Cyberscope

Audit Report

Catch

March 2024

Network	BSC
Address	0x1be3c9d5531fa944f285256a348ab067f277127d
Audited by	© cyberscope

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	ILC	Inefficient Loop Checks	Unresolved
●	MEM	Missing Error Messages	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved

Table of Contents

Analysis **1**
Diagnostics **2**
Table of Contents **3**
Review **4**
Audit Updates 4
Source Files 5
Findings Breakdown **6**
ILC - Inefficient Loop Checks 7
Description 7
Recommendation 7
MEM - Missing Error Messages 9
Description 9
Recommendation 9
L04 - Conformance to Solidity Naming Conventions 10
Description 10
Recommendation 11
Functions Analysis **12**
Inheritance Graph **18**
Flow Graph **19**
Summary **20**
Disclaimer **21**
About Cyberscope **22**

Review

Contract Name	CATCH
Compiler Version	v0.8.19+commit.7dd6d404
Optimization	200 runs
Explorer	https://bscscan.com/address/0x1be3c9d5531fa944f285256a348ab067f277127d
Address	0x1be3c9d5531fa944f285256a348ab067f277127d
Network	BSC
Symbol	CATCH
Decimals	18
Total Supply	89,998,068.556
Badge Eligibility	Yes

Audit Updates

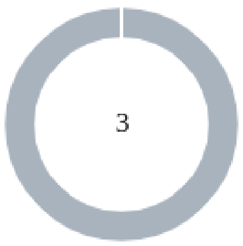
Initial Audit	15 Jan 2024 https://github.com/cyberscope-io/audits/blob/main/catchcoin/v1/audit.pdf
Corrected Phase 2	23 Jan 2024 https://github.com/cyberscope-io/audits/blob/main/catchcoin/v2/audit.pdf
Corrected Phase 3	28 Mar 2024 https://github.com/cyberscope-io/audits/blob/main/catchcoin/v3/audit.pdf

Corrected Phase 4	06 Apr 2024

Source Files

Filename	SHA256
contracts/CATCH.sol	674c276153d6224151579745bb59d437f704cf0b1dc393153b8921d8626c6b1a

Findings Breakdown



🔴	Critical	0
🟠	Medium	0
⚪	Minor / Informative	3

Severity		Unresolved	Acknowledged	Resolved	Other
🔴	Critical	0	0	0	0
🟠	Medium	0	0	0	0
⚪	Minor / Informative	3	0	0	0

ILC - Inefficient Loop Checks

Criticality	Minor / Informative
Location	CATCHCOIN.sol#L1107
Status	Unresolved

Description

The contract is structured to perform an airdrop by iterating through two separate for loops. The primary purpose of the first loop is to aggregate the total amount of tokens (`airCapacity`) intended for the airdrop. This total is then compared against the balance of the message sender (`msg.sender`) to ensure they have enough tokens to complete the airdrop. The second loop conducts the actual token transfer process. This structure, while functionally correct, introduces inefficiency by necessitating two full iterations over the arrays, even though the preliminary check for sufficient balance could be integrated into a single loop structure. This inefficiency not only increases the gas cost but also complicates the contract's logic unnecessarily.

```
  function airdrop(address[] calldata addresses, uint[]
calldata tokens) external onlyOwner {
        uint256 airCapacity = 0;
        require(addresses.length == tokens.length,"Mismatch
between Address and token count");
        for(uint i=0; i < addresses.length; i++){
            uint amount = tokens[i];
            airCapacity = airCapacity + amount;
        }
        require(balanceOf(msg.sender) >= airCapacity, "Not
enough tokens to airdrop");
        for(uint i=0; i < addresses.length; i++){
            uint amount = tokens[i];

_tokenTransfer(msg.sender,addresses[i],amount,false);
        }
    }
```

Recommendation

It is recommended to consolidate the two separate loops into a single loop to enhance efficiency and reduce gas costs. By performing the token amount aggregation and the balance check in one iteration, the contract can minimize the computational overhead associated with loop execution. This can be achieved by aggregating the `airCapacity` as the loop progresses and conducting the balance check after completing all transfers, ensuring that the `msg.sender` has a sufficient balance throughout the process. Such a refactor not only streamlines the code, making it more readable and maintainable, but also optimizes the contract's execution by reducing the number of iterations required to complete the airdrop function.

