AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

CAT CEO

DATED : 2 MARCH 23'

AUDIT SUMMARY

Project name – CAT CEO

Date: 2 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 Test Network:
all tests were done on BSC Test network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

https://bscscan.com/address/0x4937e7D93dd8d8e76Eb83659F109Cdc633FfDEe9#readContract

Token Information

Token Name : CAT CEO

Token Symbol: CATCEO

Decimals: 9

Token Supply: 420,000,000,000,000,000

Token Address:
https://bscscan.com/address/0x4937e7D93dd8d8e76Eb83659F109Cdc633FfDEe9

Checksum:
388560a874c128a9b7669f396b80ede50a44d6ce

Owner:
0xca94554ea45D90897F46F7d12aff87e4f2D997c1

Deployer:
0xca94554ea45D90897F46F7d12aff87e4f2D997c1

TOKEN OVERVIEW

Fees:

Buy Fees: 5%

Sell Fees: 5%

Transfer Fees: 5%

Fees Privilige: Static fees

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: including and excluding from rewards and 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 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

- ✅ Return values of low-level calls
- ✅ Private modifier
- ✅ Multiple Sends
- ✅ Using Suicide
- ✅ Gas Limitand Loops
- ✅ Address hardcoded
- ✅ Exception Disorder
- ✅ Using inline assembly
- ✅ Divide before multiply
- ✅ Missing Zero Address Validation
- ✅ Compiler version not fixed

- ✅ **Gasless Send**
- ✅ Using block.timestamp
- ✅ Re-entrancy
- ✅ Tautology or contradiction
- ✅ Timestamp Dependence
- ✅ Revert/require functions
- ✅ Use of tx.origin
- ✅ Integer overflow/underflow
- ✅ Dangerous strict equalities
- ✅ Using SHA3
- ✅ 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.
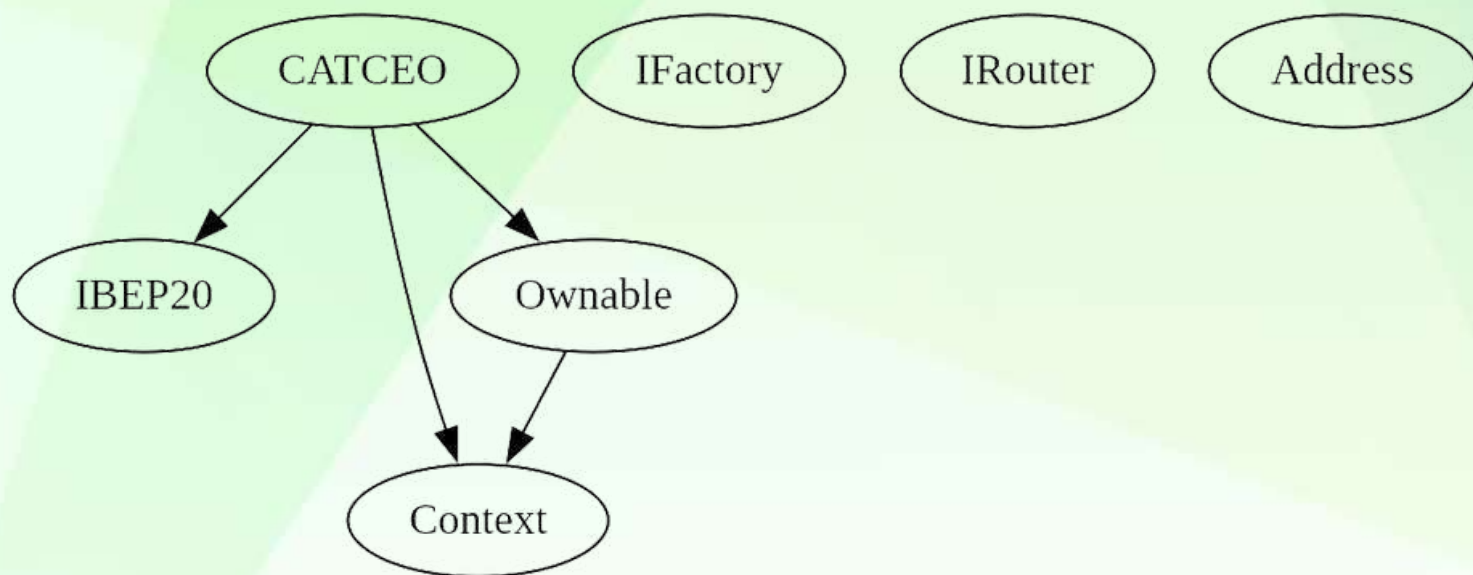
Findings

