



# Smart Contract Audit

FOR

## Stewie Inu

DATED : 18 May 23'



# AUDIT SUMMARY

**Project name -** Stewie Inu

**Date:** 18 May, 2023

**Scope of Audit-** Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

**Audit Status: Passed**

## Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	1	0	2
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0



# USED TOOLS

---

## Tools:

### 1- Manual Review:

A line by line code review has been performed by audit ace team.

**2- BSC Test Network:** All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

### 3- Slither :

The code has undergone static analysis using Slither.

### Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/token/0x91cae5ce3694b88dfddbb3dcc12809bd90aba903>

---



# Token Information

---

**Token Name:** Stewie Inu

**Token Symbol:** Stewie

**Decimals:** 9

**Token Supply:** 690,000,000,000,000

**Token Address:**

0xff0ae985C3DaEeedc03eE47268a185C31Fab5Ee2

**Checksum:**

666e75b39d29acabb5178c89e9848d7b41227d93

**Owner:**

0x1dC827EaEd70Bd835858fc590C3f8267e2E321Ee

**Deployer:**

0x1dC827EaEd70Bd835858fc590C3f8267e2E321Ee

---



# TOKEN OVERVIEW

---

## Fees:

Buy Fees: 0-25%

Sell Fees: 0-25%

Transfer Fees: 0-25%

---

## Fees Privilege: Owner

---

Ownership: 0x1dC827EaEd70Bd835858fc590C3f8267e2E321Ee

---

Minting: No mint function

---

Max Tx Amount/ Max Wallet Amount: No

---

Blacklist: No

---

Other Privileges: - changing fees

- excluding from fees
  - including in fees
  - changing swap threshold
-



# AUDIT METHODOLOGY

---

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

# VULNERABILITY CHECKLIST



Return values of low-level calls



**Gasless Send**



Private modifier



Using block.timestamp



Multiple Sends



Re-entrancy



Using Suicide



Tautology or contradiction



Gas Limit and Loops



Timestamp Dependence



Address hardcoded



Revert/require functions



Exception Disorder



Use of tx.origin



Using inline assembly



Integer overflow/underflow



Divide before multiply



Dangerous strict equalities



Missing Zero Address Validation



Using SHA3



Compiler version not fixed



Using throw

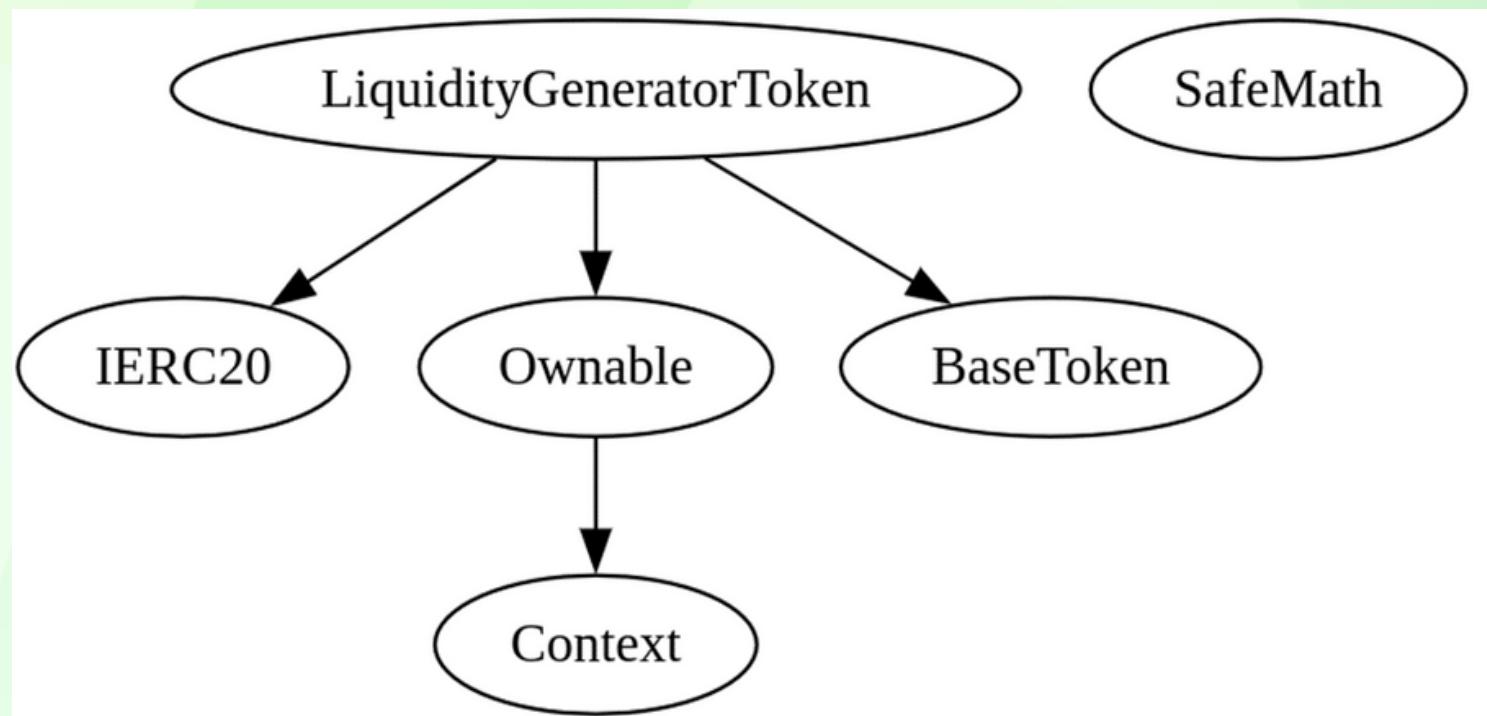
# CLASSIFICATION OF RISK

Severity	Description
◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization / Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

