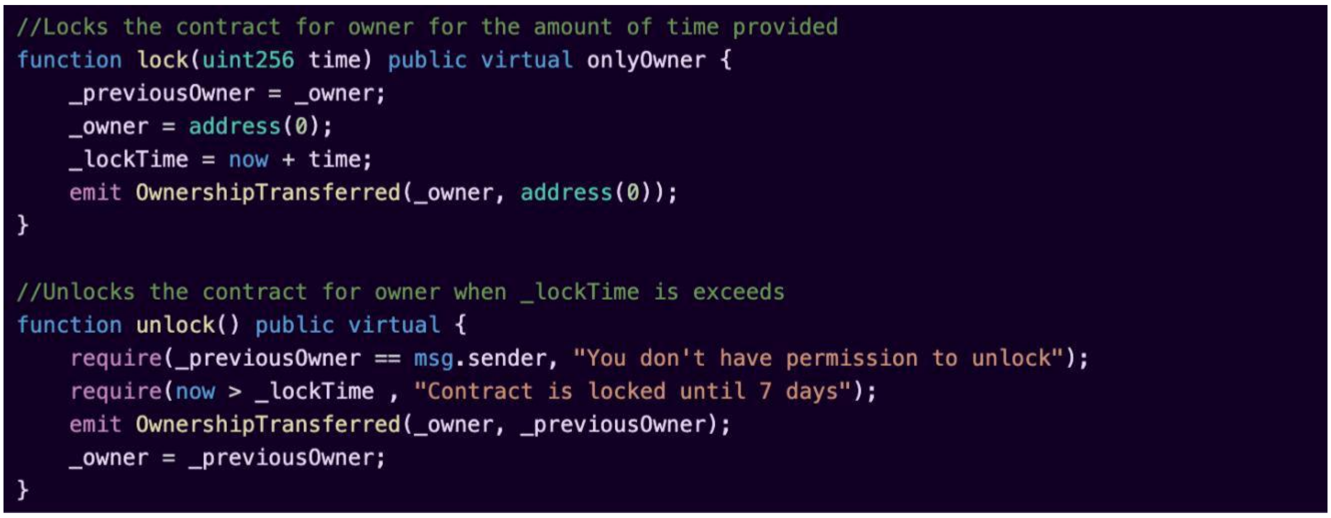
```
ftrace | funcSig
function setBuyBackEnabled(bool _enabled) public onlyOwner {
    buyBackEnabled = _enabled;
    emit BuyBackEnabledUpdated(_enabled);
}
```

- Owner can enable before and after presale modes.



```
ftrace | funcSig
function prepareForPreSale() external onlyOwner {
    setSwapAndLiquifyEnabled(false);
    _taxFee = 0;
    _liquidityFee = 0;
    _maxTxAmount = 1000000000 * 10**6 * 10**9;
}

ftrace | funcSig
function afterPreSale() external onlyOwner {
    setSwapAndLiquifyEnabled(true);
    _taxFee = 2;
    _liquidityFee = 9;
    _maxTxAmount = 3000000 * 10**6 * 10**9;
}
```

- **Owner can lock and unlock. By the way, using these functions the owner could retake privileges even after the ownership was renounced.**



```
//Locks the contract for owner for the amount of time provided
function lock(uint256 time) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = now + time;
    emit OwnershipTransferred(_owner, address(0));
}

//Unlocks the contract for owner when _lockTime is exceeds
function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(now > _lockTime , "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope. One third of the liquidity goes to marketing address.

Liquidity locking details NOT provided by the team.

Auditek note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.