



Audit Report

WEN LAMBO

Token Name / Symbol / Decimal :

WEN LAMBO / (LAMBO) / 9

Percentage Collected - Tax / Charity / Burn :

2% / 4% / 4%

Deployer Address :

0x357B174d3690998845c0A5D3B2762E8c600BB814

Owner Address :

0x7d37255545f7e5e6e68e19dc9291e2e54b72a29b

KubixSquare was asked to perform an Audit by Wen Lambo for the Smart Contract from the following address:

<https://bscscan.com/address/0x2c7b396d17e3a5184d4901380836de7a72c5cba4#code>



The audit objective was introduced in order to achieve the following objectives:

- * Identify possible security issues using smart contracts.

The information contained in this report should be used to gain an understanding of the impact on risks associated with a smart contract, and as a guide to improve the security status of a smart contract by addressing identified issues.

Contract Description Table :

Contract	Type	Bases			
:-----: :-----: :-----: :-----: :-----:					
L	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
Context Implementation					
L	_msgSender	Internal			
L	_msgData	Internal			
IBEP20 Interface					
L	totalSupply	External	!	INO !	
L	balanceOf	External	!	INO !	
L	transfer	External	!	INO !	
L	allowance	External	!	INO !	
L	approve	External	!	INO !	
L	transferFrom	External	!	INO !	
SafeMath Library					
L	add	Internal			
L	sub	Internal			
L	sub	Internal			
L	mul	Internal			
L	div	Internal			
L	div	Internal			
L	mod	Internal			
L	mod	Internal			
Address Library					
L	isContract	Internal			
L	sendValue	Internal			
L	functionCall	Internal			
L	functionCall	Internal			
L	functionCallWithValue	Internal			
L	functionCallWithValue	Internal			
L	_functionCallWithValue	Private			

|||||

| ****Ownable**** | Implementation | Context |||





























	owner	Public !	INO !	
	renounceOwnership	Public !	🔴	onlyOwner
	transferOwnership	Public !	🔴	onlyOwner

|||||

| ****CoinToken**** | Implementation | Context, IBEP20, Ownable |||



	<Constructor>	Public !	🔴 INO !	
	name	Public !	INO !	
	symbol	Public !	INO !	
	decimals	Public !	INO !	
	totalSupply	Public !	INO !	
	balanceOf	Public !	INO !	
	transfer	Public !	🔴 INO !	
	allowance	Public !	INO !	
	approve	Public !	🔴 INO !	
	transferFrom	Public !	🔴 INO !	
	increaseAllowance	Public !	🔴 INO !	
	decreaseAllowance	Public !	🔴 INO !	
	isExcluded	Public !	INO !	
	isCharity	Public !	INO !	
	totalFees	Public !	INO !	
	totalBurn	Public !	INO !	
	totalCharity	Public !	INO !	
	deliver	Public !	🔴 INO !	
	reflectionFromToken	Public !	INO !	
	tokenFromReflection	Public !	INO !	
	excludeAccount	External !	🔴	onlyOwner
	includeAccount	External !	🔴	onlyOwner
	setAsCharityAccount	External !	🔴	onlyOwner
	burn	Public !	🔴 INO !	
	updateFee	Public !	🔴	onlyOwner
	_burn	Internal 🔒	🔴	
	mint	Public !	🔴	onlyOwner
	_approve	Private 🔒	🔴	
	_transfer	Private 🔒	🔴	
	_transferStandard	Private 🔒	🔴	
	_standardTransferContent	Private 🔒	🔴	

