Cyberscope

Audit Report

Catch

January 2024

Repository https://github.com/EtherAuthority/Smart-Contracts-Library/tree/main

Commit 59010731b26cd22cbd3eb3733af12ae7ae752a05

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
Critical	IOE	Incompatible Opcode Error	Unresolved
Minor / Informative	PVC	Price Volatility Concern	Unresolved
Minor / Informative	L16	Validate Variable Setters	Unresolved

Table of Contents

Review

Contract Name	CATCH
Repository	https://github.com/EtherAuthority/Smart-Contracts-Library/tree/main
Commit	59010731b26cd22cbd3eb3733af12ae7ae752a05
Testing Deploy	https://testnet.bscscan.com/address/0xc3c7b36991116582675bec3ce4def1ecdf3e3df2
Symbol	CATCH
Decimals	18
Total Supply	90,000,000
Badge Eligibility	Yes

Audit Updates

Initial Audit	15 Jan 2024
Corrected Phase 2	23 Jan 2024

Source Files

Filename	SHA256
contracts/CATCH.sol	9065f951a59d80108d9205afb40647260fb9260c2d00b8dfa11a4384c8de1351

Findings Breakdown

Critical	1
Medium	0
Minor / Informative	2

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	1	0	0	0
Medium	0	0	0	0
Minor / Informative	2	0	0	0

IOE - Incompatible Opcode Error

Criticality	Critical
Location	catch.sol
Status	Unresolved

Description

During the testnet deployment of the contract, we encountered the error `ProviderError: invalid opcode: PUSH0`. This problem arises because Solidity version `0.8.22`, which the contract currently utilizes, introduces the `PUSH0` opcode (0x5f). However, this opcode even if is supported on the Ethereum (ETH) mainnet is not recognized by other blockchain networks. The contract sets the `_uniswapV2Router` variable to the address `0x10ED43C718714eb63d5aA57B78B54704E256024E`, which is the address of the `UniswapV2Router` on the Binance Smart Chain. As a result, when attempting to deploy the contract to the `BSC` network, the unrecognized `PUSH0` opcode leads to the incomplete deployment of the contract.

```
pragma solidity 0.8.22;
...
IUniswapV2Router02 _uniswapV2Router =
IUniswapV2Router02(0x10ED43C718714eb63d5aA57B78B54704E256024E); //BNB
Smart Chain Mainnet
```

```
ProviderError: invalid opcode: PUSH0
```

Recommendation

It is recommended to downgrade the Solidity version used in the contract to pragma `0.8.19` instead of `0.8.22`. This adjustment is essential to ensure compatibility with the Binance Smart Chain (BSC), as indicated by the contract's usage of the Uniswap v2 router value. By reverting to Solidity version `0.8.19`, the contract will avoid the `PUSH0` opcode issue, as this version does not include the opcode. This change will facilitate the

successful deployment and operation of the contract on the BSC network, in addition to other chains that do not support the `PUSH0` opcode introduced in Solidity `0.8.22` .

PVC - Price Volatility Concern

Criticality	Minor / Informative
Location	catch.sol#L762,1076
Status	Unresolved

Description

The contract accumulates tokens from the taxes to swap them for ETH. The variable `numTokensSellToAddToLiquidity` sets a threshold where the contract will trigger the swap functionality. If the variable is set to a big number, then the contract will swap a huge amount of tokens for ETH.

It is important to note that the price of the token representing it, can be highly volatile. This means that the value of a price volatility swap involving Ether could fluctuate significantly at the triggered point, potentially leading to significant price volatility for the parties involved.

```
    //set numTokensSellToAddToLiquidity value
    function updateThreshold(uint256 _amount) external onlyOwner {
        require(_amount > 0,"amount is not valid");
        numTokensSellToAddToLiquidity = _amount;
        emit ThresholdUpdated(_amount);


        bool overMinTokenBalance = contractTokenBalance >=
numTokensSellToAddToLiquidity;
        if (
            overMinTokenBalance &&
            !inSwapAndLiquify &&
            from != uniswapV2Pair &&
            swapAndLiquifyEnabled
        ) {
            contractTokenBalance = numTokensSellToAddToLiquidity;
            //add liquidity
            swapAndLiquify(contractTokenBalance);
        }
```

Recommendation

The contract could ensure that it will not sell more than a reasonable amount of tokens in a single transaction. A suggested implementation could check that the maximum amount should be less than a fixed percentage of the exchange reserves. Hence, the contract will guarantee that it cannot accumulate a huge amount of tokens in order to sell them.

