

SKELETON ECOSYSTEM

SMART CONTRACT AUDIT



Centralium
\$CTM
BEP 20

0xD5a6e81d339ABe754B9754b2C16b2Be37C50B407



Table of Contents

Table of Contents	1
Disclaimer	2
Overview	3
Creation/Audit Date	3
Verified Socials	3
Contract Functions Analysis	4
Contract Safety and Weakness	7
Detected Vulnerability Description	11
Contract Flow Graph	14
Contract Interaction Graph	15
Inheritance Graph	16
Contract Descriptions	17
Audit Scope	25

Global Disclaimer

This document serves as a disclaimer for the crypto smart contract audit conducted by Skeleton Ecosystem. The purpose of the audit was to review the codebase of the smart contracts for potential vulnerabilities and issues. It is important to note the following:

Limited Scope: The audit is based on the code and information available up to the audit completion date. It does not cover external factors, system interactions, or changes made after the audit. The audit itself can not guarantee 100% safety and can not detect common scam methods like farming and developer sell-out.

No Guarantee of Security: While we have taken reasonable steps to identify vulnerabilities, it is impossible to guarantee the complete absence of security risks or issues. The audit report provides an assessment of the contract's security as of the audit date.

Continued Development: Smart contracts and blockchain technology are evolving fields. Updates, forks, or changes to the contract post-audit may introduce new risks that were not present during the audit.

Third-party Code: If the smart contract relies on third-party libraries or code, those components were not thoroughly audited unless explicitly stated. Security of these dependencies is the responsibility of their respective developers.

Non-Exhaustive Testing: The audit involved automated analysis, manual review, and testing under controlled conditions. It is possible that certain vulnerabilities or issues may not have been identified.

Risk Evaluation: The audit report includes a risk assessment for identified vulnerabilities. It is recommended that the development team carefully reviews and addresses these risks to mitigate potential exploits.

Not Financial Advice: This audit report is not intended as financial or investment advice. Decisions regarding the use, deployment, or investment in the smart contract should be made based on a comprehensive assessment of the associated risks.

By accessing and using this audit report, you acknowledge and agree to the limitations outlined above. Skeleton Ecosystem and its auditors shall not be held liable for any direct or indirect damages resulting from the use of the audit report or the smart contract itself.

Please consult with legal, technical, and financial professionals before making any decisions related to the smart contract.

Overview

Contract Name	Centralium
Ticker/Simbol	\$CTM
Blockchain	Binance Smart Chain BEP20
Contract Address	0xD5a6e81d339ABe754B9754b2C16b2Be37C50B407
Creator Address	0x14be753eCA9A8c946030434692231A3D6AA87471
Current Owner Address	0x000000000000000000000000000000000000dead
Contract Explorer	https://bscscan.com/token/0xd5a6e81d339abe754b9754b2c16b2be37c50b407#code
Compiler Version	v0.8.17+commit.8df45f5f
License	MIT
Optimisation	Yes with 200 Runs
Total Supply	1,000,000,000 \$CTM
Decimals	9




Creation/Audit

Contract Deployed	15 Aug 2023
Audit Created	23 Nov 2023
Audit Update	V 1.0

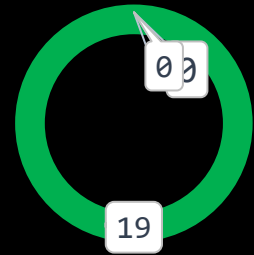
Verified Socials









Website	https://centralium.link/
Telegram	https://t.me/centralium
Twitter (X)	https://x.com/CentraliumLayer












Contract Function Analysis

 Pass
  Attention Item
  Risky Item

- Pass
- Attention
- Risk



Contract Verified		The contract source code is uploaded to blockchain explorer and is open source, so everybody can read it.
Contract Ownership		Renounced 0x00dead
Buy Tax	5 %	Shows the taxes for purchase transactions. Above 10% may be considered a high tax rate. More than 50% tax rate means may not be tradable. Fee can be set!
Sell Tax	5 %	Shows the taxes for sell transactions. Above 10% may be considered a high tax rate. More than 50% tax rate means may not be tradable. Fee can be set!
Honeypot Analyse		Holder is able to buy and sell. If honeypot: The contract blocks sell transfer from holder wallet. Multiple events may cause honeypot. Trading disabled, extremely high tax
Liquidity Status		LP Lock Status on 22.11.2023: 88.71% Pinklock for 36206 days. Lp Burned: 11.14%
Trading Disable Functions		No Trading suspendable function found. If a suspendable code is included, the token maybe neither be bought or sold (honeypot risk). If contract is renounced this function can't be used
Set Fees function	 	Fee Setting function found, but contract is renounced, this function can not be triggered. The contract owner may contain the authority to modify the transaction tax. If the transaction tax is increased to more than 49%, the tokens may not be able to be traded (honeypot risk).
Proxy Contract		Not a proxy contract!
Mint Function		No Mint Function detected Mint function is transparent or non-existent. Hidden mint functions may increase the amount of tokens in circulation and effect the price of the token. Owner can mint new tokens and sell.

