Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

August, 2021

Audit Details

Audited project

DAIKOKUTEN SAMA

Deployer address

0xC904988723de486C38be9d8FcD82710eaE6267A7

Client contacts:

DAIKOKUTEN SAMA team

Blockchain

Binance Smart Chain

Project website:

https://daikokuten.finance/

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn’t say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

TechRate was commissioned by DAIKOKUTEN SAMA to perform an audit of smart contracts:
https://bscscan.com/address/0x834613c64522725b23b458af04ed1590d189962f#code

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.

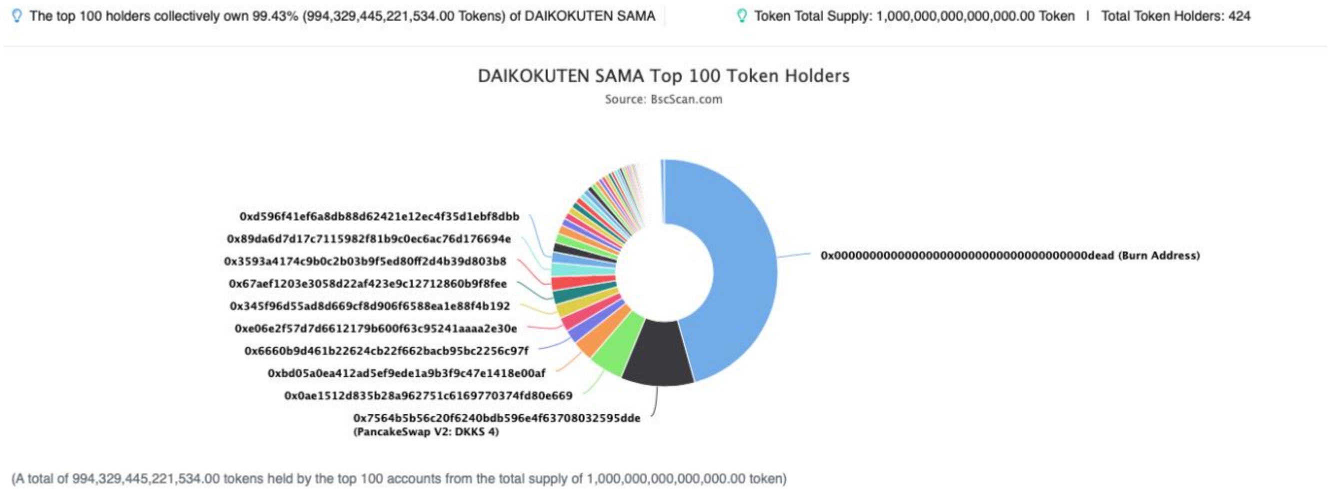
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 16.08.2021

Contract name	DKKS
Contract address	0x834613c64522725b23b458aF04ED1590D189962F
Total supply	1,000,000,000,000,000
Token ticker	DKKS
Decimals	9
Token holders	424
Transactions count	1,617
Top 100 holders dominance	99.43%
Liquidity fee	7
Tax fee	3
Total fees	15154526142171029294628
Uniswap V2 pair	0x7564b5b56c20f6240bdb596e4f63708032595dde
Contract deployer address	0xC904988723de486C38be9d8FcD82710eaE6267A7
Contract's current owner address	0xC904988723de486C38be9d8FcD82710eaE6267A7

DAIKOKUTEN SAMA Token Distribution



The top 100 holders collectively own 99.43% (994,329,445,221,534.00 Tokens) of DAIKOKUTEN SAMA

