

SKELETON ECOSYSTEM

SMART CONTRACT AUDIT



SLAPPY CAT COIN

BEP 20

0x6c637c67a1dc3dbe33872befb093617c3af22059



Table of Contents

Table of Contents	1
Disclaimer	2
Overview	3
Creation/Audit Date	3
Verified Socials	3
Contract Functions Analysis	4
Contract Safety and Weakness	6
Detected Vulnerabilities	8
Contract Flow Chart	19
Inheritance Graph	20
Contract Descriptions	21

Global Disclaimer

This document serves as a disclaimer for the crypto smart contract audit conducted by Skeleton Ecosystem. The purpose of the audit was to review the codebase of the smart contracts for potential vulnerabilities and issues. It is important to note the following:

Limited Scope: The audit is based on the code and information available up to the audit completion date. It does not cover external factors, system interactions, or changes made after the audit. The audit itself can not guarantee 100% safety and can not detect common scam methods like farming and developer sell-out.

No Guarantee of Security: While we have taken reasonable steps to identify vulnerabilities, it is impossible to guarantee the complete absence of security risks or issues. The audit report provides an assessment of the contract's security as of the audit date.

Continued Development: Smart contracts and blockchain technology are evolving fields. Updates, forks, or changes to the contract post-audit may introduce new risks that were not present during the audit.

Third-party Code: If the smart contract relies on third-party libraries or code, those components were not thoroughly audited unless explicitly stated. Security of these dependencies is the responsibility of their respective developers.

Non-Exhaustive Testing: The audit involved automated analysis, manual review, and testing under controlled conditions. It is possible that certain vulnerabilities or issues may not have been identified.

Risk Evaluation: The audit report includes a risk assessment for identified vulnerabilities. It is recommended that the development team carefully reviews and addresses these risks to mitigate potential exploits.

Not Financial Advice: This audit report is not intended as financial or investment advice. Decisions regarding the use, deployment, or investment in the smart contract should be made based on a comprehensive assessment of the associated risks.

By accessing and using this audit report, you acknowledge and agree to the limitations outlined above. Skeleton Ecosystem and its auditors shall not be held liable for any direct or indirect damages resulting from the use of the audit report or the smart contract itself.

Please consult with legal, technical, and financial professionals before making any decisions related to the smart contract.

Overview

Contract Name	SLAPPY CAT COIN
Ticker/Symbol	SLAPPYCAT
Blockchain	Binance Smart Chain Bep20
Contract Address	0x6c637c67a1dc3dbe33872befb093617c3af22059
Creator Address	0xC718D463d6ed2c915d07055DC298806fB5a7Dd5a
Current Owner Address	Renounced
Contract Explorer	https://bscscan.com/token/0x6c637c67a1dc3dbe33872befb093617c3af22059
Compiler Version	v0.8.7+commit.e28d00a7
License	MIT License
Optimisation	Yes with 200 Runs
Total Supply	420,690,000,000,000 SLAPPYCAT
Decimals	9

Creation/Audit

Contract Deployed	31 Jul 2023
Audit Created	20-Aug-23 09:00:00 UTC
Audit Update	V 0.1

Verified Socials

Website	https://www.slappycatcoin.live/
Telegram	https://t.me/SlappyCatCoinGlobal
X	https://twitter.com/SlappyCatCoin