Balance Modifier Function		<p>No Balance Modifier function found.</p> <p>If there is a function for this, the contract owner can have the authority to modify the balance of tokens at other addresses. For example revoke the bought tokens from the holders wallet. Common form of scam: You buy the token, but it's disappearing from your wallet.</p>
Blacklist Function		<p>No Blacklist Setting function found.</p> <p>If there is a blacklist, some addresses may not be able to trade normally. Example: you buy the token and right after your Wallet getting blacklisted. Like so you will be unable to sell. Honeypot Risk.</p>
Whitelist Function	 	<p>Whitelist Setting function found, but contract is renounced, this function can not be triggered.</p> <p>If there is a function for this Developer can set zero fee or no max wallet size for addresses (for example team wallets can trade without fee. Can cause farming)</p>
Hidden Owner Analysis		<p>No Hidden or multi owner with authorisation</p> <p>For contract with a hidden owner, developer can still manipulate the contract even if the ownership has been abandoned.</p>
Retrieve Ownership Function		<p>No Functions found which can retrieve ownership of the contract.</p> <p>If this function exists, it is possible for the project owner to regain ownership even after relinquishing it. Also known as fake renounce.</p>
Self Destruct Function		<p>No Self Destruct function found.</p> <p>If this function exists and is triggered, the contract will be destroyed, all functions will be unavailable, and all related assets will be erased.</p>
Specific Tax Changing Function		<p>No Specific Tax Changing Functions found.</p> <p>If it exists, the contract owner may set a very outrageous tax rate for assigned address to block it from trading. Can assign all wallets at once!</p>
Trading Cooldown Function		<p>No Trading Cooldown Function found. If there is a trading cooldown function, the user will not be able to sell the token within a certain time or block after buying. Like a temporary honeypot.</p>
Max Transaction and Holding Modify Function		<p>No Max Transaction and Holding Modify function found.</p> <p>If there is a function for this, the maximum trading amount or maximum position can be modified. Can cause honeypot</p>
Transaction Limiting Function		<p>No Transaction Limiter Function Found.</p> <p>The number of overall token transactions may be limited (honeypot risk)</p>

Details of Risk - Attention Items



Set Fee

Risk Removed → Renounced Contract!

The contract owner may contain the authority to modify the transaction tax. If the transaction tax is increased to more than 49%, the tokens may not be able to be traded

```

546 ftrace | funcSig
    function setBuyTaxes(uint256 newLiquidityTax!, uint256 newMarketingTax!, uint256 newBuyBackTax!) external onlyOwner() {
547         _buyLiquidityFee = newLiquidityTax!;
548         _buyMarketingFee = newMarketingTax!;
549         _buyBuyBackFee = newBuyBackTax!;
550
551         _totalTaxIfBuying = _buyLiquidityFee.add(_buyMarketingFee).add(_buyBuyBackFee);
552     }
553
554 ftrace | funcSig
    function setSellTaxes(uint256 newLiquidityTax!, uint256 newMarketingTax!, uint256 newBuyBackTax!) external onlyOwner() {
555         _sellLiquidityFee = newLiquidityTax!;
556         _sellMarketingFee = newMarketingTax!;
557         _sellBuyBackFee = newBuyBackTax!;
558
559         _totalTaxIfSelling = _sellLiquidityFee.add(_sellMarketingFee).add(_sellBuyBackFee);
560     }
561
  
```



Whitelist Function

Risk Removed → Renounced Contract!

