auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Zazu Coin

DATED : 14 November 23'

MANUAL TESTING

Centralization – Fees can be set more than 100%

Severity: High

function: setFees

Status: Open

Overview:

In this setFees function the owner can set fees of more than 100%.

```
function setFees(
  uint256 _ecoFee_B,
  uint256 _burnFee_B,
  uint256 _ecoFee_S,
  uint256 _burnFee_S
) external onlyOwner {
  ecoFee_BUY = _ecoFee_B;
  burnFee_BUY = _burnFee_B;
  ecoFee_SELL = _ecoFee_S;
  burnFee_SELL = _burnFee_S;
  buyTax = _ecoFee_B + _burnFee_B;
  sellTax = _ecoFee_S + _burnFee_S;
}
```

Suggestion:

It is recommended to add required check on this function to set the fees not more than 25%.

AUDIT SUMMARY

Project name – Zazu Coin

Date: 14 November 2023

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

Audit Status: **Passed with high risk**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	1	0	4	1
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/address/0xD105dE9d0720f8b427027575C501a8373DA83906#code

Token Information

Token Address: -
0xa91991347324b9C5ca2827f7BC070F9cDf89B313

Name: Zazu Coin

Symbol: ZAZU

Decimals: 18

Network: Binance smart chain

Token Type: BEP20

Owner: - 0xb364a837aE2C28F0b55B1608164009dCE223cf56

Deployer: -
0xb364a837aE2C28F0b55B1608164009dCE223cf56

Token Supply: 21000000

Checksum: 6b04bb8874dd025e8664d1360b075613

Testnet version:
The tests were performed using the contract deployed on the Binance smart chain Testnet, which can be found at the following address:
https://testnet.bscscan.com/address/0xD105dE9d0720f8b427027575C501a8373DA83906#code

TOKEN OVERVIEW

Buy Tax : 10%

Sell Tax : 10%

Transfer Tax : 0%

ecoFee_BUY 5%

ecoFee_SELL 5%

burnfee_buy 5%

burnfee_sell 5%

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 isexercised 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

- [x] Return values of low-level calls
- [x] Private modifier
- [x] Multiple Sends
- [x] Using Suicide
- [x] Gas Limitand Loops
- [x] Address hardcoded
- [x] Exception Disorder
- [x] Using inline assembly
- [x] Divide before multiply
- [x] Missing Zero Address Validation
- [x] Compiler version not fixed
- [x] **Gasless Send**
- [x] Using block.timestamp
- [x] Re-entrancy
- [x] Tautology or contradiction
- [x] Timestamp Dependence
- [x] Revert/require functions
- [x] Use of tx.origin
- [x] Integer overflow/underflow
- [x] Dangerous strict equalities
- [x] Using SHA3
- [x] 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	1
Medium-Risk	0
Low-Risk	4
Gas Optimization / Suggestions	1

