AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Kumamon Coin

DATED : 13 June 23'

AUDIT SUMMARY

Project name – Kumamon Coin

Date: 13 June, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: Passed

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	1
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1- Manual Review:
A line by line code review has been performed by audit ace team.

2- BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3- Slither :
The code has undergone static analysis using Slither.

Testnet version:
The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:
https://testnet.bscscan.com/token/0x5ad8614fD11085b7B698e50829Fe8298696lf8aE

Token Information

Token Name : Kumamon Coin

Token Symbol: KUMA

Decimals: 9

Token Supply: 100,000,000,000,000

Token Address:
0x8144619D941C8CD0BE785c98Ffa91085B310D656

Checksum:
5822cb3645e58834a2ffc5147639e1aba52038f9

Owner:
0x09ad54f7Fb06E90452c7CBe2ED288ab710779f53

Deployer:
0x09ad54f7Fb06E90452c7CBe2ED288ab710779f53

TOKEN OVERVIEW

Fees:

Buy Fees: 0-6%

Sell Fees: 0-6%

Transfer Fees: 0%

Fees Privilege: Owner

Ownership: owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: Yes

Other Privileges: Initial distribution of tokens

excluding from fees

including in fees

changing fees

AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code isexercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

VULNERABILITY CHECKLIST

- Return values of low-level calls
- Private modifier
- Multiple Sends
- Using Suicide
- Gas Limitand Loops
- Address hardcoded
- Exception Disorder
- Using inline assembly
- Divide before multiply
- Missing Zero Address Validation
- Compiler version not fixed
- **Gasless Send**
- Using block.timestamp
- Re-entrancy
- Tautology or contradiction
- Timestamp Dependence
- Revert/require functions
- Use of tx.origin
- Integer overflow/underflow
- Dangerous strict equalities
- Using SHA3
- Using throw

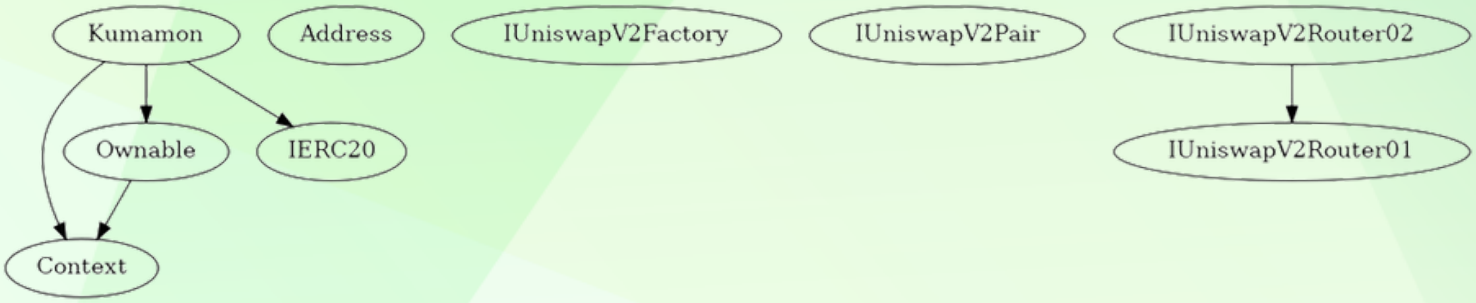
CLASSIFICATION OF RISK

Severity	Description
Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
Gas Optimization /Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
Critical	0
High-Risk	0
Medium-Risk	0
Low-Risk	0
Gas Optimization / Suggestions	1

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to set buy/sell fees higher than 6%
- Owner is not able to set fee on transfer (0% transfer fee)
- Owner is not able to set max buy/sell/transfer/hold amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	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 ❗

CONTRACT ASSESMENT

└	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 ❗	
IUniswapV2Router01	Interface				
└	factory	External ❗			NO ❗	
└	WETH	External ❗			NO ❗	
└	addLiquidity	External ❗	🛑	NO ❗	
└	addLiquidityETH	External ❗	💵	NO ❗	
└	removeLiquidity	External ❗	🛑	NO ❗	
└	removeLiquidityETH	External ❗	🛑	NO ❗	
└	removeLiquidityWithPermit	External ❗	🛑	NO ❗	
└	removeLiquidityETHWithPermit	External ❗	🛑	NO ❗	
└	swapExactTokensForTokens	External ❗	🛑	NO ❗	
└	swapTokensForExactTokens	External ❗	🛑	NO ❗	
└	swapExactETHForTokens	External ❗	💵	NO ❗	
└	swapTokensForExactETH	External ❗	🛑	NO ❗	
└	swapExactTokensForETH	External ❗	🛑	NO ❗	
└	swapETHForExactTokens	External ❗	💵	NO ❗	
└	quote	External ❗			NO ❗	
└	getAmountOut	External ❗			NO ❗	

