



# Smart Contract Audit

FOR

## Spare Inu

DATED : 5 May 23'



# AUDIT SUMMARY

**Project name - Spare Inu**

**Date:** 5 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	1	0
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/0xa92ff91743783591494f80d947cd47d73ae6ffed>

---



# Token Information

---

**Token Name:** Spare Inu

**Token Symbol:** Spare Inu

**Decimals:** 18

**Token Supply:** 1,000,000,000

**Token Address:**

0xd8E5E55838c6FD402464F5E085ea7a9659fa835D

**Checksum:**

36552580df187046a7c9f54d1f604aee2936f4e8

**Owner:**

0x3862f88396a0b50Db96B8beFF0d8A4B2Af3E5702

**(at time of writing the audit)**

**Deployer:**

0xf24539CfD9dFfBD40b78beEd25eDb40ad4b3306A

---



# TOKEN OVERVIEW

---

## Fees:

Buy Fees: up to 10%

Sell Fees: up to 10%

Transfer Fees: up to 10%

---

**Fees Privilege:** Owner

---

**Ownership:** Owned

---

**Minting:** No mint function

---

**Max Tx Amount/ Max Wallet Amount:** Yes

---

**Blacklist:** No

---

**Other Privileges:** updating fee - excluding from fees - including in fees

---



# 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



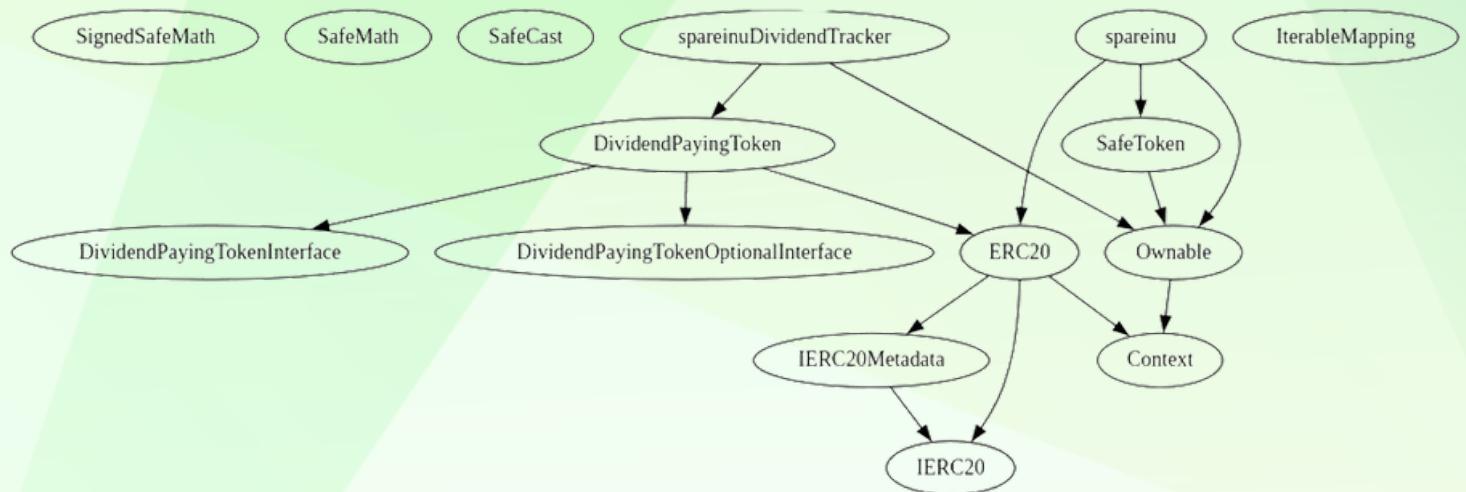
# CLASSIFICATION OF RISK

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	1
◆ Gas Optimization / Suggestions	0

# INHERITANCE TREE





## POINTS TO NOTE

---

- Owner is not able to set buy/sell/transfer taxes over 10%
- Owner is not able to set a max buy/transfer/wallet/sell amount
- Owner is able to blacklist an arbitrary wallet
- Owner is able to disable trades
- Owner is not able to mint new tokens