Contract Function Analysis



Pass



Attention Item






























Risky Item

Pass

Attention

Risk

Contract Verified		The contract source code is uploaded to blockchain explorer and is open source, so everybody can read it.
Contract Ownership		The ownership of the contract was sent to dead address. With this the owner eliminates he's rights to modify the contract. The owner can not set any of the functions anymore.
Buy Tax	0 %	Shows the taxes for purchase transactions. Above 10% may be considered a high tax rate. More than 50% tax rate means may not be tradable. Contract renounced so tax rate is fixed. 
Sell Tax	6 %	Shows the taxes for sell transactions. Above 10% may be considered a high tax rate. More than 50% tax rate means may not be tradable. Contract renounced so tax rate is fixed. 
Honeypot Analyse		Holder is able to buy and sell. If honeypot: The contract blocks sell transfer from holder wallet. Multiple events may cause honeypot. Trading disabled, extremely high tax
Liquidity Status		Locked on 20.08.2023: 98% for 349 days on Pinksale locker Note! Initial liquidity tokens scanned. For new LP Lockers allways re-check with skeleton scanner on telegram.
Trading Disable Functions		No trading suspendable function found. If a suspendable code is included, the token maybe neither be bought or sold (honeypot risk). If contract is renounced this function can't be used.  If there is authorised hidden owner, or there is Retrieve Ownership Function, the trading disable function may be used!
Set Fees function		No Fee Setting function found. The contract owner may contain the authority to modify the transaction tax. If the transaction tax is increased to more than 49%, the tokens may not be able to be traded (honeypot risk). If contract is renounced this function can't be used.  If there is authorised hidden owner, or there is Retrieve Ownership Function, the set fees function may be used!
Proxy Contract		The proxy contract means contract owner can modify the function of the token and possibly effect the price. The Owner is not the creator but the creator may have authorisation to change functions.
Mint Function		No mint function found. Mint function is transparent or non-existent. Hidden mint functions may increase the amount of tokens in circulation and effect the price of the token. Owner can mint new tokens and sell. If contract is renounced this function can't be used.

Balance Modifier Function		<p>No Balance Modifier function found.</p> <p>If there is a function for this, the contract owner can have the authority to modify the balance of tokens at other addresses. For example revoke the bought tokens from the holders wallet. Common form of scam: You buy the token, but it's disappearing from your wallet.</p> <p> If contract is renounced this function still can be used as auto self Destruct</p>
Whitelist Function		<p>Whitelist Function Found.</p> <p>If there is a function for this Developer can set zero fee or no max wallet size for addresses (for example team wallets can trade without fee. Can cause farming)</p> <p>If there is a whitelist, some addresses may not be able to trade normally (honeypot risk).  Renounced, this function can not be used.</p>
Hidden Owner Analysis		<p>No authorised hidden owner found.</p> <p>For contract with a hidden owner, developer can still manipulate the contract even if the ownership has been abandoned. Fake renounce.</p>
Retrieve Ownership Function		<p>No functions found which can retrieve ownership of the contract.</p> <p>If this function exists, it is possible for the project owner to regain ownership even after relinquishing it. Also known as fake renounce.</p>
Self Destruct Function		<p>No Self Destruct function found.</p> <p>If this function exists and is triggered, the contract will be destroyed, all functions will be unavailable, and all related assets will be erased.</p>
Specific Tax Changing Function		<p>Specific Tax Changing Functions found.</p> <p> Renounced, this function can not be used.</p> <p>If it exists, the contract owner may set a very outrageous tax rate for assigned address to block it from trading. Can assign all wallets at once!</p>
Trading Cooldown Function		<p>Trading Cooldown Function found.</p> <p> Renounced, this function can not be used.</p> <p>If there is a trading cooldown function, the user will not be able to sell the token within a certain time or block after buying. Like a temporary honeypot.</p>
Max Transaction and Holding Modify Function		<p>Max Transaction and Holding Modify function found.</p> <p> Renounced, this function can not be used.</p> <p>If there is a function for this, the maximum trading amount or maximum position can be modified. Can cause honeypot</p>
Transaction Limiting Function		<p>Transaction Limiter Function Found.</p> <p> Renounced, this function can not be used.</p> <p>The number of overall token transactions may be limited (honeypot risk)</p>

Contract Safety and Weakness

Unchecked Transfer

A transfer call made in this contract may be unstable and cause tokens to become stuck

Uninitialized Local Variables

This contract's local variables are not all initialized, potentially resulting in lost funds or other exploits.

Function Initializing State Variables

Some state variables in this contract can be set through dynamic functions calls, which may be unsafe.

Low Level Calls

This contract uses low level calls, which may be unsafe.

Costly Loop Operations

Some looping logic in this contract results in very high gas usage.

Numeric Notation Best Practices

The numeric notation used in this contract is unconventional, possibly worsening the reading/debugging experience.

State Variables Should be Declared Constant

Some state variables in this contract should be declared as constant.

Public Functions Should be Declared External

Some functions in this contract should be declared as external in order to save gas.