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 424

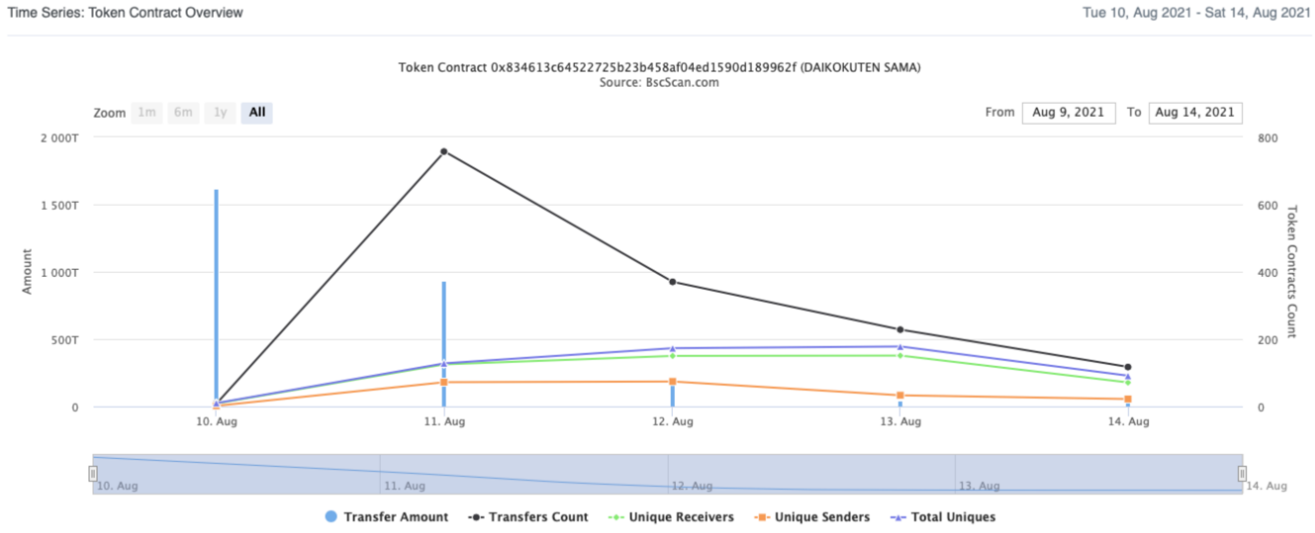
DAIKOKUTEN SAMA Top 100 Token Holders

Source: BscScan.com

0xd596f41ef6a8db88d62421e12ec4f35d1ebf8dbb
0x89da6d7d17c7115982f81b9c0ec6ac76d176694e
0x3593a4174c9b0c2b03b9f5ed80ff2d4b39d803b8
0x67aef1203e3058d22af423e9c12712860b9f8fee
0x345f96d55ad8d669cf8d906f6588ea1e88f4b192
0xe06e2f57d7d6612179b600f63c95241aaaa2e30e
0x6660b9d461b22624cb22f662bacb95bc2256c97f
0xbd05a0ea412ad5ef9ede1a9b3f9c47e1418e00af
0x0ae1512d835b28a962751c6169770374fd80e669
0x7564b5b56c20f6240bdb596e4f63708032595dde (PancakeSwap V2: DKKS 4)
0x000000000000000000000000000000000000dead (Burn Address)

(A total of 994,329,445,221,534.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

