

# SKELETON ECOSYSTEM

SMART CONTRACT AUDIT

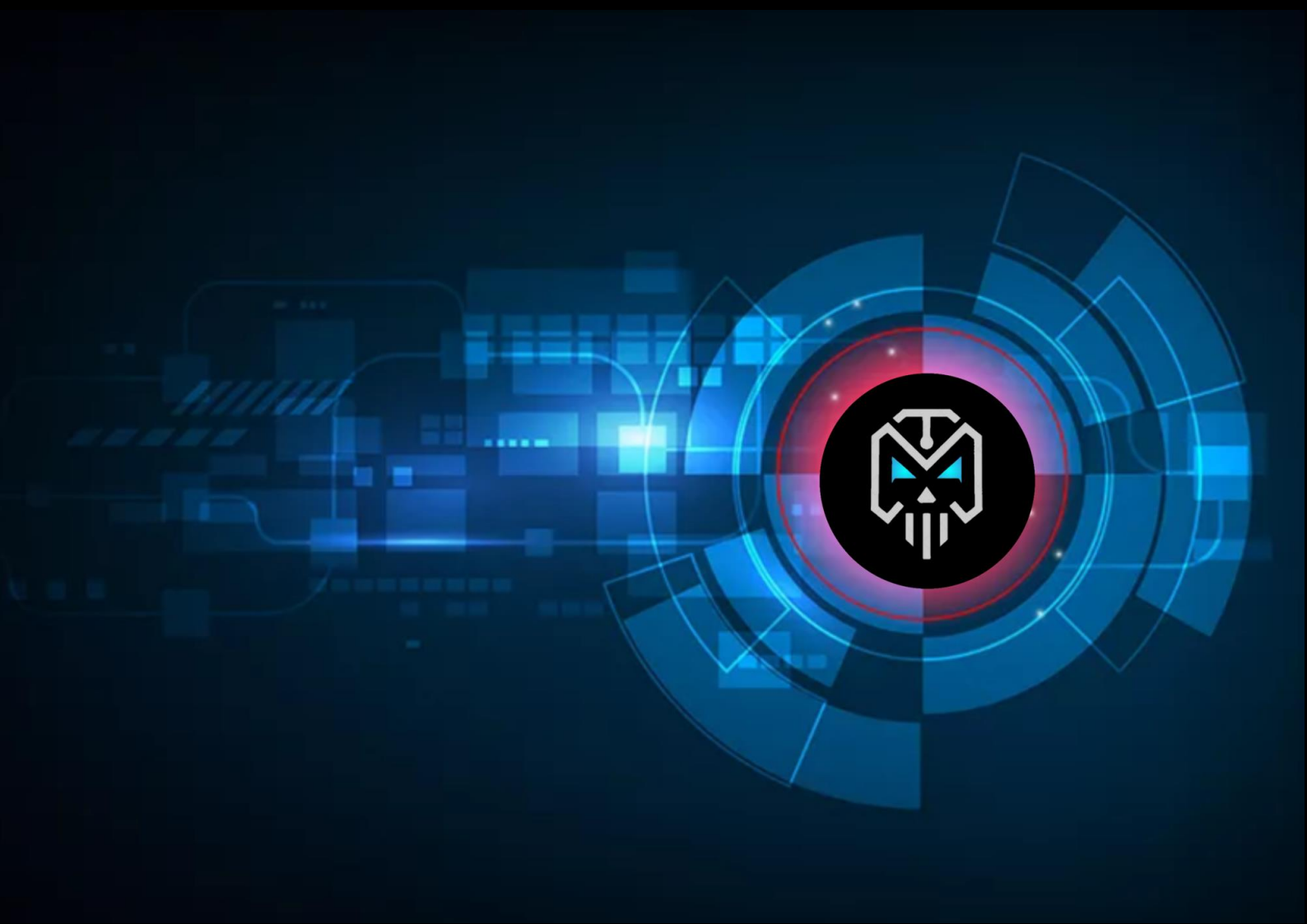


**Engine Labs**

**[EGN]**

**BEP 20**

0xe84608a7038b7e87491D681b4Bd483ebafEaE1d8



## 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	8
Detected Vulnerability Description	11
Contract Flow Graph	13
Contract Interaction Graph	14
Inheritance Graph	15
Contract Descriptions	16
Audit Scope	24