No compiler version inconsistencies found

No unchecked call responses found

No vulnerable self-destruct functions found

No assertion vulnerabilities found

No old solidity code found

No external delegated calls found

No external call dependency found

No vulnerable authentication calls found

No invalid character typos found

No RTL characters found

No dead code found

No risky data allocation found

No uninitialized state variables found

No uninitialized storage variables found

No vulnerable initialization functions found

No risky data handling found

No number accuracy bug found

No out-of-range number vulnerability found

No map data deletion vulnerabilities found

No tautologies or contradictions found

No faulty true/false values found

No innacurate divisions found

No redundant constructor calls found

No vulnerable return values found

No default function responses found

No missing arithmetic events found

No missing access control events found

No missing zero address checks found

No redundant true/false comparisons found

No invalid solidity versions found


No vulnerable payable functions found

No vulnerable message values found

Detected High Severity Vulnerabilities


⚠️ Unchecked Transfer (1 item)

A transfer call made in this contract may be unstable and cause tokens to become stuck

Function	Severity	Relevant Snippet
SlappyCat.claimStuckTokens (address) (SlappyCat.sol#563-572) ignores return value by ERC20token.transfer(msg.se nder,balance) (SlappyCat.sol#571)	 Severity : High	<pre> function claimStuckTokens(address token) external onlyOwner { require(token != address(this), "Owner cannot claim native tokens"); if (token == address(0x0)) { payable(msg.sender).sendValue(address(this).balance); return; } IERC20 ERC20token = IERC20(token); uint256 balance = ERC20token.balanceOf(address(t his)); ERC20token.transfer(msg.sender , balance); } </pre>


⚠ Uninitialized Local Variables (1 Item)


This contract's local variables are not all initialized, potentially resulting in lost funds or other exploits.

Function	Severity	Relevant Snippet
Issue Location in Code SlappyCat.constructor().router (SlappyCat.sol#402) is a local variable never initialized	 Severity : High	address router;

⚠ Function Initializing State Variables (2 Items)



Some state variables in this contract can be set through dynamic functions calls, which may be unsafe

Function	Severity	Relevant Snippet
Issue Location in Code SlappyCat._tTotal (SlappyCat.sol#357) is set pre-construction with a non-constant function or state variable: - 420_690_000_000_000 * (10 ** _decimals)	 Severity : High	uint256 private _tTotal = 420_690_000_000_000 * (10 ** _decimals);

Issue Location in Code SlappyCat._rTotal (SlappyCat.sol#358) is set pre-construction with a non-constant function or state variable: - (MAX - (MAX % _tTotal))	 Severity : High	uint256 private _rTotal = (MAX - (MAX % _tTotal));
--	---	---

Low Level Calls (2 Items)


This contract uses low level calls, which may be unsafe.

Function	Severity	Relevant Snippet
Issue Location in Code Low level call in Address.sendValue(address, uint256) (SlappyCat.sol#91-97): - (success) = recipient.call{value: amount}() (SlappyCat.sol#95)	 Severity : High	function sendValue(address payable recipient, uint256 amount) internal returns(bool){ require(address(this).balance >= amount, "Address: insufficient balance"); // solhint-disable-next- line avoid-low-level-calls, avoid-call-value (bool success,) = recipient.call{ value: amount }(""); return success; }
Issue Location in Code Low level call in Address._functionCallWithV alue(address,bytes,uint256 ,string) (SlappyCat.sol#116-137): - (success, returndata) = target.call{value: weiValue}(data) (SlappyCat.sol#120)	 Severity : High	function _functionCallWithValue(address target, bytes memory data, uint256 weiValue, string memory errorMessage) private returns (bytes memory) { require(isContract(target), "Address: call to non- contract"); // solhint-disable-next- line avoid-low-level-calls

		<pre> (bool success, bytes memory returndata) = target.call{ value: weiValue }(data); if (success) { return returndata; } else { // Look for revert reason and bubble it up if present if (returndata.length > 0) { // The easiest way to bubble the revert reason is using memory via assembly // solhint- disable-next-line no-inline- assembly assembly { let returndata_size := mload(returndata) revert(add(32, returndata), returndata_size) } } else { revert(errorMessage); } } } </pre>
--	--	---


⚠ Costly Loop Operation (1 Item)

Following looping logic in this contract results in very high gas usage.