CONTRACT ASSESMENT

└	getAmountIn	External ❗		NO ❗
└	getAmountsOut	External ❗		NO ❗
└	getAmountsIn	External ❗		NO ❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
Kumamon	Implementation	Context, IERC20, Ownable		
└	<Constructor>	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 ❗
└	increaseAllowance	Public ❗	🛑	NO ❗
└	decreaseAllowance	Public ❗	🛑	NO ❗
└	isExcludedFromReward	Public ❗		NO ❗
└	totalReflectionDistributed	Public ❗		NO ❗
└	deliver	Public ❗	🛑	NO ❗
└	reflectionFromToken	Public ❗		NO ❗
└	tokenFromReflection	Public ❗		NO ❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO ❗
└	claimStuckTokens	External ❗	🛑	NO ❗
└	setStakingAddress	External ❗	🛑	onlyOwner
└	lockToken	Public ❗	🛑	NO ❗
└	unlockToken	Public ❗	🛑	NO ❗
└	updateFeeBuy	Public ❗	🛑	onlyOwner
└	updateFeeSell	Public ❗	🛑	onlyOwner
└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		

CONTRACT ASSESMENT

└	_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 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	
└	swapAndSendMarketing	Private 🔐	🛑	
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	excludeFromFees	External ❗	🛑	onlyOwner
└	isContract	Internal 🔒		

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
Variable Kumamon._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#964) is too similar to Kumamon._transferToExcluded(address,address,uint256).tTransf
erAmount (contracts/Token.sol#947)
Variable Kumamon._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#964) is too similar to Kumamon._getTValues(uint256).tTransferAmount (contracts/Toke
n.sol#694)
Variable Kumamon._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#945) is too similar to Kumamon._transferStandard(address,address,uint256).tTransferAm
ount (contracts/Token.sol#929)
Variable Kumamon._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#945) is too similar to Kumamon._getValues(uint256).tTransferAmount (contracts/Token.s
ol#684)
Variable Kumamon._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#707) is too similar to Kumamon._getValues(uint256).tTransferAmount (contracts
/Token.sol#684)
Variable Kumamon._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#707) is too similar to Kumamon._transferStandard(address,address,uint256).tTr
ansferAmount (contracts/Token.sol#929)
Variable Kumamon._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#983) is too similar to Kumamon._transferFromExcluded(address,address,uint256).tTran
sferAmount (contracts/Token.sol#966)
Variable Kumamon._getValues(uint256).rTransferAmount (contracts/Token.sol#685) is too similar to Kumamon._transferStandard(address,address,uint256).tTransferAmount (contracts/Token.sol
#929)
Variable Kumamon._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#707) is too similar to Kumamon._transferToExcluded(address,address,uint256).t
TransferAmount (contracts/Token.sol#947)
Variable Kumamon._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#945) is too similar to Kumamon._getTValues(uint256).tTransferAmount (contracts/Token.
sol#694)
Variable Kumamon._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#707) is too similar to Kumamon._getTValues(uint256).tTransferAmount (contract
s/Token.sol#694)
Variable Kumamon._getValues(uint256).rTransferAmount (contracts/Token.sol#685) is too similar to Kumamon._getValues(uint256).tTransferAmount (contracts/Token.sol#684)
Variable Kumamon._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#707) is too similar to Kumamon._transferFromExcluded(address,address,uint256)
.tTransferAmount (contracts/Token.sol#966)
Variable Kumamon._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#983) is too similar to Kumamon._getValues(uint256).tTransferAmount (contracts/Token
.sol#684)
Variable Kumamon._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#983) is too similar to Kumamon._transferStandard(address,address,uint256).tTransfer
Amount (contracts/Token.sol#929)
Variable Kumamon._getValues(uint256).rTransferAmount (contracts/Token.sol#685) is too similar to Kumamon._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.s
ol#947)
Variable Kumamon._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#964) is too similar to Kumamon._getValues(uint256).tTransferAmount (contracts/Token
.sol#684)
Variable Kumamon._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#964) is too similar to Kumamon._transferStandard(address,address,uint256).tTransfer
Amount (contracts/Token.sol#929)
Variable Kumamon._getValues(uint256).rTransferAmount (contracts/Token.sol#685) is too similar to Kumamon._getTValues(uint256).tTransferAmount (contracts/Token.sol#694)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