INHERITANCE TREE

POINTS TO NOTE

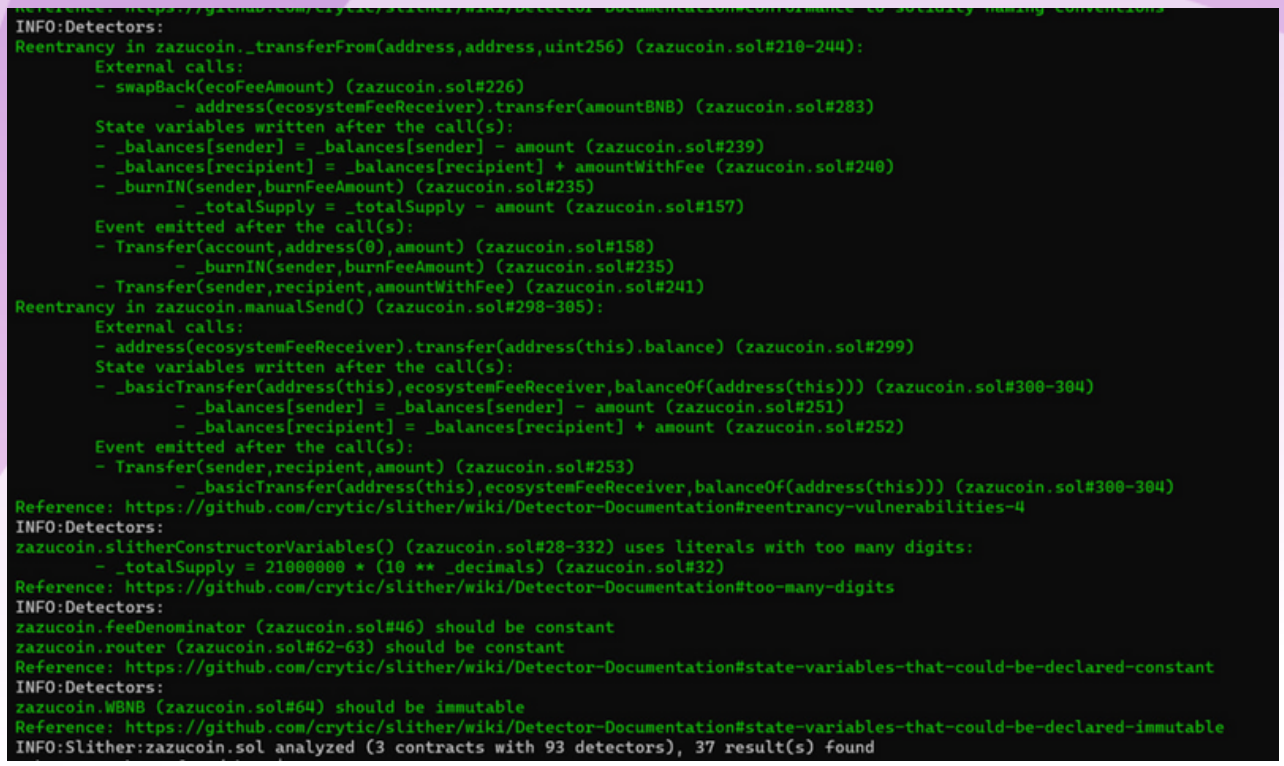
- Owner can renounce the ownership.
- Owner can transfer the ownership.
- Owner can set pair.
- Owner can exclude/include wallets from fees.
- Owner can setAutoCompoundFeeReceiver.
- Owner can set buy and sell fee more than 100%
- Owner can setSwapBackSettings.
- Owner can set swapthreshold.
- Burn fee 5% but owner can set that to 100%