Function	Severity	Relevant Snippet
Issue Location in Code SlappyCat.includeInReward(address) (SlappyCat.sol#548-559) has costly operations inside a loop: - _excluded.pop() (SlappyCat.sol#555)	 Severity : High	<pre>function includeInReward(address account) external onlyOwner() { require(!_isExcluded[account], "Account is already included"); for (uint256 i = 0; i < _excluded.length; i++) { if (_excluded[i] == account) { _excluded[i] = _excluded[_excluded.length - 1]; _tOwned[account] = 0; _isExcluded[account] = false; _excluded.pop(); break; } } }</pre>




⚠ Numeric Notation Best Practices (1 Item)

The numeric notation used in this contract is unconventional, possibly worsening the reading/debugging experience.

Function	Severity	Relevant Snippet
Contract SlappyCat uses literals with too many digits: <pre> - DEAD = 0x0000000000000000000000000000 000000000000dEaD (SlappyCat.sol#381) </pre>	 Severity : High	<pre> address private constant DEAD = 0x00000000000000000000000000000000 00000000dEaD; </pre>




State Variables Should be Declared Constant (3 Items)





Following state variables in this contract should be declared as constant.





Function	Severity	Relevant Snippet
SlappyCat._decimals (SlappyCat.sol#354) should be constant	 Severity : High	uint8 private _decimals = 9;
SlappyCat._name (SlappyCat.sol#352) should be constant	 Severity : High	string private _name = "Slappy Cat Coin";
SlappyCat._symbol (SlappyCat.sol#353) should be constant	 Severity : High	string private _symbol = "SLAPPYCAT";




Public Functions Should be Declared External (15 Items)


Following functions in this contract should be declared as external in order to save gas.

Function	Severity	Relevant Snippet
name() should be declared external: - SlappyCat.name() (SlappyCat.sol#454-456)	 Severity : High	function name() public view returns (string memory) { return _name; }
symbol() should be declared external: - SlappyCat.symbol() (SlappyCat.sol#458-460)	 Severity : High	function symbol() public view returns (string memory) { return _symbol; }
decimals() should be declared external: - SlappyCat.decimals() (SlappyCat.sol#462-464)	 Severity : High	function decimals() public view returns (uint8) { return _decimals; }