DAIKOKUTEN SAMA Contract Interaction Details



Time Series: Token Contract Overview

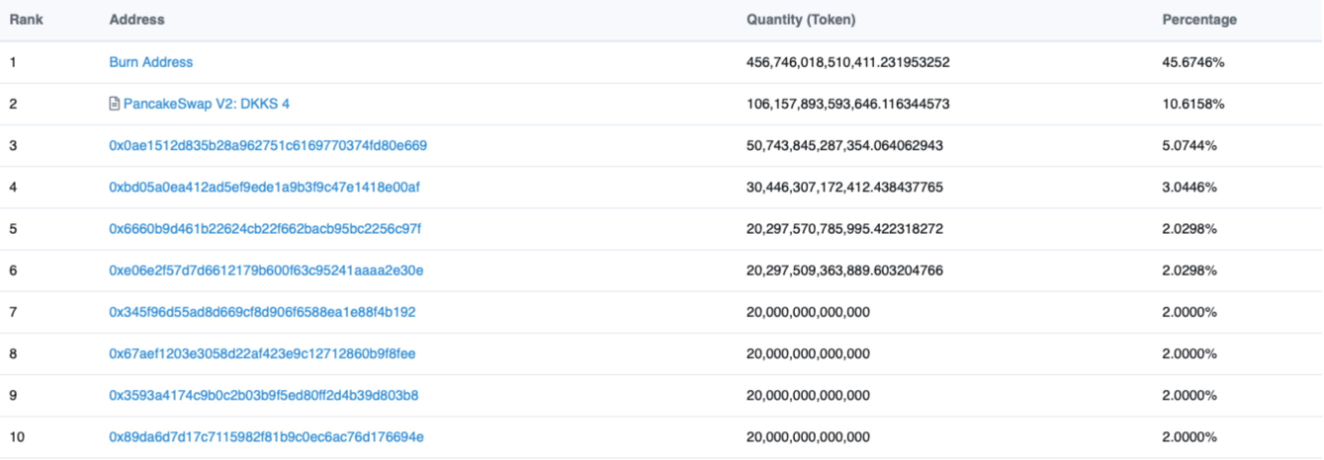
Tue 10, Aug 2021 - Sat 14, Aug 2021

Token Contract 0x834613c64522725b23b458af04ed1590d189962f (DAIKOKUTEN SAMA)

Source: BscScan.com

From Aug 9, 2021 To Aug 14, 2021

DAIKOKUTEN SAMA Top 10 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	Burn Address	456,746,018,510,411.231953252	45.6746%
2	🗎 PancakeSwap V2: DKKS 4	106,157,893,593,646.116344573	10.6158%
3	0x0ae1512d835b28a962751c6169770374fd80e669	50,743,845,287,354.064062943	5.0744%
4	0xbd05a0ea412ad5ef9ede1a9b3f9c47e1418e00af	30,446,307,172,412.438437765	3.0446%
5	0x6660b9d461b22624cb22f662bacb95bc2256c97f	20,297,570,785,995.422318272	2.0298%
6	0xe06e2f57d7d6612179b600f63c95241aaaa2e30e	20,297,509,363,889.603204766	2.0298%
7	0x345f96d55ad8d669cf8d906f6588ea1e88f4b192	20,000,000,000,000	2.0000%
8	0x67aef1203e3058d22af423e9c12712860b9f8fee	20,000,000,000,000	2.0000%
9	0x3593a4174c9b0c2b03b9f5ed80ff2d4b39d803b8	20,000,000,000,000	2.0000%
10	0x89da6d7d17c7115982f81b9c0ec6ac76d176694e	20,000,000,000,000	2.0000%

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 #

+ DKKS (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] deliver #
  - [Pub] reflectionFromToken
  - [Pub] tokenFromReflection
  - [Pub] excludeFromReward #
    - modifiers: onlyOwner
  - [Ext] includeInReward #
    - modifiers: onlyOwner
  - [Prv] _approve #
  - [Prv] _transfer #
  - [Prv] swapTokens #
    - 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] setBuyTaxFeePercent #
  - modifiers: onlyOwner
- [Ext] setSellTaxFeePercent #
  - modifiers: onlyOwner
- [Ext] setBuyLiquidityFeePercent #
  - modifiers: onlyOwner
- [Ext] setSellLiquidityFeePercent #
  - modifiers: onlyOwner
- [Ext] setMaxTxAmount #
  - modifiers: onlyOwner
- [Ext] setMarketingDivisor #
  - modifiers: onlyOwner
- [Ext] setNumTokensSellToAddToLiquidity #
  - modifiers: onlyOwner
- [Ext] setMaxTokenHolder #
  - modifiers: onlyOwner
- [Ext] setMarketingAddress #
  - modifiers: onlyOwner
- [Pub] changeRouterVersion #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Ext] prepareForPreSale #
  - modifiers: onlyOwner
- [Ext] goLive #
  - modifiers: onlyOwner
- [Pub] transferBatch #
- [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.	Passed
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

