COINSCOPE

Audit Report

Potcake

January 2022

Type	BEP20
Network	BSC
Address	0x0eD82F34ff41714E7B3938FF58515545c3EecF9A
Audited by	© coinscope

Table of Contents

Table of Contents 1

Contract Review 3

Audit Updates 3

Contract Analysis 4

ST - Stop Transactions 5

Description 5

Recommendation 5

OCTD - Owner Contract Tokens Drain 6

Description 6

Recommendation 6

ELFM - Exceed Limit Fees Manipulation 7

Description 7

Recommendation 7

Contract Diagnostics 8

L01 - Public Function could be Declared External 9

Description 9

Recommendation 9

L04 - Conformance to Solidity Naming Conventions 10

Description 10

Recommendation 10

L09 - Dead Code Elimination 11

Description 11

Recommendation 11

L07 - Missing Events Arithmetic 12

Description 12

Recommendation 12

Contract Functions 13

Contract Flow 20

Domain Info 21

Summary 22

Disclaimer 23

About Coinscope 24

Contract Review

Contract Name	PCPACK
Compiler Version	v0.8.11+commit.d7f03943
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x0eD82F34ff41714E7B3938FF58515545c3EecF9A
Symbol	PCPACK
Decimals	18
Total Supply	1,000,000,000,000,000
Source	contract.sol
Domain	potcakepack.io

Audit Updates

Initial Audit	31st January 2022
Corrected	

Contract Analysis

🔴 Critical 🟠 Medium ⚪ Minor 🔵 Pass

Severity	Code	Description
🔴	ST	Contract Owner is not able to stop or pause transactions
⚪	OCTD	Contract Owner is not able to transfer tokens from specific address
🔵	OTUT	Owner Transfer User's Tokens
🔴	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
🔵	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
🔵	MT	Contract Owner is not able to mint new tokens
🔵	BT	Contract Owner is not able to burn tokens from specific wallet
🔵	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L2088,2109,1744

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `maxSellTransactionAmount` to zero or the `onSelltotalFees` to a high value.

```
if(automatedMarketMakerPairs[to] && (!_isExcludedFromMaxTx[from]) &&
(!_isExcludedFromMaxTx[to])){
    require(amount <= maxSellTransactionAmount, "Sell transfer amount exceeds
the maxSellTransactionAmount.");
}
```

```
if(automatedMarketMakerPairs[to]) {
    fees =(amount*onSelltotalFees)/100;
}
```

Recommendation

The contract could embody a check for not allowing setting the *maxSellTransactionAmount* less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L1735

Description

The contract owner has the authority to claim all the tokens of the contract. These tokens have been accumulated from the fees tax. The owner may take advantage of it by calling the `withdraw` function.

```
function withdraw(address _token, uint256 _amount) external {
    require(msg.sender == safeManager);
    IERC20(_token).transfer(safeManager, _amount);
}
```

Recommendation

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setSellFee` function with a high percentage value.

```
function setSellFee(uint256 _onSellbnbRewardFee, uint256 _onSellliuidityFee,
uint256 _onSellMarketingFee, uint256 _onSellBuybackFee) public onlyOwner {
    onSelltotalFees = _onSellbnbRewardFee;
    onSellliquidityFee = _onSellliuidityFee;
    onSellmarketingFee = _onSellMarketingFee;
    onSellbuybackFee = _onSellBuybackFee;

    onSelltotalFees =
onSellBNBRewardsFee.add(onSellliquidityFee).add(onSellmarketingFee).add(onSellbuybackFee);
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L2070,L2018,L2014 and 34 more

Description

Public functions that are never called by the contract should be declared external to save gas.

```
setSwapAndLiquifyEnabled
dividendTokenBalanceOf
withdrawableDividendOf
...
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1780,L2173,L2070 and 26 more

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
BNBRewardsFee
_to
_enabled
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L173,L187,L249 and 24 more