## 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	Engine Labs
Ticker/Symbol	\$EGN
Blockchain	Binance Smart Chain BEP20
Contract Address	0xe84608a7038b7e87491D681b4Bd483ebafEaE1d8
Creator Address	0xfCF282432f3Be5584a244270bD2d464B59553CF1
Current Owner Address	0xfCF282432f3Be5584a244270bD2d464B59553CF1
Contract Explorer	<a href="https://bscscan.com/token/0xe84608a7038b7e87491d681b4bd483ebafeae1d8">https://bscscan.com/token/0xe84608a7038b7e87491d681b4bd483ebafeae1d8</a>
Compiler Version	v0.8.17+commit.8df45f5f
License	MIT
Optimisation	Yes with 200 Runs
Total Supply	100,000,000 \$EGN
Decimals	9

## Creation/Audit

Contract Deployed	12 Oct 2023
Audit Created	17 Oct 2023
Audit Update	V 1.0

## Verified Socials

Website	<a href="https://enginelabs.net/">https://enginelabs.net/</a>
Telegram	<a href="https://t.me/enginelabs_bsc">https://t.me/enginelabs_bsc</a>
Twitter (X)	<a href="https://x.com/Engine_Labs">https://x.com/Engine_Labs</a>

## Contract Function Analysis



Pass



Attention Item

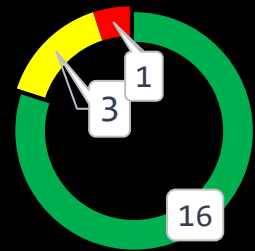


Risky Item

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		0xfCF282432f3Be5584a244270bD2d464B59553CF1
Buy Tax	10 %	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.
Sell Tax	15 %	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.
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: 99% Locked for 179 Days on Pinksale Locker
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.  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. The proxy contract means contract owner can modify the function of the token and possibly effect the price. The Owner is not the creator but the creator may have authorisation to change functions.
Mint Function		No mint Function found  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 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 function found.</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 authorised hidden owner found.</p> <p>For contract with a hidden owner, developer can still manipulate the contract even if the ownership has been abandoned. Fake renounce.</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>Max Transaction and Holding Modify function.</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 Risky - Attention Items

### **Set Fee** (Remedation: Renounce Ownership)

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).

```

553   ftrace | funcSig
      function setSellTaxes(uint256 newLiquidityTax!, uint256 newMarketingTax!, uint256 newBuyBackTax!) external onlyOwner() {
554       _sellLiquidityFee = newLiquidityTax!;
555       _sellMarketingFee = newMarketingTax!;
556       _sellBuyBackFee = newBuyBackTax!;
557
558       _totalTaxIfSelling = _sellLiquidityFee.add(_sellMarketingFee).add(_sellBuyBackFee);
559   }
  
```

### **Whitelist (Set Exluded from Fee)**

(Remedation: Renounce Ownership)

If there is a function for this, the Developer can set zero fee for adresses (for example team wallets can trade without fee)

```

540
      ftrace | funcSig
541   function setIsExcludedFromFee(address account!, bool newValue!) public onlyOwner {
542       _isExcludedFromFee[account!] = newValue!;
543   }
544
  
```

## ⚠ Max Transaction and Holding Modify Function

(Remediation: Renounce Ownership)

If there is a function for this, the maximum trading amount or maximum position can be modified. Can cause honeypot

```
569     ftrace | funcSig
570     function setMaxTxAmount(uint256 maxTxAmount!) external onlyOwner() {
571         require(maxTxAmount! <= (40 * 10**6 * 10**9), "Max wallet should be less or euqal to 4% totalSupply");
572         _maxTxAmount = maxTxAmount!;
573     }
```

```
582     ftrace | funcSig
583     function setWalletLimit(uint256 newLimit!) external onlyOwner {
584         _walletMax = newLimit!;
585     }
```



## Contract Security

Total Findings: 7

■ High 0

■ Medium 0

■ Low 4

■ Info 3



■ **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: (0)**

 **Low severity issues: (4)**

- Outdated Compiler Version
- Floating Pragma
- Missing Events
- Numeric Notation Best Practices

 **Informational severity issues: (3)**

- Variables should be immutable
- Public Functions Should be Declared External
- Hard Coded Address