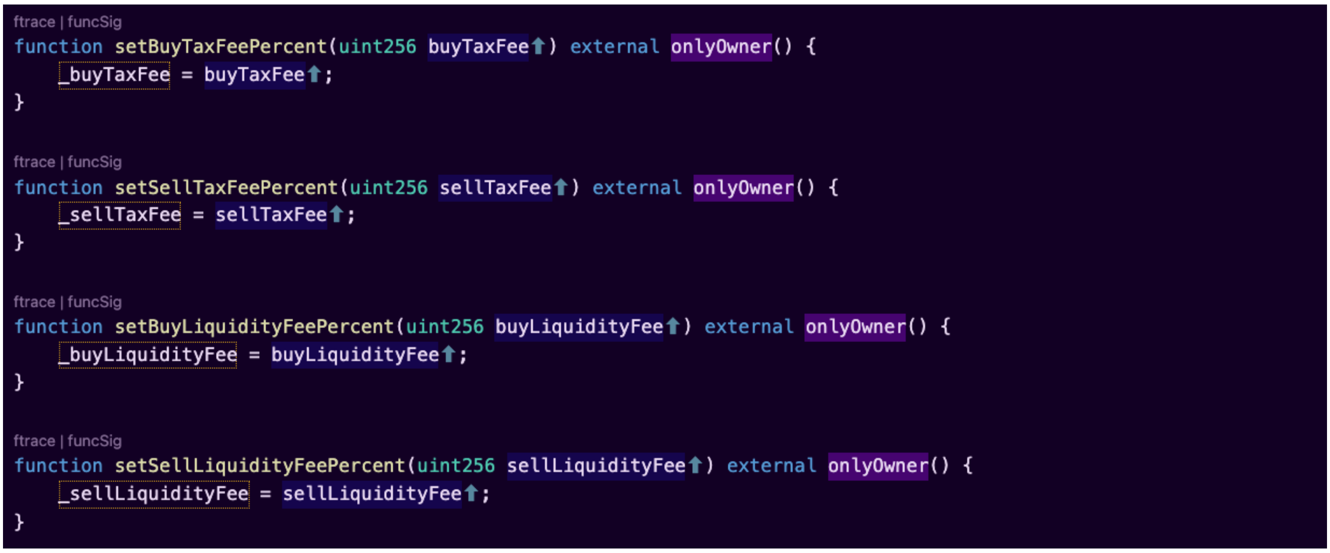
No medium severity issues found.

Low Severity Issues

No low severity issues found.

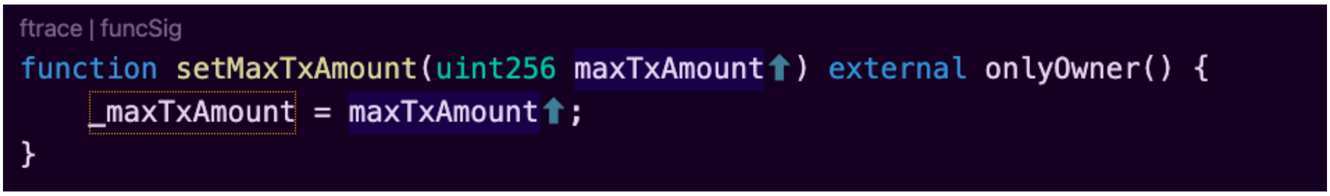
Owner privileges (In the period when the owner is not renounced)

- Owner can change buy/sell tax and liquidity fees.



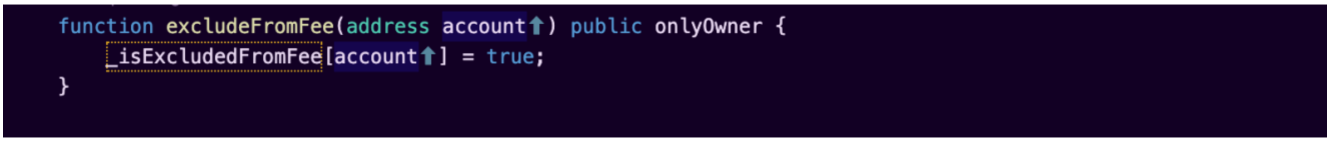
```
ftrace | funcSig
function setBuyTaxFeePercent(uint256 buyTaxFee⬆) external onlyOwner() {
    _buyTaxFee = buyTaxFee⬆;
}

ftrace | funcSig
function setSellTaxFeePercent(uint256 sellTaxFee⬆) external onlyOwner() {
    _sellTaxFee = sellTaxFee⬆;
}

ftrace | funcSig
function setBuyLiquidityFeePercent(uint256 buyLiquidityFee⬆) external onlyOwner() {
    _buyLiquidityFee = buyLiquidityFee⬆;
}

ftrace | funcSig
function setSellLiquidityFeePercent(uint256 sellLiquidityFee⬆) external onlyOwner() {
    _sellLiquidityFee = sellLiquidityFee⬆;
}
```

- Owner can change maximum transaction amount.