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

INHERITANCE TREE

POINTS TO NOTE

- Owner is not able to change buy taxes (5% static)
- Owner is not able to change sell taxes (5% static)
- Owner is not able to change transfer taxes (5% static)
- Owner is not able to set max wallet/transfer/buy/sell amounts
- 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**
IBEP20	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	
└	_setOwner	Private 🔐	🔴		
IFactory	Interface				
└	createPair	External ❗	🔴 |NO ❗	
IRouter	Interface				
└	factory	External ❗	|NO ❗	
└	WETH	External ❗	|NO ❗	
└	addLiquidityETH	External ❗	💵 |NO ❗	
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔴 |NO ❗	
Address	Library				
└	sendValue	Internal 🔒	🔴		
CATCEO	Implementation	Context, IBEP20, Ownable			
└	<Constructor>	Public ❗	🔴 |NO ❗	
└	name	Public ❗	|NO ❗	
└	symbol	Public ❗	|NO ❗	
└	decimals	Public ❗	|NO ❗	
└	totalSupply	Public ❗	|NO ❗	
└	balanceOf	Public ❗	|NO ❗	
└	allowance	Public ❗	|NO ❗	
└	approve	Public ❗	🔴 |NO ❗	

CONTRACT ASSESMENT

└	transferFrom	Public ❗	🔴	NO❗
└	increaseAllowance	Public ❗	🔴	NO❗
└	decreaseAllowance	Public ❗	🔴	NO❗
└	transfer	Public ❗	🔴	NO❗
└	isExcludedFromReward	Public ❗		NO❗
└	reflectionFromToken	Public ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	excludeFromReward	Public ❗	🔴	onlyOwner
└	includeInReward	External ❗	🔴	onlyOwner
└	excludeFromFee	Public ❗	🔴	onlyOwner
└	includeInFee	Public ❗	🔴	onlyOwner
└	isExcludedFromFee	Public ❗		NO❗
└	_reflectRfi	Private 🔐	🔴	
└	_takeMarketing	Private 🔐	🔴	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_approve	Private 🔐	🔴	
└	_transfer	Private 🔐	🔴	
└	_tokenTransfer	Private 🔐	🔴	
└	swapAndLiquify	Private 🔐	🔴	lockTheSwap
└	swapTokensForBNB	Private 🔐	🔴	
└	bulkExcludeFee	External ❗	🔴	onlyOwner
└	<Receive Ether>	External ❗	💵	NO❗
Node	Implementation	ERC20, UniSwapPoolETH, RatesTracker, SellTracker, Excludes		
└	<Constructor>	Public ❗	🔴	ERC20
└	openTrading	Public ❗	🔴	onlyOwner
└	_transfer	Internal 🔐	🔴	
└	handSwap	Internal 🔐	🔴	
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❗

CONTRACT ASSESMENT

└	approve	Public ❗	🔴	NO❗
└	transferFrom	Public ❗	🔴	NO❗
└	increaseAllowance	Public ❗	🔴	NO❗
└	decreaseAllowance	Public ❗	🔴	NO❗
└	_transfer	Internal 🔒	🔴	
└	_takeTransfer	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 🔒		
RatesTracker	Implementation	ERC20, SwitchTracker		
└	<Constructor>	Public ❗	🔴	NO❗
└	setDivision	Public ❗	🔴	onlyOwner
└	setDistributeAt	Public ❗	🔴	onlyOwner
└	setTo	Public ❗	🔴	onlyOwner
└	setRatesBuy	Public ❗	🔴	onlyOwner
└	setRatesSell	Public ❗	🔴	onlyOwner
└	processBuyFees	Internal 🔒	🔴	
└	processSellFees	Internal 🔒	🔴	
└	distribute	Internal 🔒	🔴	
SwitchTracker	Implementation	Ownable		