# CONTRACT ASSESSMENT

Contract	Type	Bases			
	L	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
**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 !					
L   createPair   External !     ●     NO !					
L   setFeeTo   External !     ●     NO !					
L   setFeeToSetter   External !     ●     NO !					
**SignedSafeMath**   Library					



# CONTRACT ASSESSMENT

```
| L | mul | Internal | 🔒 | |||  
| L | div | Internal | 🔒 | |||  
| L | sub | Internal | 🔒 | |||  
| L | add | Internal | 🔒 | |||  
|||||  
| **SafeMath** | Library | ||||  
| L | tryAdd | Internal | 🔒 | |||  
| L | trySub | Internal | 🔒 | |||  
| L | tryMul | Internal | 🔒 | |||  
| L | tryDiv | Internal | 🔒 | |||  
| L | tryMod | Internal | 🔒 | |||  
| L | add | Internal | 🔒 | |||  
| L | sub | Internal | 🔒 | |||  
| L | mul | Internal | 🔒 | |||  
| L | div | Internal | 🔒 | |||  
| L | mod | Internal | 🔒 | |||  
| L | sub | Internal | 🔒 | |||  
| L | div | Internal | 🔒 | |||  
| L | mod | Internal | 🔒 | |||  
|||||  
| **SafeCast** | Library | ||||  
| L | toUint224 | Internal | 🔒 | |||  
| L | toUint128 | Internal | 🔒 | |||  
| L | toUint96 | Internal | 🔒 | |||  
| L | toUint64 | Internal | 🔒 | |||  
| L | toUint32 | Internal | 🔒 | |||  
| L | toUint16 | Internal | 🔒 | |||  
| L | toUint8 | Internal | 🔒 | |||  
| L | toUint256 | Internal | 🔒 | |||  
| L |ToInt128 | Internal | 🔒 | |||  
| L |ToInt64 | Internal | 🔒 | |||  
| L |ToInt32 | Internal | 🔒 | |||  
| L |ToInt16 | Internal | 🔒 | |||  
| L |ToInt8 | Internal | 🔒 | |||  
| L |ToInt256 | Internal | 🔒 | |||  
|||||  
| **Context** | Implementation | ||||  
| L | _msgSender | Internal | 🔒 | |||  
| L | _msgData | Internal | 🔒 | |||  
|||||  
| **IERC20** | Interface | ||||  
| L | totalSupply | External | ! | |NO| ! |  
| L | balanceOf | External | ! | |NO| ! |
```



# CONTRACT ASSESSMENT

L   transfer   External !	●   NO !
L   allowance   External !	NO !
L   approve   External !	●   NO !
L   transferFrom   External !	●   NO !
**IERC20Metadata**   Interface   IERC20	
L   name   External !	NO !
L   symbol   External !	NO !
L   decimals   External !	NO !
**ERC20**   Implementation   Context, IERC20, IERC20Metadata	
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   _transfer   Internal 🔒	●
L   _mint   Internal 🔒	●
L   _burn   Internal 🔒	●
L   _approve   Internal 🔒	●
L   _beforeTokenTransfer   Internal 🔒	●
L   _afterTokenTransfer   Internal 🔒	●
**Ownable**   Implementation   Context	
L   <Constructor>   Public !	●   NO !
L   owner   Public !	NO !
L   renounceOwnership   Public !	●   onlyOwner
L   transferOwnership   Public !	●   onlyOwner
L   _setOwner   Private 🔒	●
**IterableMapping**   Library	
L   get   Internal 🔒	
L   getIndexOfKey   Internal 🔒	
L   getKeyAtIndex   Internal 🔒	
L   size   Internal 🔒	
L   set   Internal 🔒	●