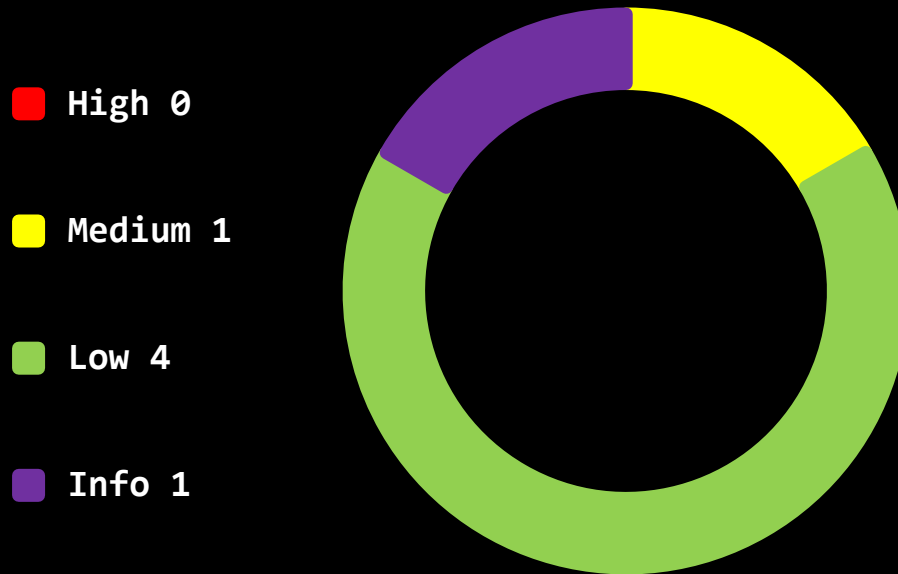
If there is a function for this, Developer can set zero fee or no max wallet size for adresses (for example team wallets can trade without fee. Can cause farming)


```


541
542 ftrace | funcSig
    function setIsExcludedFromFee(address account!, bool newValue!) public onlyOwner {
543         _isExcludedFromFee[account!] = newValue!;
544     }
545
  
```


Contract Security


Total Findings: 6



 **High Severity Issues:** High possibility to cause problems, need to be resolved.

 **Medium Severity Issue:** Will likely cause problems, recommended to resolve.

 **Low Severity Issues:** Won't cause problems, but for improvement purposes could be adjusted.

 **Informational Severity Issues:** Not harmful in any way, information for the developer team.


Contract Security

List of Found Issues


 **High severity Issues: (0)**

 **Medium severity issues: (1)**

- Incorrect Acces Control

 **Low severity issues: (4)**

- Missing Events
- Long Number Literals
- Floating Pragma
- Outdated Compiler Version

 **Informational severity issues: (1)**

- Public Functions Should be Declared External

Contract Weakness Classisication

THE SMART CONTRACT WEAKNESS CLASSIFICATION REGISTRY (SWC REGISTRY) IS AN IMPLEMENTATION OF THE WEAKNESS CLASSIFICATION SCHEME PROPOSED IN EIP-1470. IT IS LOOSELY ALIGNED TO THE TERMINOLOGIES AND STRUCTURE USED IN THE COMMON WEAKNESS ENUMERATION (CWE) WHILE OVERLAYING A WIDE RANGE OF WEAKNESS VARIANTS THAT ARE SPECIFIC TO SMART CONTRACTS.

ID	Description	AI	Manual	Result
SWC-100	Function Default Visibility	Passed	Passed	Passed
SWC-101	Integer Overflow and Underflow	Passed	Passed	Passed
SWC-102	Outdated Compiler Version	Low	Passed	Passed
SWC-103	Floating Pragma	Low	Passed	Passed
SWC-104	Unchecked Call Return Value	Passed	Passed	Passed
SWC-105	Unprotected Ether Withdrawal	Passed	Passed	Passed
SWC-106	Unprotected SELFDESTRUCT Instruction	Passed	Passed	Passed
SWC-107	Reentrancy	Passed	Passed	Passed
SWC-108	State Variable Default Visibility	Passed	Passed	Passed
SWC-109	Uninitialized Storage Pointer	Passed	Passed	Passed
SWC-110	Assert Violation	Passed	Passed	Passed
SWC-111	Use of Deprecated Solidity Functions	Passed	Passed	Passed
SWC-112	Delegatecall to Untrusted Callee	Passed	Passed	Passed
SWC-113	DoS with Failed Call	Passed	Passed	Passed
SWC-114	Transaction Order Dependence	Passed	Passed	Passed
SWC-115	Authorization through tx.origin	Passed	Passed	Passed
SWC-116	Block values as a proxy for time	Passed	Passed	Passed
SWC-117	Signature Malleability	Passed	Passed	Passed
SWC-118	Incorrect Constructor Name	Passed	Passed	Passed
SWC-119	Shadowing State Variables	Passed	Passed	Passed
SWC-120	Weak Sources of Randomness from Chain Attributes	Passed	Passed	Passed