STATIC ANALYSIS

```
INFO:Detectors:
zazucoin.setFees(uint256,uint256,uint256,uint256) (zazucoin.sol#285-297) should emit an event for:
        - ecoFee_BUY = _ecoFee_B (zazucoin.sol#291)
        - burnFee_BUY = _burnFee_B (zazucoin.sol#292)
        - ecoFee_SELL = _ecoFee_S (zazucoin.sol#293)
        - burnFee_SELL = _burnFee_S (zazucoin.sol#294)
        - buyTax = _ecoFee_B + _burnFee_B (zazucoin.sol#295)
        - sellTax = _ecoFee_S + _burnFee_S (zazucoin.sol#296)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
zazucoin.setecosystemFeeReceivers(address)._ecosystemFeeReceiver (zazucoin.sol#167) lacks a zero-check on :
                - ecosystemFeeReceiver = _ecosystemFeeReceiver (zazucoin.sol#171)
zazucoin.setAutoCompoundFeeReceivers(address)._autoCompoundFeeReceiver (zazucoin.sol#173) lacks a zero-check on :
                - autoRefillFeeReceiver = _autoCompoundFeeReceiver (zazucoin.sol#177)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in zazucoin._transferFrom(address,address,uint256) (zazucoin.sol#210-244):
        External calls:
        - swapBack(ecoFeeAmount) (zazucoin.sol#226)
                - router.swapExactTokensForETHSupportingFeeOnTransferTokens(a,0,path,address(this),block.timestamp) (zazucoin.sol#275-281)
        External calls sending eth:
        - swapBack(ecoFeeAmount) (zazucoin.sol#226)
                - address(ecosystemFeeReceiver).transfer(amountBNB) (zazucoin.sol#283)
        State variables written after the call(s):
        - _burnIN(sender,burnFeeAmount) (zazucoin.sol#235)
                - _totalSupply = _totalSupply - amount (zazucoin.sol#157)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in zazucoin._transferFrom(address,address,uint256) (zazucoin.sol#210-244):
        External calls:
        - swapBack(ecoFeeAmount) (zazucoin.sol#226)
                - router.swapExactTokensForETHSupportingFeeOnTransferTokens(a,0,path,address(this),block.timestamp) (zazucoin.sol#275-281)
        External calls sending eth:
        - swapBack(ecoFeeAmount) (zazucoin.sol#226)
                - address(ecosystemFeeReceiver).transfer(amountBNB) (zazucoin.sol#283)
        Event emitted after the call(s):
        - Transfer(account,address(0),amount) (zazucoin.sol#158)
```

```
INFO:Detectors:
zazucoin.WBNB (zazucoin.sol#64) is set pre-construction with a non-constant function or state variable:
        - router.WETH()
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state
INFO:Detectors:
Pragma version^0.8.7 (zazucoin.sol#11) allows old versions
solc-0.8.22 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Function IDEXRouter.WETH() (zazucoin.sol#19) is not in mixedCase
Contract zazucoin (zazucoin.sol#28-332) is not in CapWords
Parameter zazucoin.setPair(address,bool)._pair (zazucoin.sol#134) is not in mixedCase
Parameter zazucoin.setecosystemFeeReceivers(address)._ecosystemFeeReceiver (zazucoin.sol#167) is not in mixedCase
Parameter zazucoin.setAutoCompoundFeeReceivers(address)._autoCompoundFeeReceiver (zazucoin.sol#173) is not in mixedCase
Parameter zazucoin.setSwapBackSettings(bool)._enabled (zazucoin.sol#179) is not in mixedCase
Function zazucoin.BURNFEE(bool) (zazucoin.sol#196-202) is not in mixedCase
Function zazucoin.ECOFEE(bool) (zazucoin.sol#203-209) is not in mixedCase
Parameter zazucoin.setFees(uint256,uint256,uint256,uint256)._ecoFee_B (zazucoin.sol#286) is not in mixedCase
Parameter zazucoin.setFees(uint256,uint256,uint256,uint256)._burnFee_B (zazucoin.sol#287) is not in mixedCase
Parameter zazucoin.setFees(uint256,uint256,uint256,uint256)._ecoFee_S (zazucoin.sol#288) is not in mixedCase
Parameter zazucoin.setFees(uint256,uint256,uint256,uint256)._burnFee_S (zazucoin.sol#289) is not in mixedCase
Constant zazucoin._name (zazucoin.sol#29) is not in UPPER_CASE_WITH_UNDERSCORES
Constant zazucoin._symbol (zazucoin.sol#30) is not in UPPER_CASE_WITH_UNDERSCORES
Constant zazucoin._decimals (zazucoin.sol#31) is not in UPPER_CASE_WITH_UNDERSCORES
Variable zazucoin._isExcludedFromFee (zazucoin.sol#35) is not in mixedCase
Variable zazucoin.ecoFee_BUY (zazucoin.sol#41) is not in mixedCase
Variable zazucoin.burnFee_BUY (zazucoin.sol#42) is not in mixedCase
Variable zazucoin.ecoFee_SELL (zazucoin.sol#44) is not in mixedCase
Variable zazucoin.burnFee_SELL (zazucoin.sol#45) is not in mixedCase
Variable zazucoin.WBNB (zazucoin.sol#64) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Reentrancy in zazucoin._transferFrom(address,address,uint256) (zazucoin.sol#210-244):
        External calls:
        - swapBack(ecoFeeAmount) (zazucoin.sol#226)
                - address(ecosystemFeeReceiver).transfer(amountBNB) (zazucoin.sol#283)
        State variables written after the call(s):
        - _balances[sender] = _balances[sender] - amount (zazucoin.sol#239)
        - _balances[recipient] = _balances[recipient] + amountWithFee (zazucoin.sol#240)
```