CONTRACT ASSESMENT

└	setSwitch	Public ❗	🔴	onlyOwner
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🔴	NO ❗
└	owner	Public ❗		NO ❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🔴	onlyOwner
└	_transferOwnership	Internal 🔒	🔴	
UniSwapPoolETH	Implementation			
└	<Receive Ether>	External ❗	💵	NO ❗
└	createPair	Internal 🔒	🔴	
└	isPair	Internal 🔒		
IRouter	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
└	getAmountsOut	External ❗		NO ❗
IFactory	Interface			
└	createPair	External ❗	🔴	NO ❗
SellTracker	Implementation	ERC20, UniSwapPoolETH, SwitchTracker		
└	recordStartAt	Internal 🔒	🔴	
└	getPrice	Internal 🔒		
└	wrapPrice	Internal 🔒	🔴	
└	sellCheck	Internal 🔒	🔴	
Excludes	Implementation	Ownable		
└	setExclude	Internal 🔒	🔴	
└	setExcludes	Public ❗	🔴	onlyOwner
└	isExcludes	Internal 🔒		
└	setLiquidityer	Public ❗	🔴	onlyOwner
└	isLiquidityer	Internal 🔒		
└	removeLiquidityer	Internal 🔒	🔴	

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
Reentrancy in CATCEO.transferFrom(address,address,uint256) (contracts/Token.sol#251-266):
        External calls:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#256)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
                - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#536-542)
                - address(marketingWallet).sendValue(deltaBalance) (contracts/Token.sol#523)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#256)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (contracts/Token.sol#461)
                - _approve(sender, _msgSender(),currentAllowance - amount) (contracts/Token.sol#263)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

CATCEO.includeInReward(address) (contracts/Token.sol#341-352) has costly operations inside a loop:
        - _excluded.pop() (contracts/Token.sol#348)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#45-48) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

CATCEO._rTotal (contracts/Token.sol#151) is set pre-construction with a non-constant function or state variable:
        - (MAX - (MAX % _tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#6) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.18 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#117-128):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#93) is not in mixedCase
Struct CATCEO.valuesFromGetValues (contracts/Token.sol#175-183) is not in CapWords
Constant CATCEO._decimals (contracts/Token.sol#147) is not in UPPER_CASE_WITH_UNDERSCORES
Constant CATCEO._name (contracts/Token.sol#158) is not in UPPER_CASE_WITH_UNDERSCORES
Constant CATCEO._symbol (contracts/Token.sol#159) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#46)" inContext (contracts/Token.sol#40-49)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

CATCEO._tTotal (contracts/Token.sol#150) should be constant
CATCEO.deadWallet (contracts/Token.sol#155) should be constant
CATCEO.marketingWallet (contracts/Token.sol#156) should be constant
CATCEO.swapTokensAtAmount (contracts/Token.sol#153) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

CATCEO.pair (contracts/Token.sol#145) should be immutable
CATCEO.router (contracts/Token.sol#144) 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 features word as expected

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0xc2d0a90778b6e4374ba64c7a95
40e713f102c87fb49e75f6e43375740f245c36

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xcb2b558b7cc580f030c15e5618
323c64baa9cbc4ac74a946debb42e72020d4c0

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xa432f1212d281f0c45113fac3868
9da712d0a223c7d11fa037deaf603eddee9f

4- Transferring when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x88dbb77d2b380b4480830973c
51dcef0e867526d39ff7e25bded734e81ee9e9c

5- Buying when not excluded (5% tax) (passed):
https://testnet.bscscan.com/tx/0x8292a5d67626832429eb88590
67a6dc21418ef5d1d956d5232680d361cd46b72

6- Selling when not excluded (5% tax) (passed):
https://testnet.bscscan.com/tx/0x2b7c128ac93eeb020ff838446c6
f467bd1b8341908a3031cecda30a789596833

FUNCTIONAL TESTING

7- Transferring when not excluded (5% tax) (passed):
https://testnet.bscscan.com/tx/0x2a67ed838cad208136af21c367c
a9320c05930840927de201c8965bde3d7111e

8- Internal swap (passed):
marketing wallet received ETH
https://testnet.bscscan.com/address/0xca94554ea45d90897f46f
7d12aff87e4f2d997c1#internaltx\

9- Reflections (passed):
0x5d8d71b4b69857bf69c7800cdb38e22f89a5190b received
reflections when was included in rewards, and after excluding it
from rewards, it did not receive rewards anymore.
Also totFeesPaid shows 5501839515207546989003264 (in r
format) for rfi which means reflections work.

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.

Social Media Overview

Here are the Social Media Accounts of CAT CEO

https://t.me/catceo_global

https://twitter.com/CatCeoBSC

http://catceo.io/

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