Kumamon.DEAD (contracts/Token.sol#424) should be constant
Kumamon._decimals (contracts/Token.sol#398) should be constant
Kumamon._name (contracts/Token.sol#396) should be constant
Kumamon._symbol (contracts/Token.sol#397) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

Kumamon.DEV (contracts/Token.sol#425) should be immutable
Kumamon._tTotal (contracts/Token.sol#401) should be immutable
Kumamon.mk (contracts/Token.sol#421) should be immutable
Kumamon.mkTwo (contracts/Token.sol#422) should be immutable
Kumamon.totalBuyFees (contracts/Token.sol#418) should be immutable
Kumamon.totalSellFees (contracts/Token.sol#419) should be immutable
Kumamon.uniswapV2Pair (contracts/Token.sol#428) should be immutable
Kumamon.uniswapV2Router (contracts/Token.sol#427) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output

FUNCTIONAL TESTING

1- Adding liquidity (passed):

https://testnet.bscscan.com/tx/0xee589f8beac5d3726e4e36a726430a637bbca811bd98aefb1abd3cab2290e3a7

2- Buying when excluded (0% tax) (passed):

https://testnet.bscscan.com/tx/0x65a646b68c10410fb1870031fd243262b384038c1589926176e968e827475824

3- Selling when excluded (0% tax) (passed):

https://testnet.bscscan.com/tx/0x22a735502357bbabc350df73eccfdfd29709768e6806049c6b16096f85ca07ef

4- Transferring when excluded from fees(0% tax) (passed):

https://testnet.bscscan.com/tx/0x81cda69d9a6c13f6d515892f2749d118822e26eb6860aca974062de2ffb95484

5- Buying when not excluded from fees (0-6% tax) (passed):

https://testnet.bscscan.com/tx/0x462f242576d6bd5e5abf90509d15c8d45df561c4ccf6648c7ab7fcaa79b58c9e

6- Selling when not excluded from fees (0-6% tax) (passed):

https://testnet.bscscan.com/tx/0xf742b9d3d3e408a51a40ab6f628c1e05da938c74378f1ed7c65b26f9bbb1145f

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (0% tax) (passed):

https://testnet.bscscan.com/tx/0xf0654adb9eb3065ad47887ff047f4696618c19d5544e0cb19c1fc29c23c2cead

8- Internal swap (passed):

Marketing wallets received BNB

https://testnet.bscscan.com/tx/0xf742b9d3d3e408a51a40ab6f628c1e05da938c74378f1ed7c65b26f9bbb1145f

8- Auto-liquidity (passed):

https://testnet.bscscan.com/token/0x16cc906ad7d1365d3ebca8ebc4f01c6143f1e3d4?a=0x000000000000000000000000000000000000dead

MANUAL TESTING

Informational - Staking contract

Impact: Unknown
Function: setStakingAddress
Status: Not Resolved
Overview:
In the current implementation of the code, the owner has the ability to designate any address as a valid staking contract.

```
function setStakingAddress(address stakingAddress, bool _value) external onlyOwner {
    require(isStakingAddress[stakingAddress] != _value, "account is already                the value of '_value'");
    require(isContract(stakingAddress), "call to non-contract");

    isStakingAddress[stakingAddress] = _value;
    emit SetStakingAddress(stakingAddress, _value);
```

Staking contracts can interact with the token to lock and unlock tokens. The purpose and usage of these lockTokens and unlockTokens functions are unclear at this point, and the staking contract is not within the scope of this audit.
Recommendation:
Provide further information about the staking contract, as well as the lockTokens and unlockTokens functions, including their intended use and the role of locked tokens within the system.

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

ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.

https://auditace.tech/

https://t.me/Audit_Ace

https://twitter.com/auditace_

https://github.com/Audit-Ace