Cyberscope

Audit Report

panKUKU

April 2022

Table of Contents

Contract Review

Contract Name	Pankuku
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x84Fd7CC4Cd689fC021e E3D00759B6D255269D538
Symbol	KUKU
Decimals	18
Total Supply	1,000,000,000,000
Domain	pankuku.net

Source Files

Filename	**SHA256**
contract.sol	42fe968352188f31cb9a852d6d4c4e861af0b8b297a8b 7df8a62d7c7d3d76b69

Audit Updates

Initial Audit	2nd May 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	medium
Location	contract.sol#L1062

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `_maxTxAmount` to zero.

```
if(from != owner() && to != owner())
    require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount 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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L955,1132

Description

The contract is accumulating fees for the same addresses by two ways. First, as a proportion of the reflection fees and secondly as a proportion of the liquidity fees.

```
_takefunds(charityAddress,_rtotransferrcharity,_ttotransferrcharity);
...
charityAddress.transfer(convertbaltoone);
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L431,440,446,451,459,771,775,779,783,792,797,801,806,812,817,822,826,830,839,856,889,893,911,1042

Description

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

```
isExcludedFromFee
setSwapAndLiquifyEnabled
includeInFee
excludeFromFee
excludeFromReward
reflectionFromToken
deliver
totalFees
isExcludedFromReward
...
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L708,706,707,702,725,723,722,721,724,720,719,713

Description

Constant state variables should be declared constant to save gas.

```
transferTeamfee
teamDivisor
sellliquiduty
marketingDivisor
farmstakingDivisor
ecosystemDivisor
charityDivisor
burnDivisor
_tTotal
...
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L503,504,521,543,911,1015,1021,1234,1238,1242,1245,1249,1253,712,716,737

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.

```
_maxTxAmount
_liquidityFee
_taxFee
_selltokenswap
_farmsstakingAddress
_teamAddress
_charityAddress
_ecosystemAddress
_marketingAddress
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L897,901,905,1253,1256

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.

```
_maxTxAmount = maxTxAmount
numTokensSellToAddToLiquidity = _selltokenswap
_maxTxAmount = _tTotal.mul(maxTxPercent).div(10 ** 2)
_liquidityFee = liquidityFee
_taxFee = taxFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L357,317,327,342,352,264,291

Description

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

```
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L919,1109

Description

Performing divisions before multiplications may cause lose of prediction.

```
convertbaltoone = newBalance.div(5)
onepartliqudity = contractTokenBalance.div(_liquidityFee)
tonePart = tFee.div(_taxFee)
ronePart = rFee.div(_taxFee)
...
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

	_functionCallWithValue	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	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	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-

	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	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	✓	-
Pankuku	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	_transferBothExcluded	Private	✓	

excludeFromFee	Public	✓	onlyOwner
includeInFee	Public	✓	onlyOwner
setTaxFeePercent	External	✓	onlyOwner
setLiquidityFeePercent	External	✓	onlyOwner
setMaxTxPercent	External	✓	onlyOwner
setSwapAndLiquifyEnabled	Public	✓	onlyOwner
<Receive Ether>	External	Payable	-
_reflectFee	Private	✓	
_takefunds	Private	✓	
_getValues	Private		
_getTValues	Private		
_getRValues	Private		
_getRate	Private		
_getCurrentSupply	Private		
_takeLiquidity	Private	✓	
calculateTaxFee	Private		
calculateLiquidityFee	Private		
removeAllFee	Private	✓	
restoreAllFee	Private	✓	
isExcludedFromFee	Public		-
_approve	Private	✓	
_transfer	Private	✓	
swapAndLiquify	Private	✓	lockTheSwap
swapTokensForEth	Private	✓	
addLiquidity	Private	✓	
_tokenTransfer	Private	✓	
_transferStandard	Private	✓	
_transferToExcluded	Private	✓	
_transferFromExcluded	Private	✓	
setMarketingAddress	External	✓	onlyOwner
setEcosystemAddress	External	✓	onlyOwner
setCharityAddress	External	✓	onlyOwner
setTeamAddress	External	✓	onlyOwner
setFarmsStakingAddress	External	✓	onlyOwner
setNumTokensSellToAddToLiquidity	External	✓	onlyOwner

setMaxTxAmount	External	✓	onlyOwner
setSellFee	Internal	✓	
setTransferFee	Internal	✓	