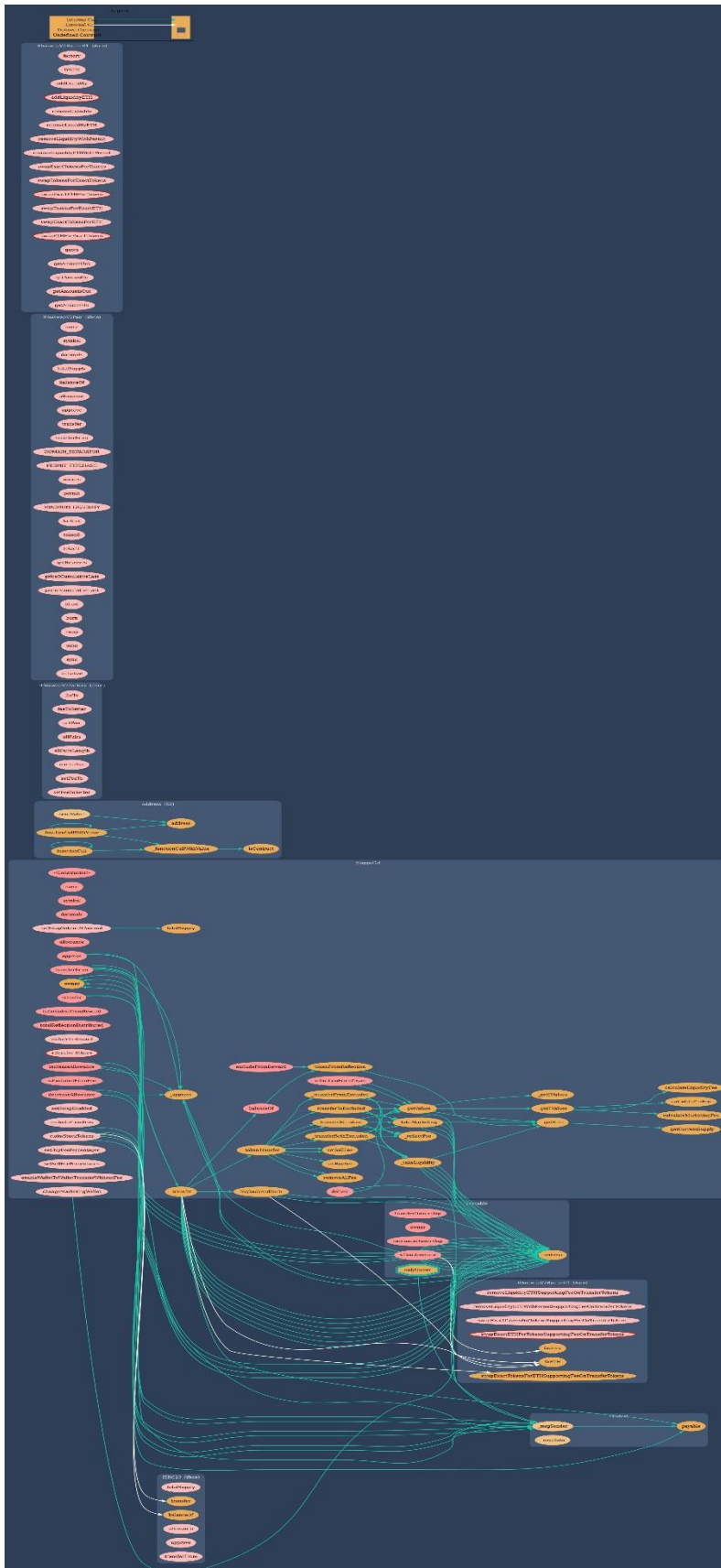
<p>Issue Location in Code</p> <p>transfer(address,uint256) should be declared external:</p> <p>-</p> <p>SlappyCat.transfer(address,uint256) (SlappyCat.sol#475-478)</p>	 Severity : High	<pre>function transfer(address recipient, uint256 amount) public override returns (bool) { _transfer(_msgSender(), recipient, amount); return true; }</pre>
<p>allowance(address,address) should be declared external:</p> <p>-</p> <p>SlappyCat.allowance(address,address) (SlappyCat.sol#480-482)</p>	 Severity : High	<pre>function allowance(address owner, address spender) public view override returns (uint256) { return _allowances[owner][spender]; }</pre>
<p>Issue Location in Code</p> <p>approve(address,uint256) should be declared external:</p> <p>-</p> <p>SlappyCat.approve(address,uint256) (SlappyCat.sol#484-487)</p>	 Severity : High	<pre>function approve(address spender, uint256 amount) public override returns (bool) { _approve(_msgSender(), spender, amount); return true; }</pre>
<p>Issue Location in Code</p> <p>transferFrom(address,address,uint256) should be declared external:</p> <p>-</p> <p>SlappyCat.transferFrom(address,address,uint256) (SlappyCat.sol#489-493)</p>	 Severity : High	<pre>function transferFrom(address sender, address recipient, uint256 amount) public override returns (bool) { _transfer(sender, recipient, amount); _approve(sender, _msgSender(), _allowances[sender][_msgSender ()] - amount); return true; }</pre>

<p>Issue Location in Code</p> <p>increaseAllowance(address, uint256) should be declared external:</p> <p>-</p> <p>SlappyCat.increaseAllowance(address, uint256) (SlappyCat.sol#495-498)</p>	<p> Severity : High</p>	<pre>function increaseAllowance(address spender, uint256 addedValue) public virtual returns (bool) { _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue); return true; }</pre>
<p>Issue Location in Code</p> <p>decreaseAllowance(address, uint256) should be declared external:</p> <p>-</p> <p>SlappyCat.decreaseAllowance(address, uint256) (SlappyCat.sol#500-503)</p>	<p> Severity : High</p>	<pre>function decreaseAllowance(address spender, uint256 subtractedValue) public virtual returns (bool) { _approve(_msgSender(), spender, _allowances[_msgSender()][spender] - subtractedValue); return true; }</pre>
<p>isExcludedFromReward(address) should be declared external:</p> <p>-</p> <p>SlappyCat.isExcludedFromReward(address) (SlappyCat.sol#505-507)</p>	<p> Severity : High</p>	<pre>function isExcludedFromReward(address account) public view returns (bool) { return _isExcluded[account]; }</pre>
<p>totalReflectionDistributed() should be declared external:</p> <p>-</p> <p>SlappyCat.totalReflectionDistributed() (SlappyCat.sol#509-511)</p>	<p> Severity : High</p>	<pre>function totalReflectionDistributed() public view returns (uint256) { return _tFeeTotal; }</pre>