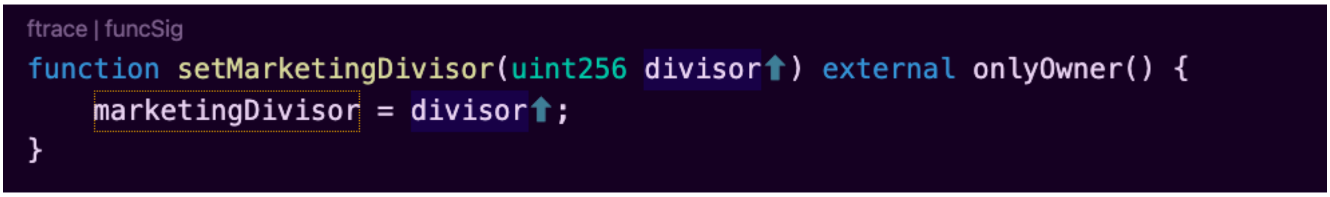
```
ftrace | funcSig
function setMaxTxAmount(uint256 maxTxAmount⬆) external onlyOwner() {
    _maxTxAmount = maxTxAmount⬆;
}
```

- Owner can exclude from the fee.



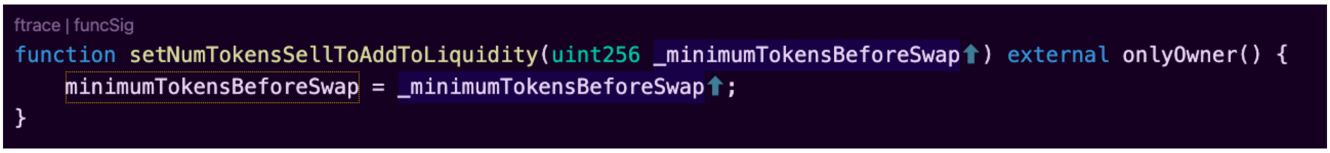
```
    function excludeFromFee(address account) public onlyOwner {
        _isExcludedFromFee[account] = true;
    }
```

- Owner can change marketingDivisor.



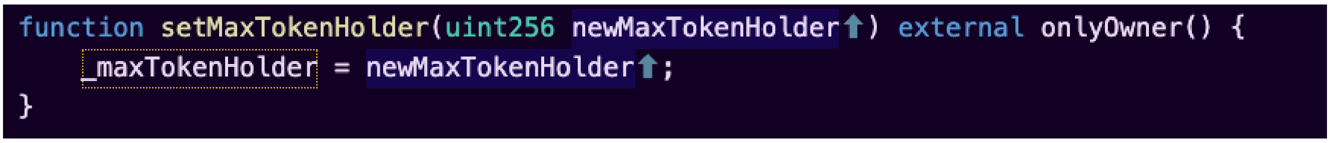
```
ftrace | funcSig
function setMarketingDivisor(uint256 divisor) external onlyOwner() {
    marketingDivisor = divisor;
}
```

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



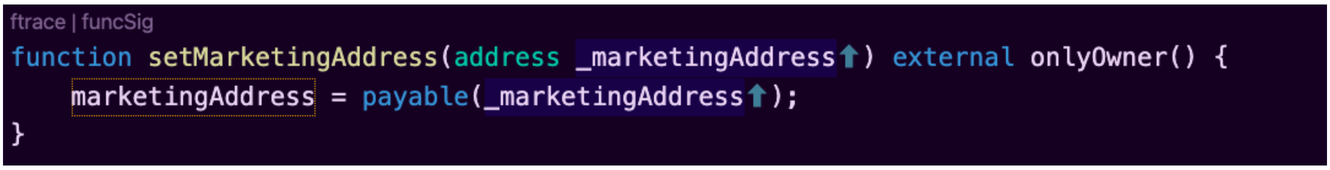
```
ftrace | funcSig
function setNumTokensSellToAddToLiquidity(uint256 _minimumTokensBeforeSwap) external onlyOwner() {
    minimumTokensBeforeSwap = _minimumTokensBeforeSwap;
}
```

- Owner can change _maxTokenHolder value.



