Cyberscope

Audit Report

KASA CENTRAL

September 2022

Type	BEP20
Network	BSC
Address	0x8106789b240E5e1b34643c052F1dc1B7a1A451A5
Audited by	© cyberscope

Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
Contract Diagnostics 5
CO - Code Optimization 6
Description 6
Recommendation 6
CR - Code Repetition 7
Description 7
Recommendation 7
L01 - Public Function could be Declared External 8
Description 8
Recommendation 8
L04 - Conformance to Solidity Naming Conventions 9
Description 9
Recommendation 9
L07 - Missing Events Arithmetic 10
Description 10
Recommendation 10
L12 - Using Variables before Declaration 11
Description 11
Recommendation 11
L13 - Divide before Multiply Operation 12
Description 12

Recommendation 12
L14 - Uninitialized Variables in Local Scope 13
Description 13
Recommendation 13
Contract Functions 14
Contract Flow 17
Domain Info 18
Summary 19
Disclaimer 20
About Cyberscope 21

Contract Review

Contract Name	KasaCentral
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	5000 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x8106789b240E5e1b34643c052F1dc1B7a1A451A5
Symbol	KASA
Decimals	18
Total Supply	100,000,000
Domain	https://kasacentral.net

Source Files

Filename	SHA256
contract.sol	8dc95db867a9597990f513a4f48bdd5ef8b3c98c8be6f53871c527fe0d3cfcff

Audit Updates

Initial Audit	17th September 2022 https://github.com/cyberscope-io/audits/blob/main/v1/kasa/audit.pdf
Corrected	19th September 2022

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	CO	Code Optimization	Unresolved
●	CR	Code Repetition	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L549
Status	Unresolved

Description

The success variable is assigned twice but none of them is used from the contract.

```
if (ratios.marketing > 0) {
    (success,) = _taxWallets.marketing.call{value: marketingBalance, gas:
35000}("");
}
if (ratios.development > 0) {
    (success,) = _taxWallets.development.call{value: developmentBalance, gas:
35000}("");
}
```

Recommendation

The contract could remove the success variable assignment.

CR - Code Repetition

Criticality	minor / informative
Location	contract.sol#L614,751
Status	Unresolved

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

```
_rOwned[account] = _tOwned[account] * _getRate();
_tOwned[account] = 0;
_isExcluded[account] = false;
_excluded.pop();
//
uint256 rOwned = _rOwned[_excluded[i]];
uint256 tOwned = _tOwned[_excluded[i]];
if (rOwned > rSupply || tOwned > tSupply) return rTotal / tTotal;
rSupply = rSupply - rOwned;
tSupply = tSupply - tOwned;
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L593,297,378,597,566
Status	Unresolved

Description

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

```
isExcludedFromReward
transfer
getCirculatingSupply
setExcludedFromReward
enableTrading
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L151,150,142,114,33,164,386,115,119,136,149,177,116,117,152
Status	Unresolved

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.

```
maxTransferTaxes
maxSellTaxes
_ratios
startingSupply
WETH
_taxWallets
_antiSnipe
_name
_tTotal
...
```

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 / informative
Location	contract.sol#L424,433
Status	Unresolved

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.

```
swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor
piSwapPercent = priceImpactSwapPercent
```

Recommendation

Emit an event for critical parameter changes.

L12 - Using Variables before Declaration

Criticality	minor / informative
Location	contract.sol#L691
Status	Unresolved

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
check
```

Recommendation

The variables should be declared before any usage of them.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L683
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
feeAmount = (tAmount * currentFee) / masterTaxDivisor
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L691,690,684
Status	Unresolved

Description

The are variables that are defined in the local scope and are not initialized.