L16 - Validate Variable Setters

Criticality	Minor / Informative
Location	contracts/CATCH.sol#L413
Status	Unresolved

Description

The contract performs operations on variables that have been configured on user-supplied input. These variables are missing of proper check for the case where a value is zero. This can lead to problems when the contract is executed, as certain actions may not be properly handled when the value is zero.

```
fundWallet = _fundWallet
```

Recommendation

By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

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
	geUnlockTime	Public		-
	lock	Public	✓	onlyOwner

	unlock	Public	✓	-
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	✓	-

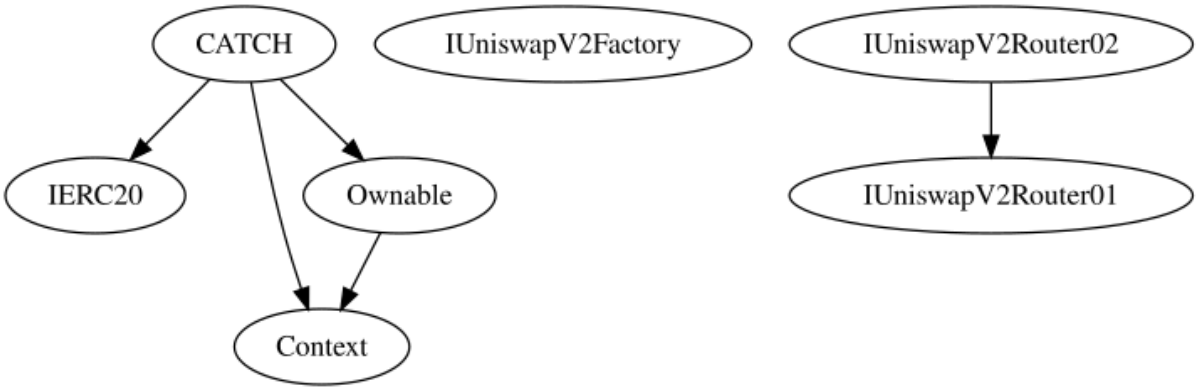
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2 Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
CATCH	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	-