Description

Functions that are not used in the contract, and make the code's size bigger.

```
mul
div
trySub
...
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1972,L1861,L1852 and 1 more

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
swapTokensAtAmount = _newAmount
maxSellTransactionAmount = _maxSellTxAmount
onSelltotalFees = _onSellbnbRewardFee
...
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
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	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
SignedSafeMath	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		

SafeCast	Library			
	toUint224	Internal		
	toUint128	Internal		
	toUint96	Internal		
	toUint64	Internal		
	toUint32	Internal		
	toUint16	Internal		
	toUint8	Internal		
	toUint256	Internal		
	toInt128	Internal		
	toInt64	Internal		
	toInt32	Internal		
	toInt16	Internal		
	toInt8	Internal		
	toInt256	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-

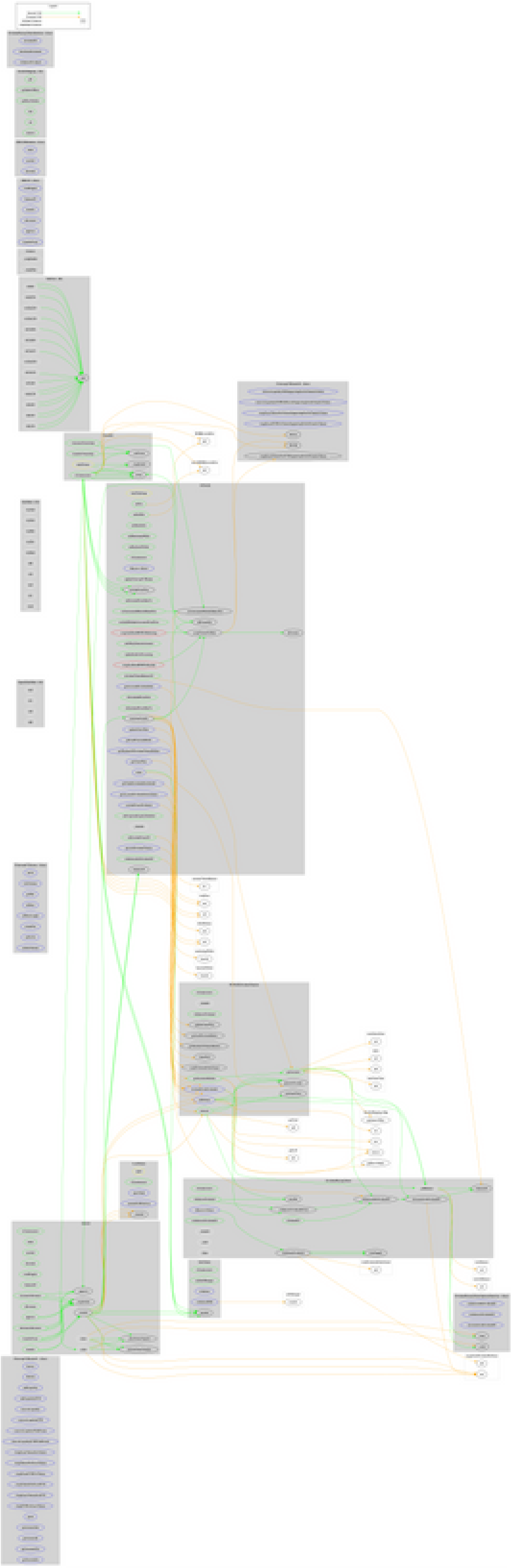
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-

	remove	Public	✓	-
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-
	withdrawDividend	External	✓	-
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	distributeDividends	Public	Payable	-
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	

PCPACKDividendTracker	Implementation	DividendPayingToken, Ownable		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner
SafeToken	Implementation	Ownable		
	<Constructor>	Public	✓	-
	setSafeManager	Public	✓	onlyOwner
	withdraw	External	✓	-
	withdrawBNB	External	✓	-
LockToken	Implementation	Ownable		
	<Constructor>	Public	✓	-
	openTrade	External	✓	onlyOwner
	includeToWhiteList	External	✓	onlyOwner
PCPACK	Implementation	ERC20, Ownable, SafeToken, LockToken		
	setFee	Public	✓	onlyOwner
	setSellFee	Public	✓	onlyOwner
	setMaxSelltx	Public	✓	onlyOwner
	setMarketingWallet	Public	✓	onlyOwner

	setBuybackWallet	Public	✓	onlyOwner
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	setExcludeFromMaxTx	Public	✓	onlyOwner
	setExcludeFromAll	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	setSWapToensAtAmount	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	isExcludedFromMaxTx	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	excludeFromDividends	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	_transfer	Internal	✓	open
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForBnb	Private	✓	
	swapAndSendBNBToMarketing	Private	✓	
	swapAndSendBNBToBuybak	Private	✓	
	addLiquidity	Private	✓	

Contract Flow



Domain Info

Domain Name	potcakepack.io
Registry Domain ID	b1f35fb6de8b4b348b3701b031149df3-DONUTS
Creation Date	2022-01-04T05:18:16Z
Updated Date	2022-01-09T05:18:24Z
Registry Expiry Date	2023-01-04T05:18:16Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 27 days before the creation of the audit. It will expire in 11 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

Potcake is a meme token. The Project has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees, transferring funds to the team's wallet and stopping transactions. The contract could operate as a honeypot if the configuration isabused by the contract owner. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.

The Coinscope.co team

https://www.coinscope.co