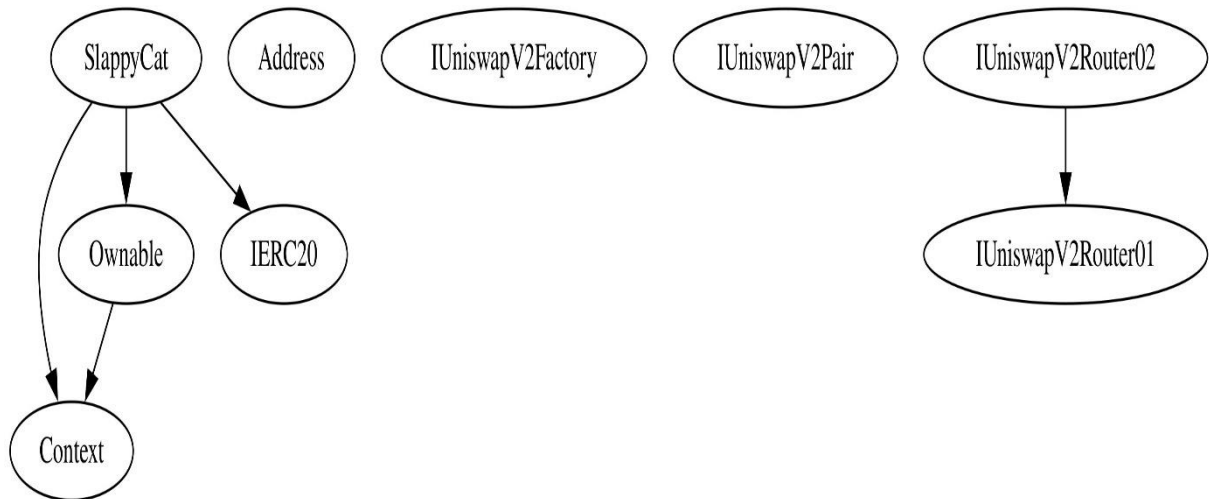
<p>Issue Location in Code</p> <p>deliver(uint256) should be declared external:</p> <p>-</p> <p>SlappyCat.deliver(uint256) (SlappyCat.sol#513-520)</p>	<p> Severity : High</p>	<pre>function deliver(uint256 tAmount) public { address sender = _msgSender(); require(!_isExcluded[sender], "Excluded addresses cannot call this function"); (uint256 rAmount,,,,,) = _getValues(tAmount); _rOwned[sender] = _rOwned[sender] - rAmount; _rTotal = _rTotal - rAmount; _tFeeTotal = _tFeeTotal + tAmount; }</pre>
<p>Issue Location in Code</p> <p>reflectionFromToken(uint256,bool) should be declared external:</p> <p>-</p> <p>SlappyCat.reflectionFromToken(uint256,bool) (SlappyCat.sol#522-531)</p>	<p> Severity : High</p>	<pre>function reflectionFromToken(uint256 tAmount, bool deductTransferFee) public view returns(uint256) { require(tAmount <= _tTotal, "Amount must be less than supply"); if (!deductTransferFee) { (uint256 rAmount,,,,,) = _getValues(tAmount); return rAmount; } else { (,uint256 rTransferAmount,,,,,) = _getValues(tAmount); return rTransferAmount; } }</pre>
<p>excludeFromReward(address) should be declared external:</p> <p>-</p> <p>SlappyCat.excludeFromReward(address) (SlappyCat.sol#539-546)</p>	<p> Severity : High</p>	<pre>function excludeFromReward(address account) public onlyOwner() { require(!_isExcluded[account], "Account is already excluded"); if(_rOwned[account] > 0) { _tOwned[account] = tokenFromReflection(_rOwned[ac count]); } }</pre>

		<pre> _isExcluded[account] = true; _excluded.push(account); } </pre>
<p>isExcludedFromFee(address) should be declared external:</p> <p>-</p> <p>SlappyCat.isExcludedFromFee(address) (SlappyCat.sol#675-677)</p>	<p> Severity : High</p>	<pre> function isExcludedFromFee(address account) public view returns(bool) { return _isExcludedFromFees[account]; } </pre>








