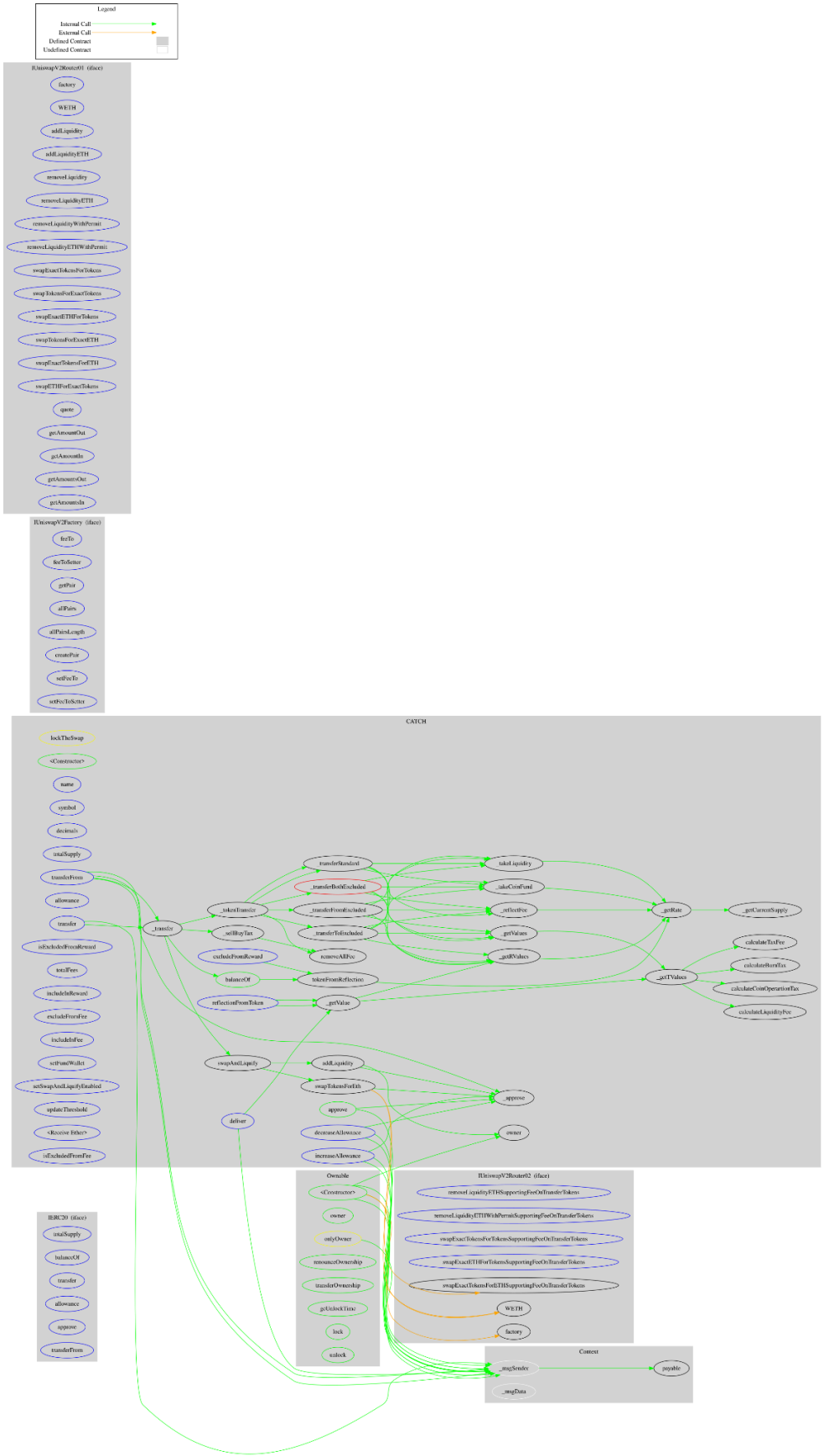
	_reflectFee	Private	✓	
	_takeCoinFund	Private	✓	
	_getValues	Private		
	_getValue	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	calculateCoinOperartionTax	Private		
	calculateBurnTax	Private		
	removeAllFee	Private	✓	
	isExcludedFromFee	External		-
	_approve	Private	✓	
	_transfer	Private	✓	
	_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
setPrizeWallet
setSwapAndLiquifyEnabled
updateThreshold
<Receive Ether>
isExcludedFromFee
_transfer
_tokenTransfer
_sellBuyTax
excludeFromReward
balanceOf
reflectionFromToken
transferStandard
_transferBothExcluded
_transferFromExcluded
_transferToExcluded
removeAllFee
tokenFromReflection
takeLiquidity
_takeCoinFund
_reflectFee
getValues
_getRValues
_getRate
_getCurrentSupply
_getTValues
calculateTaxFee
calculateBurnTax
calculateCoinOperationTax
calculateLiquidityFee
_getValue
swapAndLiquify
addLiquidity
swapTokensForEth
approve
decreaseAllowance
increaseAllowance
_approve
owner
deliver
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Ownable
<Constructor>
owner
onlyOwner
renounceOwnership
transferOwnership
getUnlockTime
lock
unlock
IUniswapV2Router02 (iface)
removeLiquidityETHSupportingFeeOnTransferTokens
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens
swapExactTokensForTokensSupportingFeeOnTransferTokens
swapExactETHForTokensSupportingFeeOnTransferTokens
swapExactTokensForETHSupportingFeeOnTransferTokens
WETH
factory
Context
_msgSender
payable
_msgData

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 one critical issue during the contract deployment. 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