



Smart Contract Audit

FOR

Itachi Inu

DATED : 20 MAR 23'



AUDIT SUMMARY

Project name – Itachi Inu

Date: 20 March, 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	0	0	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 Testnet network:

all tests were done on Bsc Testnet network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: Already Launched

Note:

At the time of preparing this audit, the token has already been launched and is being traded. It is important to note that the audit was conducted after the token's introduction on PCSV2. As such, any potential malicious activities by the owner or issues associated with the contract are beyond the scope of this audit.



Token Information

Token Name : Itachi Inu

Token Symbol: Itachi Inu

Decimals: 9

Token Supply: 500,000,000

Token Address:

0xDdf121D1406e11D6574cb62DbC3c11Fd45A9e9c1

Checksum:

46dda4c1b172d20efb681c3c4ba56a60bd5edb31

Owner:

0x00

(renounced)

Deployer:

0x8134b687be5752eFF8361B663030420D47648bfF



TOKEN OVERVIEW

Fees:

Buy Fees: 9%

Sell Fees: 9%

Transfer Fees: 9%

Fees Privilege: None

Ownership : renounced

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: None



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 Limitand 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

0

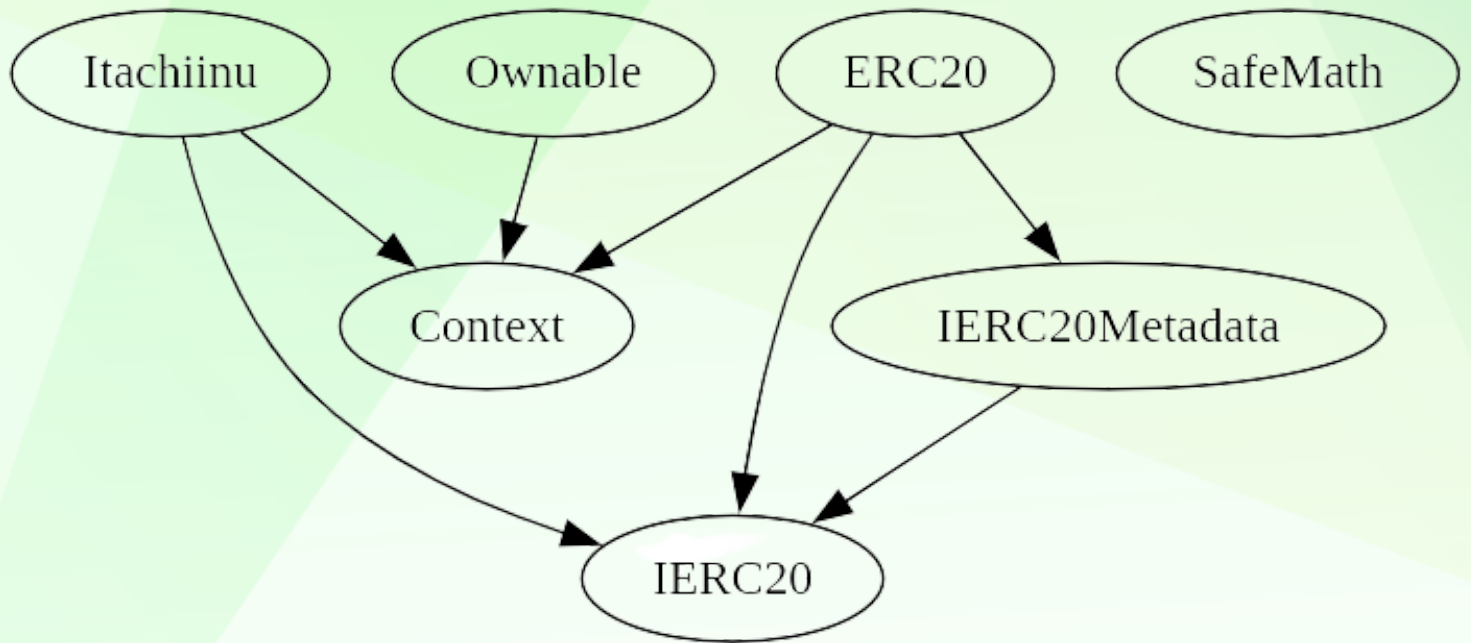
◆ Low-Risk

0

◆ Gas Optimization / Suggestions

0

INHERITANCE TREE



POINTS TO NOTE

- **Ownership is renounced, meaning owner has not control over the contract functions**
 - Owner is not able to modify buy/sell/transfer fees (9% for each)
 - Owner is not able to set max buy/sell/transfer/hold amount
 - Owner is not able to blacklist an arbitrary wallet
 - Owner is not able to disable trades
 - Owner is not able to mint new tokens
-