SWC-121	Missing Protection against Signature Replay Attacks	Passed	Passed	Passed
SWC-122	Lack of Proper Signature Verification	Passed	Passed	Passed
SWC-123	Requirement Violation	Passed	Passed	Passed
SWC-124	Write to Arbitrary Storage Location	Passed	Passed	Passed
SWC-125	Incorrect Inheritance Order	Passed	Passed	Passed
SWC-126	Insufficient Gas Griefing	Passed	Passed	Passed
SWC-127	Arbitrary Jump with Function Type Variable	Passed	Passed	Passed
SWC-128	DoS With Block Gas Limit	Passed	Passed	Passed
SWC-129	Typographical Error	Passed	Passed	Passed
SWC-130	Right-To-Left-Override control character (U+202E)	Passed	Passed	Passed
SWC-131	Presence of unused variables	Passed	Passed	Passed
SWC-132	Unexpected Ether balance	Passed	Passed	Passed
SWC-133	Hash Collisions With Multiple Variable Length Arguments	Passed	Passed	Passed
SWC-134	Message call with hardcoded gas amount	Passed	Passed	Passed
SWC-135	Code With No Effects	Passed	Passed	Passed
SWC-136	Unencrypted Private Data On-Chain	Passed	Passed	Passed

Detected High and Medium Severity Vulnerability Description.

⚠️ Incorrect Acces Control (3 Item)

Item: 1	Location:	Line 520-525	Severity:	■ Medium
---------	-----------	--------------	-----------	----------

Function	<p>Access control plays an important role in segregation of privileges in smart contracts and other applications. If this is misconfigured or not properly validated on sensitive functions, it may lead to loss of funds, tokens and in some cases compromise of the smart contract.</p> <p>The contract NetronProtocol is importing an access control library @openzeppelin/contracts/access/Ownable.sol but the function approve is missing the modifier onlyOwner.</p>
Remediation	<p>It is recommended to go through the contract and observe the functions that are lacking an access control modifier. If they contain sensitive administrative actions, it is advised to add a suitable modifier to the same</p>

```

521      ftrace | funcSig
522      function approve(address spender!, uint256 amount!) public override returns (bool) {
523          _approve(_msgSender(), spender!, amount!);
524          return true;
525      }
  
```

Item: 2	Location:	Line 637-642	Severity: ■ Medium
---------	-----------	--------------	---

Function	The contract NetronProtocol is importing an access control library @openzeppelin/contracts/access/Ownable.sol but the function transfer is missing the modifier onlyOwner.
Remediation	It is recommended to go through the contract and observe the functions that are lacking an access control modifier. If they contain sensitive administrative actions, it is advised to add a suitable modifier to the same

```

637 |
638 |     ftrace | funcSig
639 |     function transfer(address recipient!, uint256 amount!) public override returns (bool) {
640 |         _transfer(msgSender(), recipient!, amount!);
641 |         return true;
642 |     }
  
```

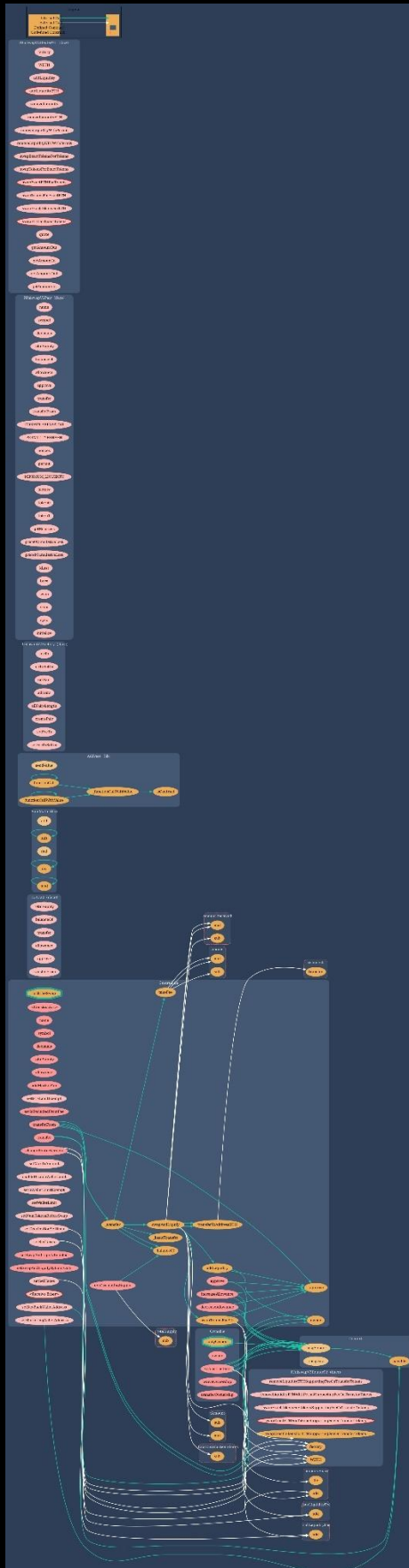
Item: 3	Location:	Line 642-648	Severity: Medium
---------	-----------	--------------	---