# CONTRACT ASSESSMENT

```
| L | remove | Internal 🔒 | ● || | |
|||  
| **DividendPayingTokenOptionalInterface** | Interface | |||  
| L | withdrawableDividendOf | External ! | |NO ! |  
| L | withdrawnDividendOf | External ! | |NO ! |  
| L | accumulativeDividendOf | External ! | |NO ! |  
|||  
| **DividendPayingTokenInterface** | Interface | |||  
| L | dividendOf | External ! | |NO ! |  
| L | distributeDividends | External ! | | $ | |NO ! |  
| L | withdrawDividend | External ! | ● | |NO ! |  
|||  
| **DividendPayingToken** | Implementation | ERC20, DividendPayingTokenInterface,  
DividendPayingTokenOptionalInterface |||  
| L | <Constructor> | Public ! | ● | ERC20 | | |
| L | <Receive Ether> | External ! | | $ | |NO ! |  
| L | distributeDividends | Public ! | | $ | |NO ! |  
| L | withdrawDividend | Public ! | ● | |NO ! |  
| L | _withdrawDividendOfUser | Internal 🔒 | ● | |  
| L | dividendOf | Public ! | |NO ! |  
| L | withdrawableDividendOf | Public ! | |NO ! |  
| L | withdrawnDividendOf | Public ! | |NO ! |  
| L | accumulativeDividendOf | Public ! | |NO ! |  
| L | _transfer | Internal 🔒 | ● | |  
| L | mint | Internal 🔒 | ● | |  
| L | burn | Internal 🔒 | ● | |  
| L | _setBalance | Internal 🔒 | ● | |  
|||  
| **spareinuDividendTracker** | Implementation | DividendPayingToken, Ownable |||  
| L | <Constructor> | Public ! | ● | DividendPayingToken |  
| L | _transfer | Internal 🔒 | | |  
| L | withdrawDividend | Public ! | |NO ! |  
| L | excludeFromDividends | External ! | ● | onlyOwner |  
| L | updateClaimWait | External ! | ● | onlyOwner |  
| L | getLastProcessedIndex | External ! | |NO ! |  
| L | getNumberOfTokenHolders | External ! | |NO ! |  
| L | getAccount | Public ! | |NO ! |  
| L | getAccountAtIndex | Public ! | |NO ! |  
| L | canAutoClaim | Private 🔒 | | |  
| L | setBalance | External ! | ● | onlyOwner |  
| L | process | Public ! | ● | |NO ! |  
| L | processAccount | Public ! | ● | onlyOwner |  
||||
```

# CONTRACT ASSESSMENT

```
| **SafeToken** | Implementation | Ownable ||| |
| L |<Constructor> | Public ! | ● | NO ! |
| L | setSafeManager | Public ! | ● | onlyOwner |
| L | withdraw | External ! | ● | NO ! |
| L | withdrawBNB | External ! | ● | NO ! |
|||||||  
| **spareinu** | Implementation | ERC20, Ownable, SafeToken |||
| L | setFee | Public ! | ● | onlyOwner |
| L | setExtraFeeOnSell | Public ! | ● | onlyOwner |
| L | setMarketingWallet | Public ! | ● | onlyOwner |
| L |<Constructor> | Public ! | ● | ERC20 |
| L |<Receive Ether> | External ! | ● | NO ! |
| L | excludeFromFees | Public ! | ● | onlyOwner |
| L | setExcludeFromMaxTx | Public ! | ● | onlyOwner |
| L | setExcludeFromAll | Public ! | ● | onlyOwner |
| L | excludeMultipleAccountsFromFees | Public ! | ● | onlyOwner |
| L | setAutomatedMarketMakerPair | Public ! | ● | onlyOwner |
| L | setSWapTokensAtAmount | Public ! | ● | onlyOwner |
| L | _setAutomatedMarketMakerPair | Private 🔒 | ● |||
| L | updateGasForProcessing | Public ! | ● | onlyOwner |
| L | updateClaimWait | External ! | ● | onlyOwner |
| L | getClaimWait | External ! | | NO ! |
| L | getTotalDividendsDistributed | External ! | | NO ! |
| L | isExcludedFromFees | Public ! | | NO ! |
| L | isExcludedFromMaxTx | Public ! | | NO ! |
| L | withdrawableDividendOf | Public ! | | NO ! |
| L | dividendTokenBalanceOf | Public ! | | NO ! |
| L | getAccountDividendsInfo | External ! | | NO ! |
| L | getAccountDividendsInfoAtIndex | External ! | | NO ! |
| L | processDividendTracker | External ! | ● | NO ! |
| L | claim | External ! | ● | NO ! |
| L | getLastProcessedIndex | External ! | | NO ! |
| L | getNumberOfDividendTokenHolders | External ! | | NO ! |
| L | excludeFromDividends | External ! | ● | onlyOwner |
| L | setSwapAndLiquifyEnabled | Public ! | ● | onlyOwner |
| L | _transfer | Internal 🔒 | ● |||
| L | swapAndLiquify | Private 🔒 | ● | lockTheSwap |
| L | swapTokensForBnb | Private 🔒 | ● |||
| L | swapAndSendBNBToMarketing | Private 🔒 | ● |||
| L | addLiquidity | Private 🔒 | ● |||
```