```
check
checked
values
```

Recommendation

All the local scoped variables should be initialized.

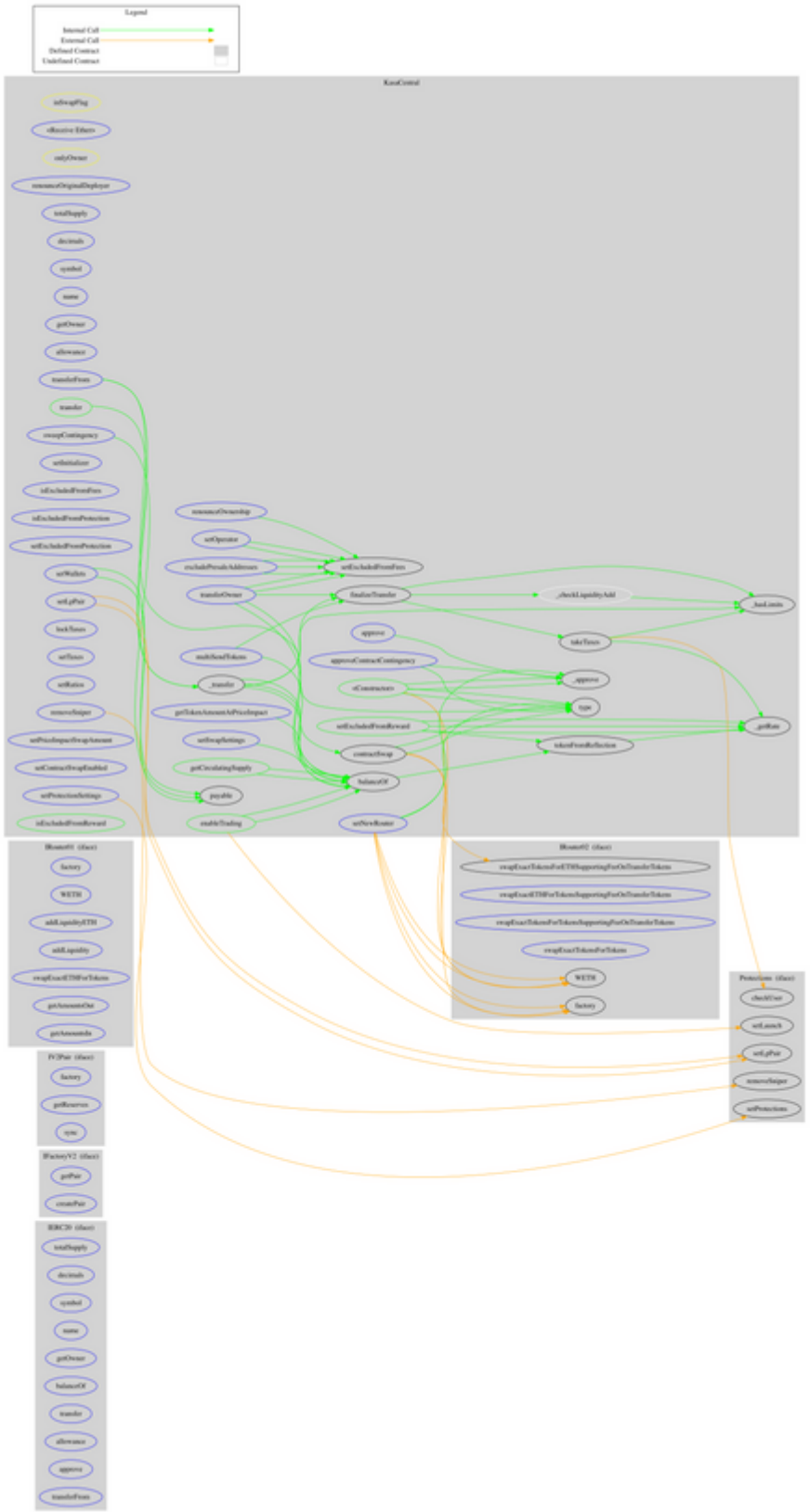
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IFactoryV2	Interface			
	getPair	External		-
	createPair	External	✓	-
IV2Pair	Interface			
	factory	External		-
	getReserves	External		-
	sync	External	✓	-
IRouter01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	addLiquidity	External	✓	-
	swapExactETHForTokens	External	Payable	-

	getAmountsOut	External		-
	getAmountsIn	External		-
IRouter02	Interface	IRouter01		
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokens	External	✓	-
Protections	Interface			
	checkUser	External	✓	-
	setLaunch	External	✓	-
	setLpPair	External	✓	-
	setProtections	External	✓	-
	removeSniper	External	✓	-
KasaCentral	Implementation	IERC20		
	<Constructor>	Public	Payable	-
	<Receive Ether>	External	Payable	-
	transferOwner	External	✓	onlyOwner
	renounceOwnership	External	✓	onlyOwner
	setOperator	External	✓	-
	renounceOriginalDeployer	External	✓	-
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	allowance	External		-
	balanceOf	Public		-
	transfer	Public	✓	-
	approve	External	✓	-
	_approve	Internal	✓	

	approveContractContingency	External	✓	onlyOwner
	transferFrom	External	✓	-
	setNewRouter	External	✓	onlyOwner
	setLpPair	External	✓	onlyOwner
	setInitializer	External	✓	onlyOwner
	isExcludedFromFees	External		-
	isExcludedFromProtection	External		-
	setExcludedFromFees	Public	✓	onlyOwner
	setExcludedFromProtection	External	✓	onlyOwner
	getCirculatingSupply	Public		-
	removeSniper	External	✓	onlyOwner
	setProtectionSettings	External	✓	onlyOwner
	lockTaxes	External	✓	onlyOwner
	setTaxes	External	✓	onlyOwner
	setRatios	External	✓	onlyOwner
	setWallets	External	✓	onlyOwner
	getTokenAmountAtPriceImpact	External		-
	setSwapSettings	External	✓	onlyOwner
	setPriceImpactSwapAmount	External	✓	onlyOwner
	setContractSwapEnabled	External	✓	onlyOwner
	excludePresaleAddresses	External	✓	onlyOwner
	_hasLimits	Internal		
	_transfer	Internal	✓	
	contractSwap	Internal	✓	inSwapFlag
	_checkLiquidityAdd	Internal	✓	
	enableTrading	Public	✓	onlyOwner
	sweepContingency	External	✓	onlyOwner
	multiSendTokens	External	✓	onlyOwner
	isExcludedFromReward	Public		-
	setExcludedFromReward	Public	✓	onlyOwner
	tokenFromReflection	Public		-
	finalizeTransfer	Internal	✓	
	takeTaxes	Internal	✓	
	_getRate	Internal		

Contract Flow



Domain Info

Domain Name	kasacentral.net
Registry Domain ID	9213270
Creation Date	2022-07-09T10:09:10Z
Updated Date	2022-07-09T10:09:13Z
Registry Expiry Date	2023-07-09T10:09:10Z
Registrar WHOIS Server	whois.bluehost.com
Registrar URL	http://www.bluehost.com/
Registrar	FastDomain Inc.
Registrar IANA ID	1154

The domain was created 2 months before the creation of the audit. It will expire in 10 months.

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

Summary

KASA CENTRAL 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 max 9% fees.

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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io