Function	The contract NetronProtocol is importing an access control library @openzeppelin/contracts/access/Ownable.sol but the function transferFrom is missing the modifier onlyOwner.
Remediation	It is recommended to go through the contract and observe the functions that are lacking an access control modifier. If they contain sensitive administrative actions, it is advised to add a suitable modifier to the same

```

642      ftrace | funcSig
643      function transferFrom(address sender!, address recipient!, uint256 amount!) public override returns (bool) {
644          _transfer(sender!, recipient!, amount!);
645          _approve(sender!, _msgSender(), allowances[sender!][_msgSender()].sub(amount!, "ERC20: transfer amount exceeds allowance"));
646          return true;
647      }
648  
```

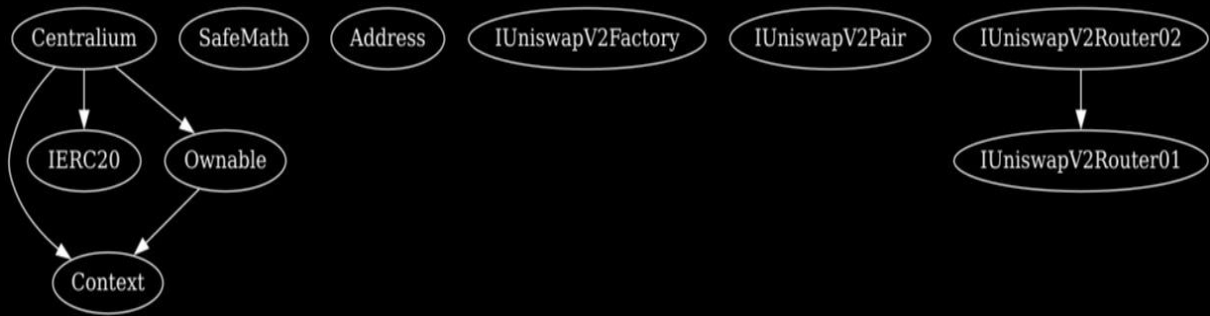
Contract Flow Graph



Contract Interaction Graph
















































Inheritance Graph























































































Contract Functions











Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		
IERC20	Interface			
L	totalSupply	External 		NO 
L	balanceOf	External 		NO 
L	transfer	External 		NO 
L	allowance	External 		NO 
L	approve	External 		NO 
L	transferFrom	External 		NO 
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 		

Contract	Type	Bases		
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		
L		Public 🔓	⬢	NO 🔓
L	owner	Public 🔓		NO 🔓
L	waiveOwnership	Public 🔓	⬢	onlyOwner
L	transferOwnership	Public 🔓	⬢	onlyOwner
IUniswapV2Factory	Interface			
L	feeTo	External 🔓		NO 🔓
L	feeToSetter	External 🔓		NO 🔓
L	getPair	External 🔓		NO 🔓
L	allPairs	External 🔓		NO 🔓
L	allPairsLength	External 🔓		NO 🔓
















Contract	Type	Bases		
L	createPair	External 		NO 
L	setFeeTo	External 		NO 
L	setFeeToSetter	External 		NO 
IUniswapV2Pair	Interface			
L	name	External 		NO 
L	symbol	External 		NO 
L	decimals	External 		NO 
L	totalSupply	External 		NO 
L	balanceOf	External 		NO 
L	allowance	External 		NO 
L	approve	External 		NO 
L	transfer	External 		NO 
L	transferFrom	External 		NO 
L	DOMAIN_SEPARATOR	External 		NO 
L	PERMIT_TYPEHASH	External 		NO 
L	nonces	External 		NO 
L	permit	External 		NO 
L	MINIMUM_LIQUIDITY	External 		NO 
L	factory	External 		NO 
L	token0	External 		NO 