CONTRACT ASSESMENT

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 !	
SafeMath Library					
└─	add	Internal 🔒			
└─	sub	Internal 🔒			
└─	mul	Internal 🔒			
└─	div	Internal 🔒			
└─	sub	Internal 🔒			
└─	div	Internal 🔒			
Context Implementation					
└─	_msgSender	Internal 🔒			
└─	_msgData	Internal 🔒			
Address Library					
└─	isContract	Internal 🔒			
└─	sendValue	Internal 🔒	●		
└─	functionCall	Internal 🔒	●		
└─	functionCall	Internal 🔒	●		
└─	functionCallWithValue	Internal 🔒	●		
└─	functionCallWithValue	Internal 🔒	●		
└─	functionStaticCall	Internal 🔒			
└─	functionStaticCall	Internal 🔒			
└─	functionDelegateCall	Internal 🔒	●		
└─	functionDelegateCall	Internal 🔒	●		
└─	_verifyCallResult	Private 🔒			
IUniswapV2Factory Interface					
└─	feeTo	External !		NO !	
└─	feeToSetter	External !		NO !	
└─	getPair	External !		NO !	
└─	allPairs	External !		NO !	
└─	allPairsLength	External !		NO !	



CONTRACT ASSESMENT

```
|  | createPair | External  |  |  | NO  |
|  | setFeeTo | External  |  |  | NO  |
|  | setFeeToSetter | External  |  |  | NO  |
|||||
| **IUniswapV2Pair** | Interface | |||
|  | name | External  |  |  | NO  |
|  | symbol | External  |  |  | NO  |
|  | decimals | External  |  |  | NO  |
|  | totalSupply | External  |  |  | NO  |
|  | balanceOf | External  |  |  | NO  |
|  | allowance | External  |  |  | NO  |
|  | approve | External  |  |  | NO  |
|  | transfer | External  |  |  | NO  |
|  | transferFrom | External  |  |  | NO  |
|  | DOMAIN_SEPARATOR | External  |  |  | NO  |
|  | PERMIT_TYPEHASH | External  |  |  | NO  |
|  | nonces | External  |  |  | NO  |
|  | permit | External  |  |  | NO  |
|  | MINIMUM_LIQUIDITY | External  |  |  | NO  |
|  | factory | External  |  |  | NO  |
|  | token0 | External  |  |  | NO  |
|  | token1 | External  |  |  | NO  |
|  | getReserves | External  |  |  | NO  |
|  | price0CumulativeLast | External  |  |  | NO  |
|  | price1CumulativeLast | External  |  |  | NO  |
|  | kLast | External  |  |  | NO  |
|  | burn | External  |  |  | NO  |
|  | swap | External  |  |  | NO  |
|  | skim | External  |  |  | NO  |
|  | sync | External  |  |  | NO  |
|  | initialize | External  |  |  | NO  |
|||||
| **IUniswapV2Router01** | Interface | |||
|  | factory | External  |  |  | NO  |
|  | WETH | External  |  |  | NO  |
|  | addLiquidity | External  |  |  | NO  |
|  | addLiquidityETH | External  |  |  | NO  |
|  | removeLiquidity | External  |  |  | NO  |
|  | removeLiquidityETH | External  |  |  | NO  |
|  | removeLiquidityWithPermit | External  |  |  | NO  |
|  | removeLiquidityETHWithPermit | External  |  |  | NO  |
|  | swapExactTokensForTokens | External  |  |  | NO  |
```