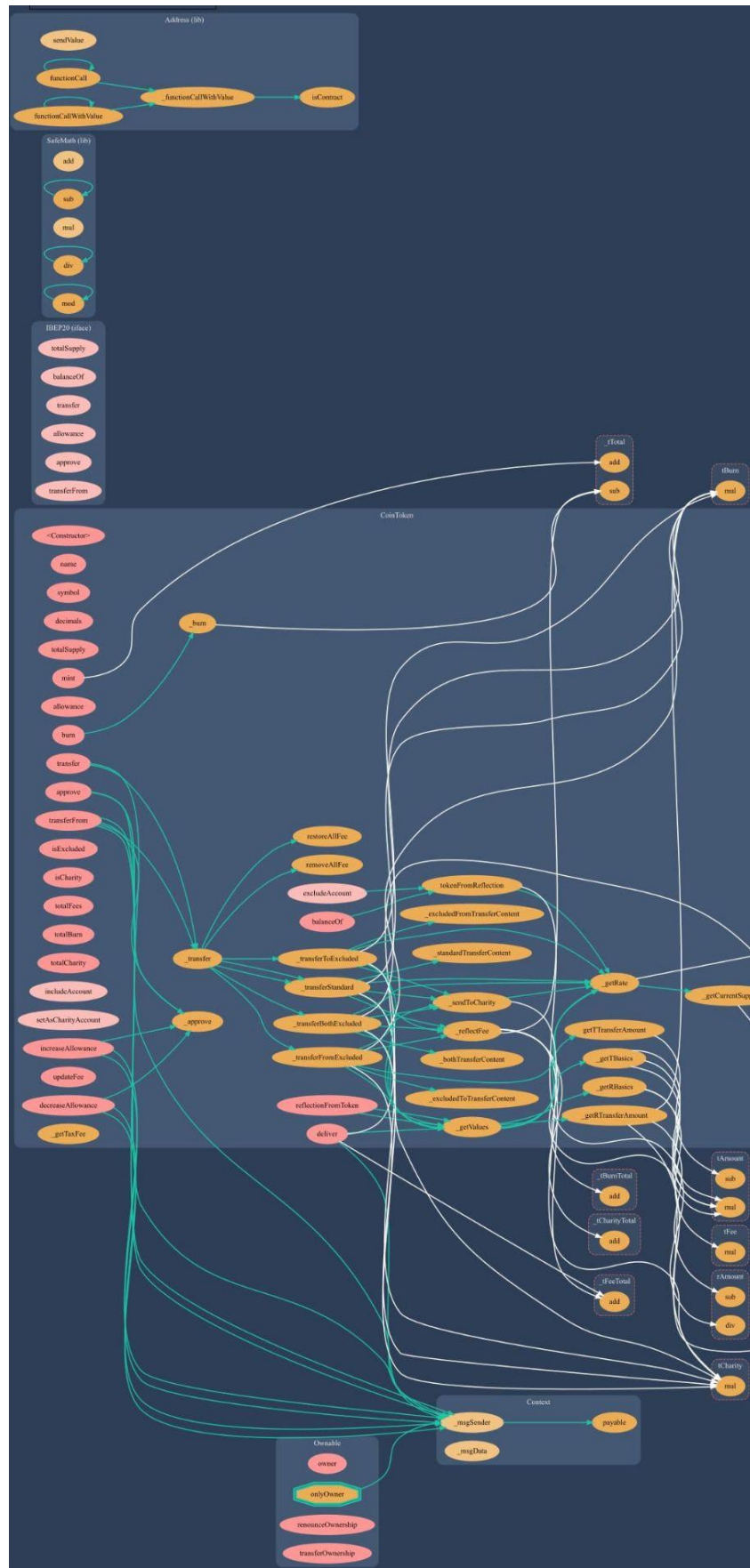
```

| | | _transferToExcluded | Private |  |  | |
| | | _excludedFromTransferContent | Private |  |  | |
| | | _transferFromExcluded | Private |  |  | |
| | | _excludedToTransferContent | Private |  |  | |
| | | _transferBothExcluded | Private |  |  | |
| | | _bothTransferContent | Private |  |  | |
| | | _reflectFee | Private |  |  | |
| | | _getValues | Private |  | | |
| | | _getTBasics | Private |  | | |
| | | getTTransferAmount | Private |  | | |
| | | _getRBasics | Private |  | | |
| | | _getRTransferAmount | Private |  | | |
| | | _getRate | Private |  | | |
| | | _getCurrentSupply | Private |  | | |
| | | _sendToCharity | Private |  |  | |
| | | removeAllFee | Private |  |  | |
| | | restoreAllFee | Private |  |  | |
| | | _getTaxFee | Private |  | | |