Contract	Type	Bases		
L	token1	External 		NO 
L	getReserves	External 		NO 
L	price0CumulativeLast	External 		NO 
L	price1CumulativeLast	External 		NO 
L	kLast	External 		NO 
L	burn	External 		NO 
L	swap	External 		NO 
L	skim	External 		NO 
L	sync	External 		NO 
L	initialize	External 		NO 
IUniswapV2Router01	Interface			
L	factory	External 		NO 
L	WETH	External 		NO 
L	addLiquidity	External 		NO 
L	addLiquidityETH	External 		NO 
L	removeLiquidity	External 		NO 
L	removeLiquidityETH	External 		NO 
L	removeLiquidityWithPermit	External 		NO 
L	removeLiquidityETHWithPermit	External 		NO 

Contract	Type	Bases		
L	swapExactTokensForTokens	External 		NO 
L	swapTokensForExactTokens	External 		NO 
L	swapExactETHForTokens	External 		NO 
L	swapTokensForExactETH	External 		NO 
L	swapExactTokensForETH	External 		NO 
L	swapETHForExactTokens	External 		NO 
L	quote	External 		NO 
L	getAmountOut	External 		NO 
L	getAmountIn	External 		NO 
L	getAmountsOut	External 		NO 
L	getAmountsIn	External 		NO 
IUniswapV2Router02	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External 		NO 
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External 		NO 
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External 		NO 

Contract	Type	Bases		
	pportingFeeOnTransferTokens			
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO!
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO!
Centralium	Implementation	Context, IERC20, Ownable		
L		Public !		NO!
L	name	Public !		NO!
L	symbol	Public !		NO!
L	decimals	Public !		NO!
L	totalSupply	Public !		NO!
L	balanceOf	Public !		NO!
L	allowance	Public !		NO!
L	increaseAllowance	Public !		NO!
L	decreaseAllowance	Public !		NO!
L	approve	Public !		NO!
L	_approve	Private 		
L	addMarketPair	Public !		onlyOwner
L	setIsTxLimitExempt	External !		onlyOwner

Contract	Type	Bases		
L	setIsExcludedFromFee	Public 		onlyOwner
L	setBuyTaxes	External 		onlyOwner
L	setSellTaxes	External 		onlyOwner
L	setDistributionSettings	External 		onlyOwner
L	setMaxTxAmount	External 		onlyOwner
L	enableDisableWalletLimit	External 		onlyOwner
L	setIsWalletLimitExempt	External 		onlyOwner
L	setWalletLimit	External 		onlyOwner
L	setNumTokensBeforeSwap	External 		onlyOwner
L	setMarketingWalletAddress	External 		onlyOwner
L	setBuyBackWalletAddress	External 		onlyOwner
L	setSwapAndLiquifyEnabled	Public 		onlyOwner
L	setSwapAndLiquifyByLimitOnly	Public 		onlyOwner
L	getCirculatingSupply	Public 		NO 
L	transferToAddressETH	Private 		
L	changeRouterVersion	Public 		onlyOwner

Contract	Type	Bases		
L		External !		NO!
L	transfer	Public !		NO!
L	transferFrom	Public !		NO!
L	_transfer	Private 		
L	_basicTransfer	Internal 		
L	swapAndLiquify	Private 		lockTheSwap
L	swapTokensForEth	Private 		
L	addLiquidity	Private 		
L	takeFee	Internal 		


 Function
can modify
state

 Function
is payable

Audit Scope

Audit Method.

Our smart contract audit is an extensive methodical examination and analysis of the smart contract's code that is used to interact with the blockchain. Goal: discover errors, issues and security vulnerabilities in the code. Findings getting reported and improvements getting suggested.

Automatic and Manual Review

We are using automated tools to scan functions and weaknesses of the contract. Transfers, integer over-undeflow checks such as all CWE events.

Tools we use:

Visual Studio Code

CWE

SWC

Solidity Scan

SVD

In manual code review our auditor looking at source code and performing line by line examination. This method helps to clarify developer's coding decisions and business logic.

Skeleton Ecosystem

<https://skeletonecosystem.com>

<https://github.com/SkeletonEcosystem/Audits>