STATIC ANALYSIS



```
INFO:Detectors:
Reentrancy in zazucoin._transferFrom(address,address,uint256) (zazucoin.sol#210-244):
        External calls:
        - swapBack(ecoFeeAmount) (zazucoin.sol#226)
                - address(ecosystemFeeReceiver).transfer(amountBNB) (zazucoin.sol#283)
        State variables written after the call(s):
        - _balances[sender] = _balances[sender] - amount (zazucoin.sol#239)
        - _balances[recipient] = _balances[recipient] + amountWithFee (zazucoin.sol#240)
        - _burnIN(sender,burnFeeAmount) (zazucoin.sol#235)
                - _totalSupply = _totalSupply - amount (zazucoin.sol#157)
        Event emitted after the call(s):
        - Transfer(account,address(0),amount) (zazucoin.sol#158)
                - _burnIN(sender,burnFeeAmount) (zazucoin.sol#235)
        - Transfer(sender,recipient,amountWithFee) (zazucoin.sol#241)
Reentrancy in zazucoin.manualSend() (zazucoin.sol#298-305):
        External calls:
        - address(ecosystemFeeReceiver).transfer(address(this).balance) (zazucoin.sol#299)
        State variables written after the call(s):
        - _basicTransfer(address(this),ecosystemFeeReceiver,balanceOf(address(this))) (zazucoin.sol#300-304)
                - _balances[sender] = _balances[sender] - amount (zazucoin.sol#251)
                - _balances[recipient] = _balances[recipient] + amount (zazucoin.sol#252)
        Event emitted after the call(s):
        - Transfer(sender,recipient,amount) (zazucoin.sol#253)
                - _basicTransfer(address(this),ecosystemFeeReceiver,balanceOf(address(this))) (zazucoin.sol#300-304)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-4
INFO:Detectors:
zazucoin.slitherConstructorVariables() (zazucoin.sol#28-332) uses literals with too many digits:
        - _totalSupply = 21000000 * (10 ** _decimals) (zazucoin.sol#32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
zazucoin.feeDenominator (zazucoin.sol#46) should be constant
zazucoin.router (zazucoin.sol#62-63) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
zazucoin.WBNB (zazucoin.sol#64) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:zazucoin.sol analyzed (3 contracts with 93 detectors), 37 result(s) 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

1- Approve (passed):

https://testnet.bscscan.com/tx/0xdcd0df44fa643dd295f341b39bb658c0d9e7b00abef5763209f4a6e4e43a4844

2- Exclude From Fee(passed):

https://testnet.bscscan.com/tx/0xd6b337ff0e860b46813cf4f0e2274b86d335d541a9b7fedae3d98b25e4369802

3- Include In Fee(passed):

https://testnet.bscscan.com/tx/0x6a8e555c3745d353277f77284342b0e621b898f09e6d77b6b275ab5bd0595b52

4- Manual Send (passed):

https://testnet.bscscan.com/tx/0xb4a81928fd203ecbbffa205645f133e5d6d1ead772f61753b91175652dc33f27

5- Set Swap Back Settings (passed):

https://testnet.bscscan.com/tx/0xf5983c00adb2d1a149c1041ca2688c95f762e6bcd5a892af82f2a690acf9a54b

6- Set Pair (passed):

https://testnet.bscscan.com/tx/0xd190b149a8e347327c089642b23fe13ffcbf7addeb2bc3d0c048e63f61be8fc9

7- Set Ecosystem Fee Receiver (passed):

https://testnet.bscscan.com/tx/0x349c982eb837055cfdb4cea961ce06df27556a57c38e977a346d9f2fcc8caa99

MANUAL TESTING

Centralization – Fees can be set more than 100%

Severity: High

function: setFees

Status: Open

Overview:

In this setFees function the owner can set fees of more than 100%.