Contract Flow
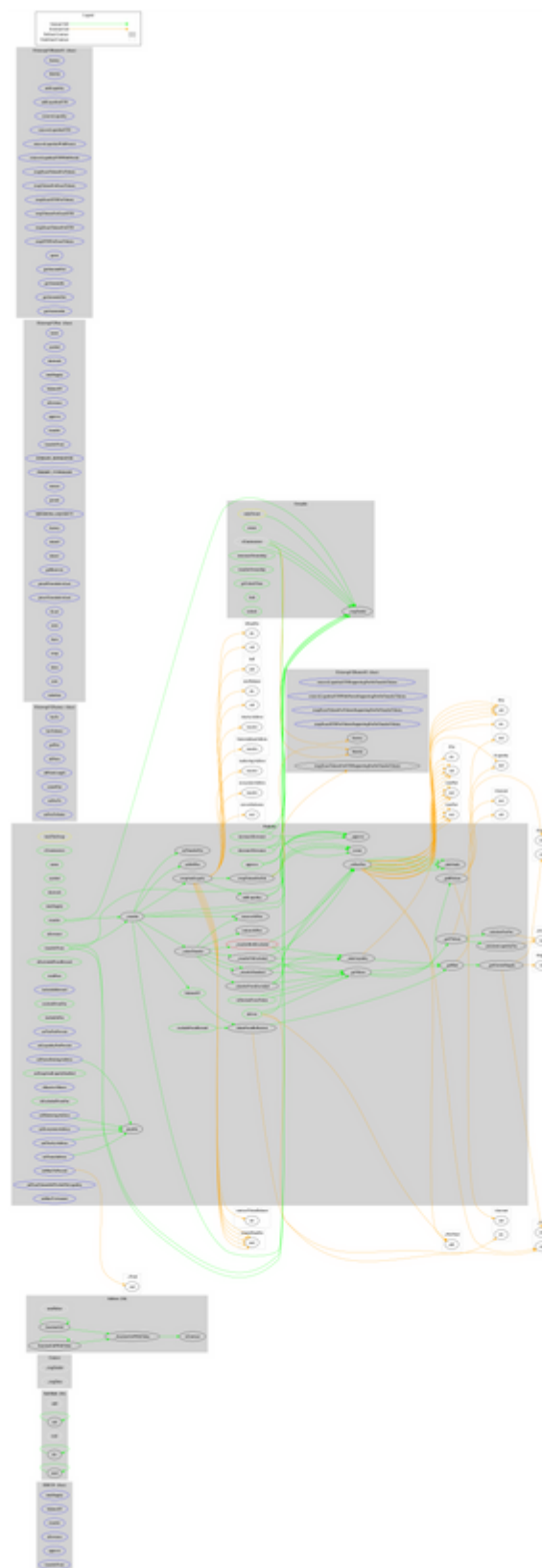
Domain Info

Domain Name	pankuku.net
Registry Domain ID	2673928447_DOMAIN_NET-VRSN
Creation Date	2022-02-09T02:03:48Z
Updated Date	2022-02-09T02:03:48Z
Registry Expiry Date	2024-02-09T02:03:48Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	https://www.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 3 months before the creation of the audit. It will expire in almost 2 years.

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

Summary

There are some functions that can be abused by the owner like stopping transactions and transferring funds to the team's wallet. 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.

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

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

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.

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

The Cyberscope team

https://www.cyberscope.io