Contract Flow Graph




























Inheritance Graph


























































Contract Descriptions




Contract	Type	Bases		
		Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal 		
	_msgData	Internal 		
Ownable	Implementation	Context		
		Public 		NO 
	owner	Public 		NO 
	renounceOwnership	Public 		onlyOwner
	transferOwnership	Public 		onlyOwner
IERC20	Interface			
	totalSupply	External 		NO 
	balanceOf	External 		NO 
	transfer	External 		NO 
	allowance	External 		NO 
	approve	External 		NO 
	transferFrom	External 		NO 
Address	Library			
	isContract	Internal 		
	sendValue	Internal 		
	functionCall	Internal 		
	functionCall	Internal 		
	functionCallWithValue	Internal 		
	functionCallWithValue	Internal 		
	_functionCallWithValue	Private 		

IUniswapV2Factory	Interface			
	feeTo	External !		NO !
	feeToSetter	External !		NO !
	getPair	External !		NO !
	allPairs	External !		NO !
	allPairsLength	External !		NO !
	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 !
IUniswapV2Route r01	Interface			
	factory	External !		NO !
	WETH	External !		NO !
	addLiquidity	External !		NO !
	addLiquidityETH	External !		NO !
	removeLiquidity	External !		NO !
	removeLiquidity ETH	External !		NO !
	removeLiquidity WithPermit	External !		NO !
	removeLiquidity ETHWithPermit	External !		NO !
	swapExactTokens ForTokens	External !		NO !
	swapTokensForEx actTokens	External !		NO !
	swapExactETHFor Tokens	External !		NO !
	swapTokensForEx actETH	External !		NO !
	swapExactTokens ForETH	External !		NO !
	swapETHForExact Tokens	External !		NO !
	quote	External !		NO !
	getAmountOut	External !		NO !
	getAmountIn	External !		NO !
	getAmountsOut	External !		NO !
	getAmountsIn	External !		NO !
IUniswapV2Route r02	Interface	IUniswapV2Route r01		
	removeLiquidity ETHSupportingFe eOnTransferToke ns	External !		NO !

	removeLiquidity ETHWithPermitSu pportingFeeOnTr ansferTokens	External !		NO!
	swapExactTokens ForTokensSupport ingFeeOnTransf erTokens	External !		NO!
	swapExactETHFor TokensSupportin gFeeOnTransferT okens	External !		NO!
	swapExactTokens ForETHSupportin gFeeOnTransferT okens	External !		NO!
SlappyCat	Implementation	Context, IERC20, Ownable		
		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!
	increaseAllowan ce	Public !		NO!
	decreaseAllowan ce	Public !		NO!
	isExcludedFromR eward	Public !		NO!
	totalReflection Distributed	Public !		NO!
	deliver	Public !		NO!
	reflectionFromT oken	Public !		NO!
	tokenFromReflec tion	Public !		NO!

	excludeFromReward	Public !		onlyOwner
	includeInReward	External !		onlyOwner
		External !		NO!
	claimStuckTokens	External !		onlyOwner
	_reflectFee	Private 		
	_getValues	Private 		
	_getTValues	Private 		
	_getRValues	Private 		
	_getRate	Private 		
	_getCurrentSupply	Private 		
	_takeLiquidity	Private 		
	_takeMarketing	Private 		
	calculateTaxFee	Private 		
	calculateLiquidityFee	Private 		
	calculateMarketingFee	Private 		
	removeAllFee	Private 		
	setBuyFee	Private 		
	setSellFee	Private 		
	isExcludedFromFee	Public !		NO!
	_approve	Private 		
	_transfer	Private 		
	buybackAndBurn	Private 		
	setSwapTokensAtAmount	External !		onlyOwner
	setSwapEnabled	External !		onlyOwner
	_tokenTransfer	Private 		
	_transferStandard	Private 		
	_transferToExcluded	Private 		
	_transferFromExcluded	Private 		
	_transferBothExcluded	Private 		
	excludeFromFees	External !		onlyOwner
	changeMarketingWallet	External !		onlyOwner

	setBuyFeePercentages	External !		onlyOwner
	setSellFeePercentages	External !		onlyOwner
	enableWalletToWalletTransferWithoutFee	External !		onlyOwner



Function
can modify
state



Function
is payable

Source:

File Name SHA-1 Hash
c:\Solidity\slappy.sol

e3a0254b6a3fc5cffa8ab771444a891230040601