## Findings

Severity	Found
◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	1
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	2

# INHERITANCE TREE





# SEAVERTY

Category	Subject	Severity	Suggestion/Summary
Centralization	Fee setting and updating	Medium	Implement a safe range for fees
Centralization	Exclusion from fees and rewards	Informational	-----
Centralization	Swap and liquify settings	Informational	-----

# CONTRACT ASSESSMENT

Contract	Type	Bases			
Function Name   Visibility   Mutability   Modifiers					
**IERC20**   Interface					
totalSupply   External !   NO !					
balanceOf   External !	!	NO !			
transfer   External !	!	●   NO !			
allowance   External !	!	NO !			
approve   External !	!	●   NO !			
transferFrom   External !	!	●   NO !			
**Context**   Implementation					
_msgSender   Internal 🔒					
_msgData   Internal 🔒					
**Ownable**   Implementation   Context					
<Constructor>   Public !	!	●   NO !			
owner   Public !	!	NO !			
renounceOwnership   Public !	!	●   onlyOwner			
transferOwnership   Public !	!	●   onlyOwner			
_setOwner   Private 🔒	!	●			
**SafeMath**   Library					
tryAdd   Internal 🔒					
trySub   Internal 🔒					
tryMul   Internal 🔒					
tryDiv   Internal 🔒					
tryMod   Internal 🔒					
add   Internal 🔒					
sub   Internal 🔒					
mul   Internal 🔒					
div   Internal 🔒					
mod   Internal 🔒					
sub   Internal 🔒					
div   Internal 🔒					
mod   Internal 🔒					
**Address**   Library					
isContract   Internal 🔒					
sendValue   Internal 🔒	!	●			
functionCall   Internal 🔒	!	●			
functionCall   Internal 🔒	!	●			

# CONTRACT ASSESSMENT

L	functionCallWithValue	Internal	🔒	●	
L	functionCallWithValue	Internal	🔒	●	
L	functionStaticCall	Internal	🔒		
L	functionStaticCall	Internal	🔒		
L	functionDelegateCall	Internal	🔒	●	
L	functionDelegateCall	Internal	🔒	●	
L	verifyCallResult	Internal	🔒		

\*\*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 !	
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 !	
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		💸	NO !	
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		●	NO !	

\*\*IUniswapV2Factory\*\*	Interface				
L	feeTo	External !		NO !	
L	feeToSetter	External !		NO !	
L	getPair	External !		NO !	
L	allPairs	External !		NO !	
L	allPairsLength	External !		NO !	