CONTRACT ASSESMENT

```

|  | swapTokensForExactTokens | External ! | ● | NO ! |
|  | swapExactETHForTokens | External ! | 🟢 | NO ! |
|  | swapTokensForExactETH | External ! | ● | NO ! |
|  | swapExactTokensForETH | External ! | ● | NO ! |
|  | swapETHForExactTokens | External ! | 🟢 | NO ! |
|  | quote | External ! | | NO ! |
|  | getAmountOut | External ! | | NO ! |
|  | getAmountIn | External ! | | NO ! |
|  | getAmountsOut | External ! | | NO ! |
|  | getAmountsIn | External ! | | NO ! |
|||||
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
|  | removeLiquidityETHSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|  | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|
|  | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|  | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! | 🟢 | NO ! |
|  | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|||||
| **Itachiinu** | Implementation | Context, IERC20 |||
|  | owner | Public ! | | NO ! |
|  | renounceOwnership | Public ! | ● | NO ! |
|  | <Constructor> | Public ! | ● | NO ! |
|  | name | Public ! | | NO ! |
|  | symbol | Public ! | | NO ! |
|  | decimals | Public ! | | NO ! |
|  | totalSupply | Public ! | | NO ! |
|  | balanceOf | Public ! | | NO ! |
|  | transfer | Public ! | ● | NO ! |
|  | allowance | Public ! | | NO ! |
|  | approve | Public ! | ● | NO ! |
|  | transferFrom | Public ! | ● | NO ! |
|  | increaseAllowance | Public ! | ● | NO ! |
|  | decreaseAllowance | Public ! | ● | NO ! |
|  | <Receive Ether> | External ! | 🟢 | NO ! |
|  | _getCurrentSupply | Private 🔒 | | |
|  | _approve | Private 🔒 | ● | |
|  | _transfer | Private 🔒 | ● | |
|  | sendToWallet | Private 🔒 | ● | |
|  | swapAndLiquify | Private 🔒 | ● | lockTheSwap |
|  | swapTokensForBNB | Private 🔒 | ● | |
|  | addLiquidity | Private 🔒 | ● | |

```

CONTRACT ASSESMENT

```

|  | remove_Random_Tokens | Public ! | ● | NO ! |
|  | _tokenTransfer | Private 🔒 | ● | |
|||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
|  | <Constructor> | Public ! | ● | NO ! |
|  | name | Public ! | | NO ! |
|  | symbol | Public ! | | NO ! |
|  | decimals | Public ! | | NO ! |
|  | totalSupply | Public ! | | NO ! |
|  | balanceOf | Public ! | | NO ! |
|  | transfer | Public ! | ● | NO ! |
|  | allowance | Public ! | | NO ! |
|  | approve | Public ! | ● | NO ! |
|  | transferFrom | Public ! | ● | NO ! |
|  | increaseAllowance | Public ! | ● | NO ! |
|  | decreaseAllowance | Public ! | ● | NO ! |
|  | _transfer | Internal 🔒 | ● | |
|  | _mint | Internal 🔒 | ● | |
|  | _burn | Internal 🔒 | ● | |
|  | _approve | Internal 🔒 | ● | |
|  | _spendAllowance | Internal 🔒 | ● | |
|  | _beforeTokenTransfer | Internal 🔒 | ● | |
|  | _afterTokenTransfer | Internal 🔒 | ● | |
|||||
| **IERC20** | Interface | |||
|  | totalSupply | External ! | | NO ! |
|  | balanceOf | External ! | | NO ! |
|  | transfer | External ! | ● | NO ! |
|  | allowance | External ! | | NO ! |
|  | approve | External ! | ● | NO ! |
|  | transferFrom | External ! | ● | NO ! |
|||||
| **IERC20Metadata** | Interface | IERC20 |||
|  | name | External ! | | NO ! |
|  | symbol | External ! | | NO ! |
|  | decimals | External ! | | NO ! |
|||||
| **Context** | Implementation | |||
|  | _msgSender | Internal 🔒 | | |
|  | _msgData | Internal 🔒 | | |
|||||
| **Ownable** | Implementation | Context |||