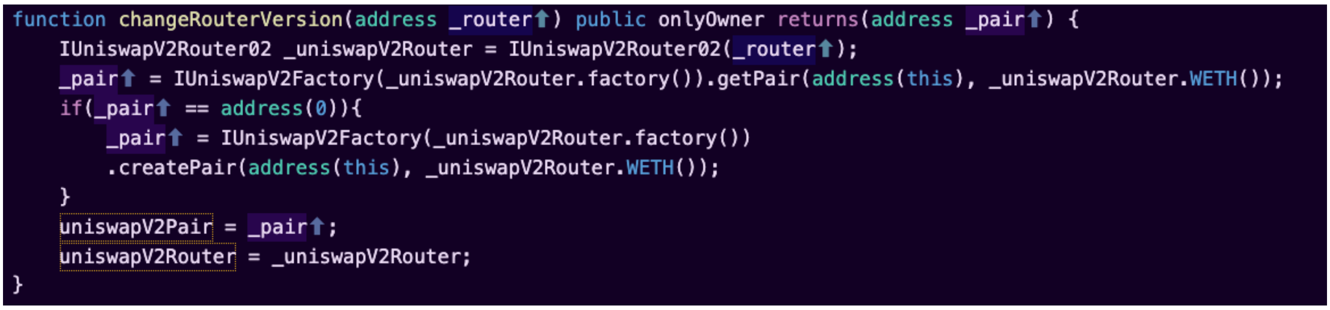
```
function setMaxTokenHolder(uint256 newMaxTokenHolder) external onlyOwner() {
    _maxTokenHolder = newMaxTokenHolder;
}
```

- Owner can change marketing address.



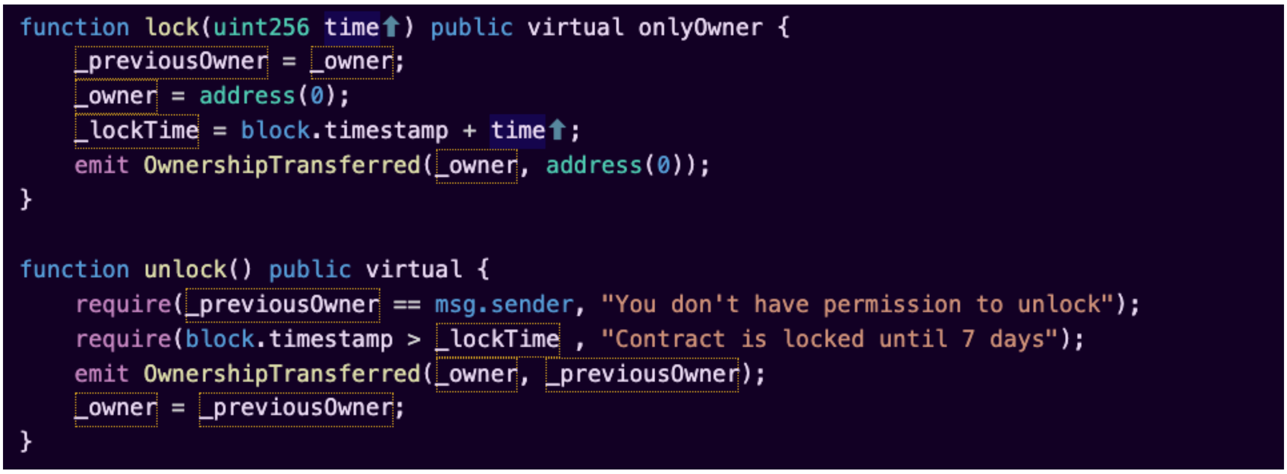
```
ftrace | funcSig
function setMarketingAddress(address _marketingAddress) external onlyOwner() {
    marketingAddress = payable(_marketingAddress);
}
```

- Owner can change router address.