MEM - Missing Error Messages

Criticality	Minor / Informative
Location	CATCHCOIN.sol#L1069
Status	Unresolved

Description

The contract is missing error messages. These are no error messages to accurately reflect the problem, making it difficult to identify and fix the issue. As a result, the users will not be able to find the root cause of the error.

```
require(tradeEnabled)
```

Recommendation

The team is suggested to provide a descriptive message to the errors. This message can be used to provide additional context about the error that occurred or to explain why the contract execution was halted. This can be useful for debugging and for providing more information to users that interact with the contract.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	CATCHCOIN.sol#L196,717,729,745,945,956,968,981
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
function WETH() external pure returns (address);
function setFundWallet(address _fundWallet) external onlyOwner
function setSwapAndLiquifyEnabled(bool _enabled) external onlyOwner
function updateThreshold(uint256 _amount) external onlyOwner
function calculateTaxFee(uint256 _amount) private view returns (uint256)
function calculateLiquidityFee(uint256 _amount) private view returns
(uint256)
function calculateCoinOperartionTax(uint256 _amount) private view
returns (uint256)
function calculateBurnTax(uint256 _amount) private view returns

(uint256)
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-

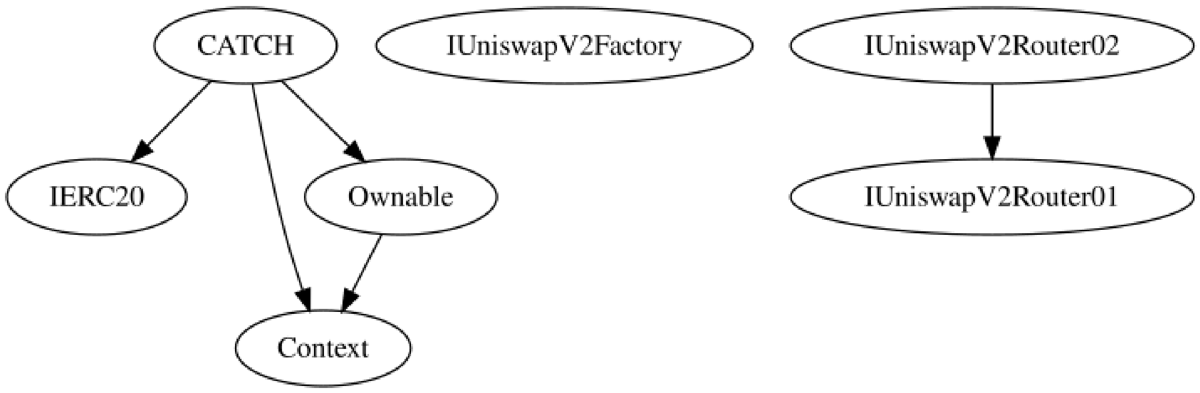
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
CATCHCOIN	Implementation	Context, IERC20, Ownable		
		Public	✓	-
	name	External		-
	symbol	External		-
	decimals	External		-

	totalSupply	External		-
	balanceOf	Public		-
	transfer	External	✓	-
	allowance	External		-
	approve	Public	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	isExcludedFromReward	External		-
	totalFees	External		-
	deliver	External	✓	-
	reflectionFromToken	External		-
	tokenFromReflection	Public		-
	excludeFromReward	External	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	_transferBothExcluded	Private	✓	
	excludeFromFee	External	✓	onlyOwner
	includeInFee	External	✓	onlyOwner
	setFundWallet	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	External	✓	onlyOwner
	updateThreshold	External	✓	onlyOwner
		External	Payable	-
	_reflectFee	Private	✓	