## 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.

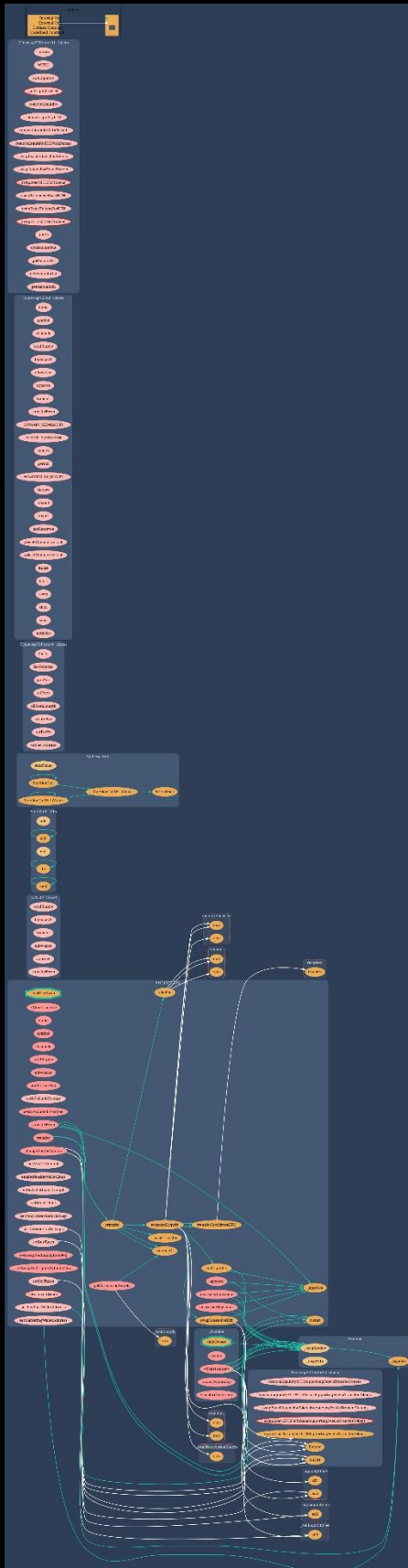
ID	Description	AI	Manual	Result
SWC-100	Function Default Visibility	Passed	Passed	Passed
SWC-101	Integer Overflow and Underflow	Passed	Passed	Passed
SWC-102	Outdated Compiler Version	Low	Passed	Passed
SWC-103	Floating Pragma	low	Passed	Passed
SWC-104	Unchecked Call Return Value	Passed	Passed	Passed
SWC-105	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

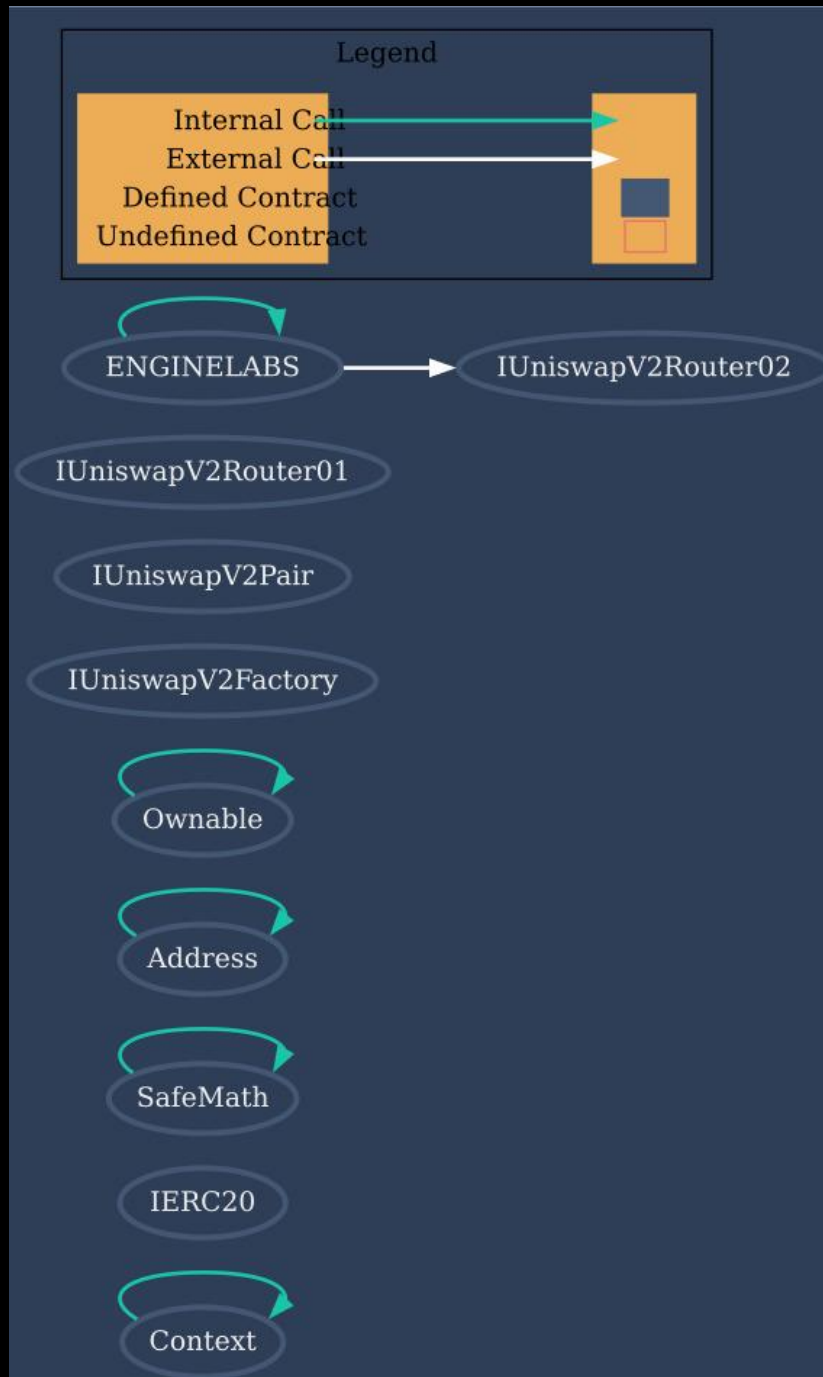
## Detected High and Medium Severity Vulnerability Description