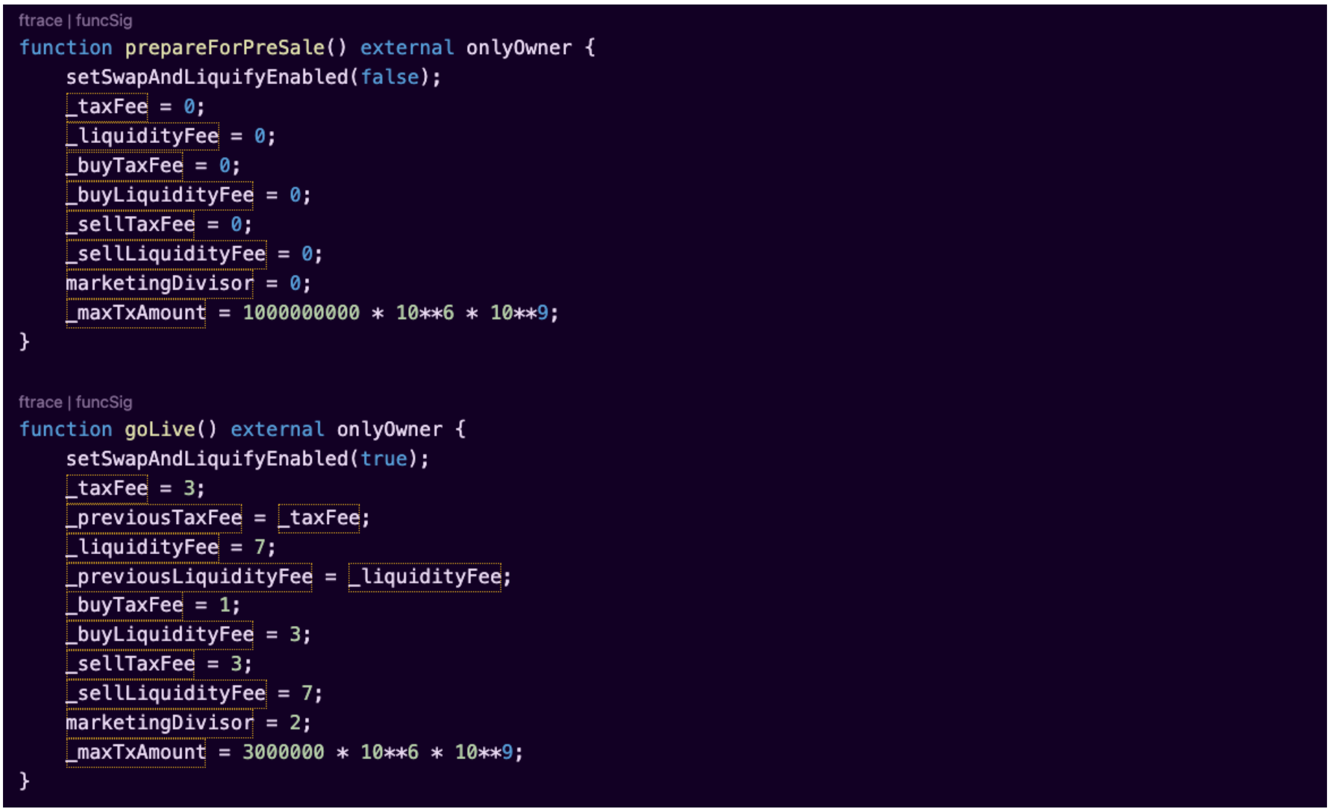
```
function changeRouterVersion(address _router) public onlyOwner returns(address _pair) {
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(_router);
    _pair = IUniswapV2Factory(_uniswapV2Router.factory()).getPair(address(this), _uniswapV2Router.WETH());
    if(_pair == address(0)){
        _pair = IUniswapV2Factory(_uniswapV2Router.factory())
        .createPair(address(this), _uniswapV2Router.WETH());
    }
    uniswapV2Pair = _pair;
    uniswapV2Router = _uniswapV2Router;
}
```

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



```
function lock(uint256 time⬆) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = block.timestamp + time⬆;
    emit OwnershipTransferred(_owner, address(0));
}

function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(block.timestamp > _lockTime , "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

- **Owner can enable presale and live setting presets.**



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

ftrace | funcSig
function goLive() external onlyOwner {
    setSwapAndLiquifyEnabled(true);
    _taxFee = 3;
    _previousTaxFee = _taxFee;
    _liquidityFee = 7;
    _previousLiquidityFee = _liquidityFee;
    _buyTaxFee = 1;
    _buyLiquidityFee = 3;
    _sellTaxFee = 3;
    _sellLiquidityFee = 7;
    marketingDivisor = 2;
    _maxTxAmount = 3000000 * 10**6 * 10**9;
}
```

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope. The further transfers and operations with the funds raise are not related to this particular contract.

Liquidity locking details provided by the team:
https://dxsale.app/app/v2_9/dxlockview?id=0&add=0xC904988723de486C38be9d8FcD82710eaE6267A7&type=lplock&chain=BSC

TechRate 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.