# CONTRACT ASSESSMENT

L   createPair   External !   ●   NO !
L   setFeeTo   External !   ●   NO !
L   setFeeToSetter   External !   ●   NO !
**BaseToken**   Implementation
**LiquidityGeneratorToken**   Implementation   IERC20, Ownable, BaseToken
L   <Constructor>   Public !   💸   NO !
L   name   Public !     NO !
L   symbol   Public !     NO !
L   decimals   Public !     NO !
L   totalSupply   Public !     NO !
L   balanceOf   Public !     NO !
L   transfer   Public !   ●   NO !
L   allowance   Public !     NO !
L   approve   Public !   ●   NO !
L   transferFrom   Public !   ●   NO !
L   increaseAllowance   Public !   ●   NO !
L   decreaseAllowance   Public !   ●   NO !
L   isExcludedFromReward   Public !     NO !
L   totalFees   Public !     NO !
L   deliver   Public !   ●   NO !
L   reflectionFromToken   Public !     NO !
L   tokenFromReflection   Public !     NO !
L   excludeFromReward   Public !   ●   onlyOwner
L   includeInReward   External !   ●   onlyOwner
L   _transferBothExcluded   Private 🔒   ●
L   excludeFromFee   Public !   ●   onlyOwner
L   setTaxFeePercent   External !   ●   onlyOwner
L   setLiquidityFeePercent   External !   ●   onlyOwner
L   setCharityFeePercent   External !   ●   onlyOwner
L   setSwapBackSettings   External !   ●   onlyOwner
L   <Receive Ether>   External !   💸   NO !
L   _reflectFee   Private 🔒   ●
L   _getValues   Private 🔒
L   _getTValues   Private 🔒
L   _getRValues   Private 🔒
L   _getRate   Private 🔒
L   _getCurrentSupply   Private 🔒
L   _takeLiquidity   Private 🔒   ●
L   _takeCharityFee   Private 🔒   ●
L   calculateTaxFee   Private 🔒

# CONTRACT ASSESSMENT

L   calculateLiquidityFee   Private		
L   calculateCharityFee   Private		
L   removeAllFee   Private		
L   restoreAllFee   Private		
L   isExcludedFromFee   Public		NO 
L   _approve   Private		
L   _transfer   Private		
L   swapAndLiquify   Private		   lockTheSwap
L   swapTokensForEth   Private		
L   addLiquidity   Private		
L   _tokenTransfer   Private		
L   _transferStandard   Private		
L   _transferToExcluded   Private		
L   _transferFromExcluded   Private		

## Legend

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



## POINTS TO NOTE

---

- Owner is able to change buy/sell/transfer fees in range 0-25%
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to set max buy/sell/transfer/hold amount to 0
- Owner is not able to mint new tokens



# STATIC ANALYSIS

```
Variable LiquidityGeneratorToken._getValues(uint256).rTransferAmount (contracts/Token.sol#1331) is too similar to LiquidityGeneratorToken._getValues(uint256).tTransferAmount (contracts/Token.sol#1326)
Variable LiquidityGeneratorToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1208) is too similar to LiquidityGeneratorToken._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken._getValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1372-1374) is too similar to LiquidityGeneratorToken._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1609)
Variable LiquidityGeneratorToken.transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1256)
Variable LiquidityGeneratorToken.transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1587) is too similar to LiquidityGeneratorToken._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1634)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._getValues(uint256).tTransferAmount (contracts/Token.sol#1355-1357)
Variable LiquidityGeneratorToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1208) is too similar to LiquidityGeneratorToken._getValues(uint256).tTransferAmount (contracts/Token.sol#1326)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol#1589)
Variable LiquidityGeneratorToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1254) is too similar to LiquidityGeneratorToken._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1587) is too similar to LiquidityGeneratorToken._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1634)
Variable LiquidityGeneratorToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1587) is too similar to LiquidityGeneratorToken._getValues(uint256).tTransferAmount (contracts/Token.sol#1326)
Variable LiquidityGeneratorToken._getValues(uint256).rTransferAmount (contracts/Token.sol#1331) is too similar to LiquidityGeneratorToken._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1256)
Variable LiquidityGeneratorToken._getValues(uint256).rTransferAmount (contracts/Token.sol#1331) is too similar to LiquidityGeneratorToken._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1634)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._transferFromExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1634)
Variable LiquidityGeneratorToken._getValues(uint256).rTransferAmount (contracts/Token.sol#1331) is too similar to LiquidityGeneratorToken._getValues(uint256).tTransferAmount (contracts/Token.sol#1355-1357)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1632) is too similar to LiquidityGeneratorToken._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1208) is too similar to LiquidityGeneratorToken._getValues(uint256).tTransferAmount (contracts/Token.sol#1256)
Variable LiquidityGeneratorToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1208) is too similar to LiquidityGeneratorToken._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1256)
Variable LiquidityGeneratorToken._getValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1372-1374) is too similar to LiquidityGeneratorToken._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1256)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._getValues(uint256).tTransferAmount (contracts/Token.sol#1326)
Variable LiquidityGeneratorToken.transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1587) is too similar to LiquidityGeneratorToken._transferBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1256)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

LiquidityGeneratorToken._charityAddress (contracts/Token.sol#1003) should be immutable
LiquidityGeneratorToken._decimals (contracts/Token.sol#990) should be immutable
LiquidityGeneratorToken._name (contracts/Token.sol#988) should be immutable
LiquidityGeneratorToken._symbol (contracts/Token.sol#989) should be immutable
LiquidityGeneratorToken._tTotal (contracts/Token.sol#984) should be immutable
LiquidityGeneratorToken.swapAndLiquifyEnabled (contracts/Token.sol#1006) should be immutable
LiquidityGeneratorToken.uniswapV2Pair (contracts/Token.sol#1002) should be immutable
LiquidityGeneratorToken.uniswapV2Router (contracts/Token.sol#1001) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
Contracts Token.sol analyzed (0 contracts with 94 statements, 105 results('')) found
```