 No High – Medium Severity Vulnerability found!

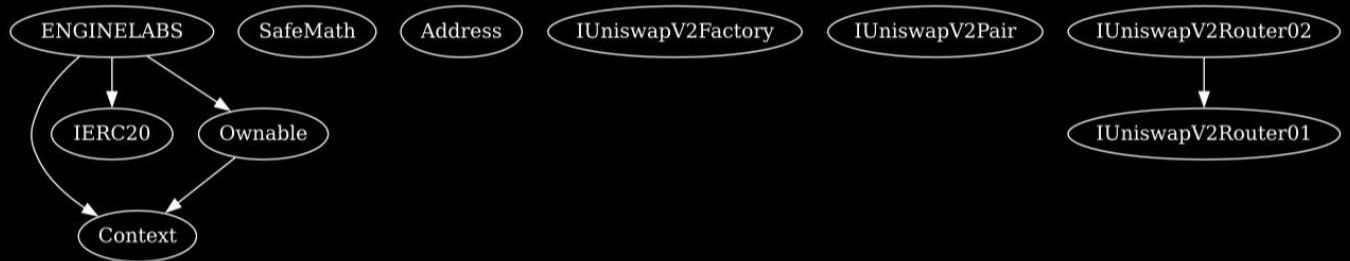
## Contract Flow Graph



## Interaction Graph



























## Inheritance Graph



































































































## Contract Functions

Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
<b>Context</b>	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		
<b>IERC20</b>	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 
<b>SafeMath</b>	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 🔒		
<b>Address</b>	Library			
L	isContract	Internal 🔒		
L	sendValue	Internal 🔒	⚙️	
L	functionCall	Internal 🔒	⚙️	
L	functionCall	Internal 🔒	⚙️	
L	functionCallWithValue	Internal 🔒	⚙️	
L	functionCallWithValue	Internal 🔒	⚙️	
L	_functionCallWithValue	Private 🔒🔑	⚙️	
<b>Ownable</b>	Implementation	Context		
L		Public ⚠️	⚙️	NO⚠️
L	owner	Public ⚠️		NO⚠️
L	waiveOwnership	Public ⚠️	⚙️	onlyOwner
L	transferOwnership	Public ⚠️	⚙️	onlyOwner
<b>IUniswapV2Factory</b>	Interface			
L	feeTo	External ⚠️		NO⚠️
L	feeToSetter	External ⚠️		NO⚠️
L	getPair	External ⚠️		NO⚠️
L	allPairs	External ⚠️		NO⚠️
L	allPairsLength	External ⚠️		NO⚠️





L	createPair	External 		NO 
L	setFeeTo	External 		NO 
L	setFeeToSetter	External 		NO 
<b>IUniswapV2Pair</b>	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 
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 ⓘ
<b>IUniswapV2Router01</b>	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 ⓘ
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 
<b>IUniswapV2Router02</b>	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External 		NO 
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External 		NO 
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External 		NO 
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External 		NO 

	rtngFeeOnTran sferTokens			
L	swapExactTokesForETHSupportingFeeOnTransferTokens	External !		NO!
<b>ENGINE LABS</b>	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
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
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>