# CONTRACT ASSESSMENT

---

## Legend

Symbol	Meaning
●	Function can modify state
\$	Function is payable



# STATIC ANALYSIS

```
External calls:
- swapAndLiquify(contractTokenBalance) (contracts/GPT.sol#2082)
  - marketingWallet.transfer(marketingAmount) (contracts/GPT.sol#2140)
External calls sending eth:
- swapAndLiquify(contractTokenBalance) (contracts/GPT.sol#2082)
  - uniswapV2Router.addLiquidityETH(value: ethAmount)(address(this),tokenAmount,0,0,owner(),block.timestamp) (contracts/GPT.sol#2183-2190)
  - marketingWallet.transfer(marketingAmount) (contracts/GPT.sol#2140)
  - (success) = address(dividendTracker).call(value: dividends)() (contracts/GPT.sol#2143)
State variables written after the call(s):
- super._transfer(from,address(this),fees) (contracts/GPT.sol#2095)
  - balances[sender] = senderBalance - amount (contracts/GPT.sol#1023)
  - balances[recipient] += amount (contracts/GPT.sol#1025)
- super._transfer(from,to,amount) (contracts/GPT.sol#2098)
  - balances[sender] = senderBalance - amount (contracts/GPT.sol#1023)
  - balances[recipient] += amount (contracts/GPT.sol#1025)
Event emitted after the call(s):
- ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,tx.origin) (contracts/GPT.sol#2107)
- Transfer(sender,recipient,amount) (contracts/GPT.sol#1027)
  - super._transfer(from,to,amount) (contracts/GPT.sol#2098)
- Transfer(sender,recipient,amount) (contracts/GPT.sol#1027)
  - super._transfer(from,address(this),fees) (contracts/GPT.sol#2095)
Reentrancy in spareinu.swapAndLiquify(uint256) (contracts/GPT.sol#2115-2150):
External calls:
- marketingWallet.transfer(marketingAmount) (contracts/GPT.sol#2140)
External calls sending eth:
- addLiquidity(tokensToAddLiquidityWith,bnbToAddLiquidityWith) (contracts/GPT.sol#2137)
  - uniswapV2Router.addLiquidityETH(value: ethAmount)(address(this),tokenAmount,0,0,owner(),block.timestamp) (contracts/GPT.sol#2183-2190)
- marketingWallet.transfer(marketingAmount) (contracts/GPT.sol#2140)
  - (success) = address(dividendTracker).call(value: dividends)() (contracts/GPT.sol#2143)
Event emitted after the call(s):
- SendDividends(toSwap - tokensToAddLiquidityWith,dividends) (contracts/GPT.sol#2146)
- SwapAndLiquify(tokensToAddLiquidityWith,deltaBalance) (contracts/GPT.sol#2149)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-4

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/GPT.sol#12) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/GPT.sol#13)
Variable DividendPayingToken._withdrawnDividendOfUser(address)._withdrawableDividend (contracts/GPT.sol#1413) is too similar to spareinu.DividendTracker.getAccount(address).withdrawableDividends (contracts/GPT.sol#1575)
Variable spareinu.BNBRewardsFee (contracts/GPT.sol#1780) is too similar to spareinu.setFee(uint256,uint256,uint256),_bnbRewardFee (contracts/GPT.sol#1837)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

spareinu.constructor() (contracts/GPT.sol#1861-1908) uses literals with too many digits:
- _mint(owner(),1000000000 * (10 ** 18)) (contracts/GPT.sol#1907)
spareinu.updateGasForProcessing(uint256) (contracts/GPT.sol#1966-1971) uses literals with too many digits:
- require(bool,string)(newValue >= 200000 && newValue <= 500000,spareinu: gasForProcessing must be between 200,000 and 500,000) (contracts/GPT.sol#1967)
spareinu.slitherConstructorVariables() (contracts/GPT.sol#1765-2194) uses literals with too many digits:
- maxSellTransactionAmount = 100000000 * (10 ** 18) (contracts/GPT.sol#1777)
spareinu.slitherConstructorVariables() (contracts/GPT.sol#1765-2194) uses literals with too many digits:
- gasForProcessing = 300000 (contracts/GPT.sol#1788)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

spareinu.dividendTracker (contracts/GPT.sol#1775) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

**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/0x84a76510bfa1dde78153ccf80b8feeaa41e747883c7e095a3035e9a9863ffb6f1>

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

<https://testnet.bscscan.com/tx/0x6363a1fd181b235d8aa8fb67f4a0a871134d7b83c968ab01b881407962ff3c0>

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

<https://testnet.bscscan.com/tx/0x9c7f915d14e26c6c17ff21834e5a406d747b45a3149f5519c84432b2e9deb7e9>

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

<https://testnet.bscscan.com/tx/0xad192abd79112daa9e14c05323eabdbc320731d53e47bbf987ebeca7f15550a8>

### 5- Buying when not excluded from fees (up to 10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x934fb7e0f162818ac902c9a1eb2c3ffa00505acf3966a36f21476149329eb3f>

### 6- Selling when not excluded from fees (up to 0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xf13cdd3bb8abcf431bcd0025309e611fe96e2782e5569d2d85024e7f6495676a>



# FUNCTIONAL TESTING

---

**7- Transferring when not excluded from fees (up to 10% tax)(passed):**

<https://testnet.bscscan.com/tx/0x9c4f1d351f7dce24fedad425a66c36b91fdbf470b4f25eb37972f38aae198939>

**8- Internal swap (passed):**

**Auto-liquidity & Reflections & fee wallets receiving BNB**

<https://testnet.bscscan.com/tx/0xf13cdd3bb8abcf431cd0025309e611fe96e2782e5569d2d85024e7f6495676a>



# MANUAL TESTING

---

## Centralization – LP tokens sent to an EOA

**Severity:** Medium

**Function:** addLiquidity

**Status:** Not Resolved

### Overview:

The contract owner's wallet (an EOA or Externally Owned Account) receives auto-generated liquidity pool share tokens. Over time, these tokens will accumulate and could potentially be used to remove a significant portion of liquidity, negatively impacting the token's liquidity and price.