**Result => A static analysis of contract's source code has been performed using slither,  
No major issues were found in the output**



# FUNCTIONAL TESTING

## Router (PCS V2):

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

All the functionalities have been tested, no issues were found

### 1- Adding liquidity (**passed**):

<https://testnet.bscscan.com/tx/0x0930cf255c16067a771929f79f67f58f3b8626a3164e0a21260cfdf6d4695ec7>

### 2- Buying when excluded (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xc5bf066033c95a354bfd80a6c8fc95d4245464a932d0dd9ef5fcc2c216871b84>

### 3- Selling when excluded (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x8e171d5738f487bd42dfa967c6f536a8f6e3aa0919bd1fd082dc3e898ca7e624>

### 4- Transferring when excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xc6690df4f931c0c37fba5a939e9df654133115e2edc4258822a03211a3db7a79>

### 5- Buying from a regular wallet (0-25% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xffaf535e8fe0a10e7993c87db321c9e8251f310b693642e1683a4cbf10c64663>

### 6- Selling from a regular wallet (0-25% tax) (**passed**):

<https://testnet.bscscan.com/tx/0-x3ffd693d3c83802a81f6a8738009edaf2bcf574af0da5146e5fc3ed2ca496781>



# FUNCTIONAL TESTING

---

**7- Transferring from a regular wallet (0-25% tax) (passed):**

<https://testnet.bscscan.com/tx/0x0f1acaca5a87a504b1a086319f913c3229c3d3c8b4cc0e5f4749fc5e3e6aaa2f>

**8- Internal swap (marketing bnb + auto-liquidity) (passed):**

<https://testnet.bscscan.com/tx/0x3ffd693d3c83802a81f6a8738009edaf2bcf574af0da5146e5fc3ed2ca496781>



# MANUAL TESTING

**Category:** Centralization

**Subject:** Fee setting and updating

**Severity:** Medium

**Status:** not applicable

**Overview:**

The contract allows the owner to set and update various fees, including tax, liquidity, and charity fees. This centralizes control over the fee structure.

Each type of tax (buy, sell, transfer) can have 0-25% fee.

**Code:**

```
function setTaxFeePercent(uint256 taxFeeBps) external onlyOwner { ... }  
function setLiquidityFeePercent(uint256 liquidityFeeBps) external onlyOwner { ... }  
function setCharityFeePercent(uint256 charityFeeBps) external onlyOwner { ... }
```

**Suggestion:**

Ensure that sum of max buy and sell fee is less than 25%

buy + sell fee <= 25%

transfer fee <= 5%



# MANUAL TESTING

**Category:** Centralization

**Subject:** Exclusion from fees and rewards

**Severity:** Informational

**Status:** not applicable

**Overview:**

The contract allows the owner to exclude certain addresses from fees and rewards. This centralizes control over the fee and reward distribution.

**Code:**

```
function excludeFromReward(address account) public onlyOwner { ... }  
function includeInReward(address account) external onlyOwner { ... }  
function excludeFromFee(address account) public onlyOwner { ... }
```

**Suggestion:**

Consider implementing a decentralized governance mechanism to allow the community to decide on the exclusion or inclusion of addresses in fees and rewards.



# MANUAL TESTING

**Category:** Centralization

**Subject:** Swap and liquify settings

**Severity:** Informational

**Status:** not applicable

**Overview:**

The contract allows the owner to set the swap back settings, which affects the swap and liquify process. This centralizes control over the contract's liquidity management.

Setting swap threshold to a large number can increase slippage % on sells

**Code:**

```
function setSwapBackSettings(uint256 _amount) external onlyOwner { ... }
```

**Suggestion:**

Consider implementing a decentralized governance mechanism to allow the community to decide on the swap back settings and other liquidity management parameters.



# DISCLAIMER

---

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.



# ABOUT AUDITACE

---

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



**<https://auditace.tech/>**



**[https://t.me/Audit\\_Ace](https://t.me/Audit_Ace)**



**[https://twitter.com/auditace\\_](https://twitter.com/auditace_)**



**<https://github.com/Audit-Ace>**

---