```
function setFees(
  uint256 _ecoFee_B,
  uint256 _burnFee_B,
  uint256 _ecoFee_S,
  uint256 _burnFee_S
) external onlyOwner {
  ecoFee_BUY = _ecoFee_B;
  burnFee_BUY = _burnFee_B;
  ecoFee_SELL = _ecoFee_S;
  burnFee_SELL = _burnFee_S;
  buyTax = _ecoFee_B + _burnFee_B;
  sellTax = _ecoFee_S + _burnFee_S;
}
```

Suggestion:

It is recommended to add required check on this function to set the fees not more than 25%.

MANUAL TESTING

Centralization – Missing Zero Address

Severity: Low

function: setecosystemFeeRecrivers and setAutoCompoundFeeReceivers

Status: Open

Overview:

functions can take a zero address as a parameter (0x00000...). If a function parameter of address type is not properly validated by checking for zero addresses, there could be serious consequences for the contract's functionality.

```
 function setecosystemFeeReceivers(address
_ecosystemFeeReceiver)
   external
   onlyOwner
 {
   ecosystemFeeReceiver = _ecosystemFeeReceiver;
 }
 function setAutoCompoundFeeReceivers(address
_autoCompoundFeeReceiver)
   external
   onlyOwner
 {
   autoRefillFeeReceiver = _autoCompoundFeeReceiver;
 }
```

Suggestion:

It is suggested that the address should not be zero or dead.

MANUAL TESTING

Centralization – Missing Visibility
Severity: Low
function: Mapping and Uint256
Status: Open

Overview:
It's simply saying that no visibility was specified, so it's going with the default. This has been related to security issues in contracts.

```
 uint256 _totalSupply = 21000000 *
(10**_decimals);
  mapping(address => uint256) _balances;
  mapping(address => mapping(address =>
uint256)) _allowances;
```

Suggestion:
You can easily silence the warning by adding the mapping and Uint256 public/private.

MANUAL TESTING

Severity: Low

subject: Missing Events

Status: Open

Overview:

They serve as a mechanism for emitting and recording data onto the blockchain, making it
transparent and easily accessible.

```
function setPair(address _pair, bool io) public onlyOwner {
    pair[_pair] = io;
  }
function setecosystemFeeReceivers(address
_ecosystemFeeReceiver)
    external
    onlyOwner
  {
    ecosystemFeeReceiver = _ecosystemFeeReceiver;
  }
  function setAutoCompoundFeeReceivers(address
_autoCompoundFeeReceiver)
    external
    onlyOwner
  {
    autoRefillFeeReceiver = _autoCompoundFeeReceiver;
  }
  function setSwapBackSettings(bool _enabled) external onlyOwner
{
    swapEnabled = _enabled;
  }
```

MANUAL TESTING

Severity: Low

subject: Missing error message

Status: Open

Overview:

Missing requires an error message.

```
require(amount != 0);
    require(amount <= _balances[account]);
```

Suggestion:

It is suggested that to pass some error message in require check.

MANUAL TESTING

Severity: Informational
subject: floating Pragma Solidity version
Status: Open

Overview:
It is considered best practice to pick one compiler version and stick with it. With a floating pragma, contracts may accidentally be deployed using an outdated.

pragma solidity ^0.8.7;

Suggestion
Adding the latest constant version of solidity is recommended, as this prevents the unintentional deployment of a contract with an outdated compiler that contains unresolved bugs.

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 specializes 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