```
function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {  
  
    // add the liquidity  
    uniswapV2Router.addLiquidityETH{value: ethAmount}(  
        address(this),  
        tokenAmount,  
        0, // slippage is unavoidable  
        0, // slippage is unavoidable  
        owner(),  
        block.timestamp  
    );  
  
}
```

### Recommendation:

To mitigate this issue, consider the following options:

- Burn new LP tokens.
- Lock new LP tokens.
- Renounce ownership of the contract, which would result in LP tokens being burned automatically.



# MANUAL TESTING

---

## Logical – 0 Swapthreshold can disable some sells and transfers or increase slippage if is set to a high value

**Severity:** Low

**Function:** setSWapTokensAtAmount - swapAndLiquify

**Lines:** 2188

**Status:** Not Resolved

### Overview:

Setting the **swapTokensAtAmount** to 0 can disable some sell/transfer operations if the contract's token balance is 0. This is because swapAndLiquify does not check whether the contract token balance is greater than 0 and attempts to swap 0 tokens for BNB. Additionally, setting the swap threshold to a very high value can increase slippage on sell/transfer operations, potentially exposing traders' tokens to front-runner bots.

```
function setSWapTokensAtAmount(uint256 _newAmount) public onlyOwner {  
    swapTokensAtAmount = _newAmount;  
}
```

### Recommendation:

there are several ways to mitigate this issue

- Ensure that the new swap threshold cannot be set to an unreasonably low or high value.
- Renounce ownership of the contract, which would prevent the swap threshold from being changed arbitrarily.



# 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>**

---