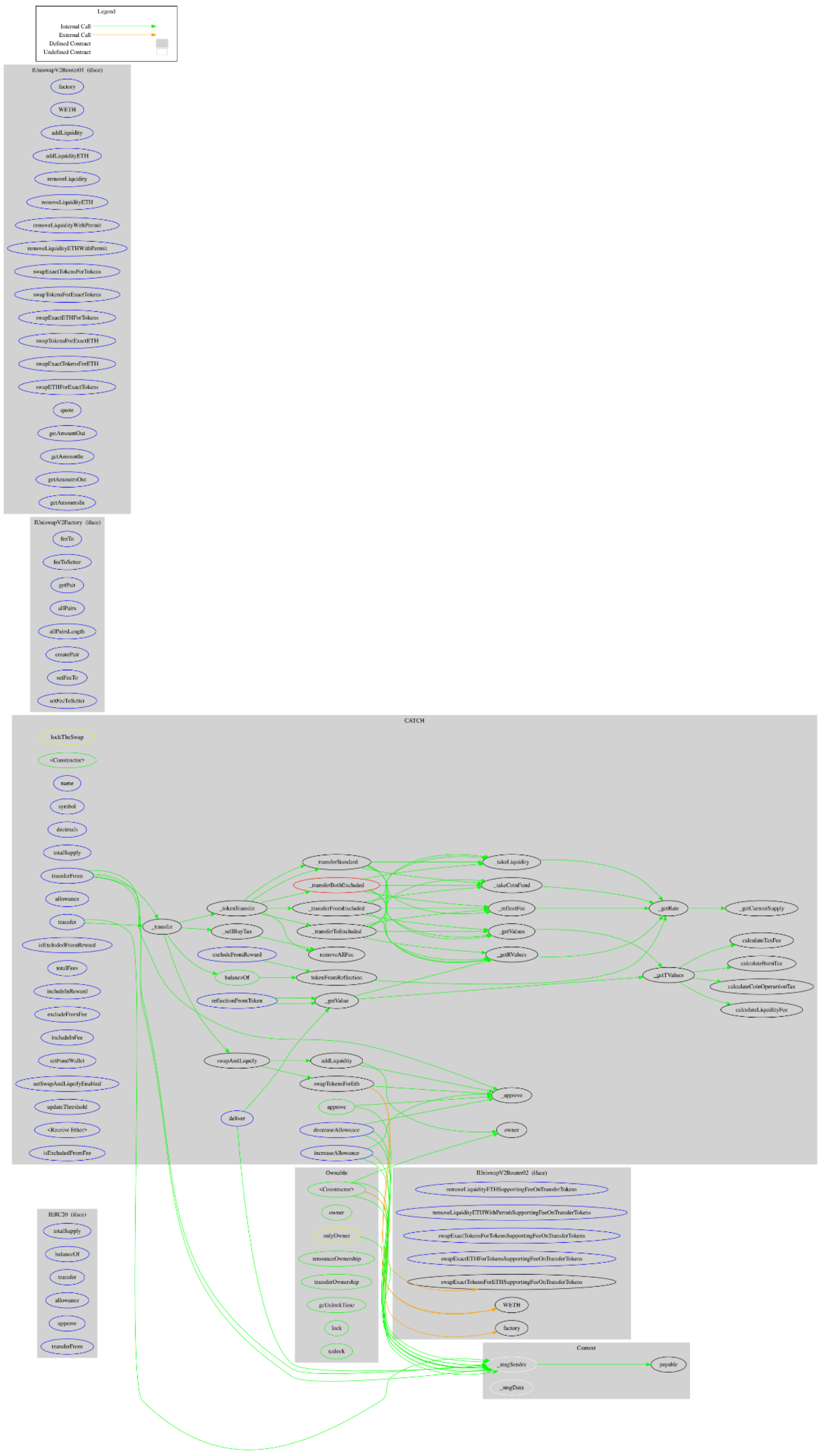
	_takeCoinFund	Private	✓	
	_getValues	Private		
	_getValue	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	calculateCoinOperartionTax	Private		
	calculateBurnTax	Private		
	removeAllFee	Private	✓	
	isExcludedFromFee	External		-
	_approve	Private	✓	
	startTrading	External	✓	onlyOwner
	_transfer	Private	✓	
	airdrop	External	✓	onlyOwner
	_sellBuyTax	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	

	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	

Inheritance Graph



Flow Graph



Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IUniswapV2Router01 (iface)
factory
WETH
addLiquidity
addLiquidityETH
removeLiquidity
removeLiquidityETH
removeLiquidityWithPermit
removeLiquidityETHWithPermit
swapExactTokensForTokens
swapTokensForExactTokens
swapExactETHForTokens
swapTokensForExactETH
swapExactTokensForETH
swapETHForExactTokens
quote
getAmountOut
getAmountIn
getAmountsOut
getAmountsIn
IUniswapV2Factory (iface)
feeTo
feeToSetter
getPair
allPairs
allPairsLength
createPair
setFeeTo
setFeeToSetter
CATCH
lockTheSwap
<Constructor>
name
symbol
decimals
totalSupply
transferFrom
allowance
transfer
isExcludedFromReward
totalFees
includeInReward
excludeFromFee
includeInFee
setFundWallet
setSwapAndLiquifyEnabled
updateThreshold
<Receive Ether>
isExcludedFromFee
_transfer
_tokenTransfer
_sellBuyTax
excludeFromReward
balanceOf
reflectionFromToken
transferStandard
_transferBothExcluded
_transferFromExcluded
_transferToExcluded
removeAllFee
tokenFromReflection
_getValue
_takeLiquidity
_takeCoinFund
_reflectFee
_getValues
_getRValues
_getRate
_getCurrentSupply
_getTValues
calculateTaxFee
calculateBurnTax
calculateCoinOperartionTax
calculateLiquidityFee
swapAndLiquify
addLiquidity
swapTokensForEth
approve
decreaseAllowance
increaseAllowance
deliver
_approve
owner
Ownable
<Constructor>
owner
onlyOwner
renounceOwnership
transferOwnership
getUnlockTime
lock
unlock
IUniswapV2Router02 (iface)
removeLiquidityETHSupportingFeeOnTransferTokens
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens
swapExactTokensForTokensSupportingFeeOnTransferTokens
swapExactETHForTokensSupportingFeeOnTransferTokens
swapExactTokensForETHSupportingFeeOnTransferTokens
WETH
factory
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Context
_msgSender
_msgData
payable

Summary

Catch contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. Catch is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of a 3% fee on buy transactions and a 6% fee on sell transactions.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope’s prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io