```

Legend

Symbol	Meaning
:-----: -----	
	Function can modify state
	Function is payable



Issues Check Status

- Compiler errors. ✓
- Race conditions and Reentrancy. Cross-function race conditions. ✓
- Possible delays in data delivery. ✓
- Oracle calls. ✓
- Front running. ✓
- Timestamp dependence. ✓
- Integer Overflow and Underflow. ✓
- DoS with Revert. ✓
- DoS with block gas limit. ✓
- Methods execution permissions. ✓
- The impact of the exchange rate on logic. ✓
- Private user data leaks. ✓
- Malicious Event log. ✓
- Scoping and Declarations. ✓
- Uninitialized storage pointers. ✓
- Arithmetic accuracy. ✓
- Design Logic. ✓
- Cross-function race conditions. ✓
- Safe Open Zeppelin contracts implementation and usage. ✓
- Fallback function security. ✓

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


- 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) {
    ftrace | funcSig
    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]]);
        ftrace | funcSig
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
    ftrace | funcSig
}
```

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 charity address.

```
function setAsCharityAccount(address account) external onlyOwner() {
    require(!_isCharity[account], "Account is already charity account");
    _isCharity[account] = true;
    FeeAddress = account;
}
```

- Owner can mint.

```
function mint(address account, uint256 amount) onlyOwner() public {

    _tTotal = _tTotal.add(amount);
    _rOwned[account] = _rOwned[account].add(amount);
    emit Transfer(address(0), account, amount);
}
```

- Owner can change fees.

```
function updateFee(uint256 _txFee,uint256 _burnFee,uint256 _charityFee) onlyOwner() public{
    _TAX_FEE = _txFee* 100;
    _BURN_FEE = _burnFee * 100;
    _CHARITY_FEE = _charityFee* 100;
    ORIG_TAX_FEE = _TAX_FEE;
    ORIG_BURN_FEE = _BURN_FEE;
    ORIG_CHARITY_FEE = _CHARITY_FEE;
}
```



Conclusion :

Smart contracts do not contain high and medium severity issues. Liquidity locking details were not provided by the team and Liquidity pair contract's security is not checked due to out of scope.

NOTE:

Please check the disclaimer below 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.

Disclaimer :

This is a short report on our findings, in our analysis and in accordance with best practices, such as as of the date of this report, regarding cybersecurity vulnerabilities and issues in context, and algorithms based on smart contracts, details that have been detailed in this report. To get a complete picture of our analysis, it is very important to read the full report. We have made every effort to conduct this analysis and prepare a report, it is important to remember that you do not need to refer to this report, and cannot have any complaints against us based on what was said or written, or how it was prepared, and it is important for you to conduct your own independent research before making any decisions. We will discuss this in more detail, and you can find the next nuance listed below in full.

DISCLAIMER: By reading this report or any part of it, you agree to be bound by the terms of this disclaimer at any time. If you do not agree to all the terms and conditions, you must immediately stop reading this report and delete and destroy all copies of the report that will be downloaded and / or printed for you. This report is intended for informational purposes and based on lack of trust, and is not considered investment advice. Neither party may link to the report or its contents, and KubixSquare and its subsidiaries (including financial holding companies, shareholders, subsidiaries, officers, directors, board members, and other members of this agreement) (KubixSquare) are under no obligation to care for you or any other person, and KubixSquare makes no warranties or representations of any person as to the accuracy or completeness of the report's contents. The report is provided on an "as is" basis, without any terms, warranties, or any other conditions other than the provisions of these terms of use, and KubixSquare does not exclude any representations, warranties, conditions, or other conditions (including, but not limited to, implied warranties of satisfactory quality, fitness for a particular purpose, and use of reasonable care and skill) that, if not for this element, may apply to the report. Except and only to the extent prohibited by law, KubixSquare disclaims all liability and liability, and neither you nor any other person shall have any claim against KubixSquare for any amount or type of loss or damage suffered by you or any other person (including, without limitation, any direct, indirect, incidental, special, consequential, special, punitive, consequential or purely economic loss or damage, as well as loss of revenue, revenue, profits, reputation, data, contracts, use of money or interruption of business, or instagram (including without limitation, negligence), breach of contract, breach of statutory duty, misrepresentation (whether innocent or guilty) or otherwise for the purpose of any claim of any kind, in any jurisdiction) in any way arising out of or in connection with this report, as well as from the use, inability to use or results of use of this report, and with any reliance on this report.

Security analysis is based solely on smart contracts. It doesn't have any apps or activities that have been evaluated for security reasons. This is not the product code that was changed.

Visit - <https://kubixsquare.com>

Github - <https://github.com/kubixsquare>