```



CONTRACT ASSESMENT

```
| L | <Constructor> | Public ! | ● | NO ! |
| L | owner | Public ! | | NO ! |
| L | _checkOwner | Internal 🔒 | |
| L | renounceOwnership | Public ! | ● | onlyOwner |
| L | transferOwnership | Public ! | ● | onlyOwner |
| L | _transferOwnership | 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 🔒 | |
```

| Symbol | Meaning |

|:-----:|-----|

| ● | Function can modify state |

| 💰 | Function is payable |



STATIC ANALYSIS

```
Reentrancy in Itachiinu.transferFrom(address,address,uint256) (contracts/Token.sol#669-684):
  External calls:
    - _transfer(sender,recipient,amount) (contracts/Token.sol#674)
    - wallet.transfer(amount) (contracts/Token.sol#789)
  External calls sending eth:
    - _transfer(sender,recipient,amount) (contracts/Token.sol#674)
    - wallet.transfer(amount) (contracts/Token.sol#789)
    - uniswapV2Router.addLiquidityETH(value: BNBAmount)(address(this),tokenAmount,0,0,Wallet_Burn,block.timestamp) (contracts/Token.sol#828-835)
  State variables written after the call(s):
    - _approve(sender,_msgSender(),allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (contracts/Token.sol#675-682)
    - allowances[theOwner][theSpender] = amount (contracts/Token.sol#728)
  Event emitted after the call(s):
    - Approval(theOwner,theSpender,amount) (contracts/Token.sol#729)
    - _approve(sender,_msgSender(),allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (contracts/Token.sol#675-682)
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/Token.sol#355) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#356)
Variable Itachiinu.swapAndLiquify(uint256).tokens_to_D (contracts/Token.sol#789) is too similar to Itachiinu.swapAndLiquify(uint256).tokens_to_M (contracts/Token.sol#788)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

Itachiinu.slitherConstructorVariables() (contracts/Token.sol#543-883) uses literals with too many digits:
  - tTotal = 1000000000000000 * 10 ** 6 * 10 ** 6 * 10 ** 6 * 10 ** 6 decimals (contracts/Token.sol#577-578)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

Itachiinu.MAX (contracts/Token.sol#575) is never used in Itachiinu (contracts/Token.sol#543-883)
Itachiinu.previousMaxWalletToken (contracts/Token.sol#590) is never used in Itachiinu (contracts/Token.sol#543-883)
Itachiinu.previousMaxTxAmount (contracts/Token.sol#592) is never used in Itachiinu (contracts/Token.sol#543-883)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

Itachiinu.Percent_AutoLP (contracts/Token.sol#588) should be constant
Itachiinu.Percent_Burn (contracts/Token.sol#587) should be constant
Itachiinu.Percent_Dev (contracts/Token.sol#586) should be constant
Itachiinu.Percent_Marketing (contracts/Token.sol#585) should be constant
Itachiinu.Wallet_Dev (contracts/Token.sol#571-572) should be constant
Itachiinu.Wallet_Marketing (contracts/Token.sol#569-570) should be constant
Itachiinu.Tax_On_Buy (contracts/Token.sol#583) should be constant
Itachiinu.Tax_On_Sell (contracts/Token.sol#584) should be constant
Itachiinu.swapAndLiquifyEnabled (contracts/Token.sol#596) should be constant
Itachiinu.swapTrigger (contracts/Token.sol#582) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

Itachiinu_maxTxAmount (contracts/Token.sol#591) should be immutable
Itachiinu_maxWalletToken (contracts/Token.sol#589) should be immutable
Itachiinu_previousMaxTxAmount (contracts/Token.sol#592) should be immutable
Itachiinu_previousMaxWalletToken (contracts/Token.sol#590) should be immutable
Itachiinu.uniswapV2Pair (contracts/Token.sol#594) should be immutable
Itachiinu.uniswapV2Router (contracts/Token.sol#593) 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 issues found



FUNCTIONAL TESTING

Token is already launched at time of writing this audit and is being traded with an active (BNB/Itachi) pool at PCS v2. buying/selling/burning/transferring/internal swap seem to be working as expected.

At the time of preparing this audit, the token has already been launched and is being traded. It is important to note that the audit was conducted after the token's introduction on PCSV2. As such, any potential malicious activities by the owner or issues associated with the contract are beyond the scope of this audit.



MANUAL TESTING

No Issues Found





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://twitter.com/auditace_



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