AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Dorami Inu

DATED : 16 April 23'

AUDIT SUMMARY

Project name – Dorami Inu

Date: 16 April, 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	0
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 done on BSC Test network, each test has its transaction has 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 Link:
https://testnet.bscscan.com/token/0xB70D10A86547bE0e83357D125472098Bc9Adbd26

Token Information

Token Name : Dorami Inu

Token Symbol: DORAMIINU

Decimals: 9

Token Supply: 1,000,000,000

Token Address:
0xb2aEd8b85DC9E0EeBDb6a1DC010aAF87E2D736d9

Checksum:
cf8cacf094bc014eff2fc6a5e7fd6fa2ae0ae3ff

Owner:
0xef2fe9c3461Bc72c7294217C14329cB9884595C1
(at time of writing the audit)

TOKEN OVERVIEW

Fees:

Buy Fees: 8%

Sell Fees: 8%

Transfer Fees: 8%

Fees Privilege: None

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: including and excluding form fee - changing swap threshold

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

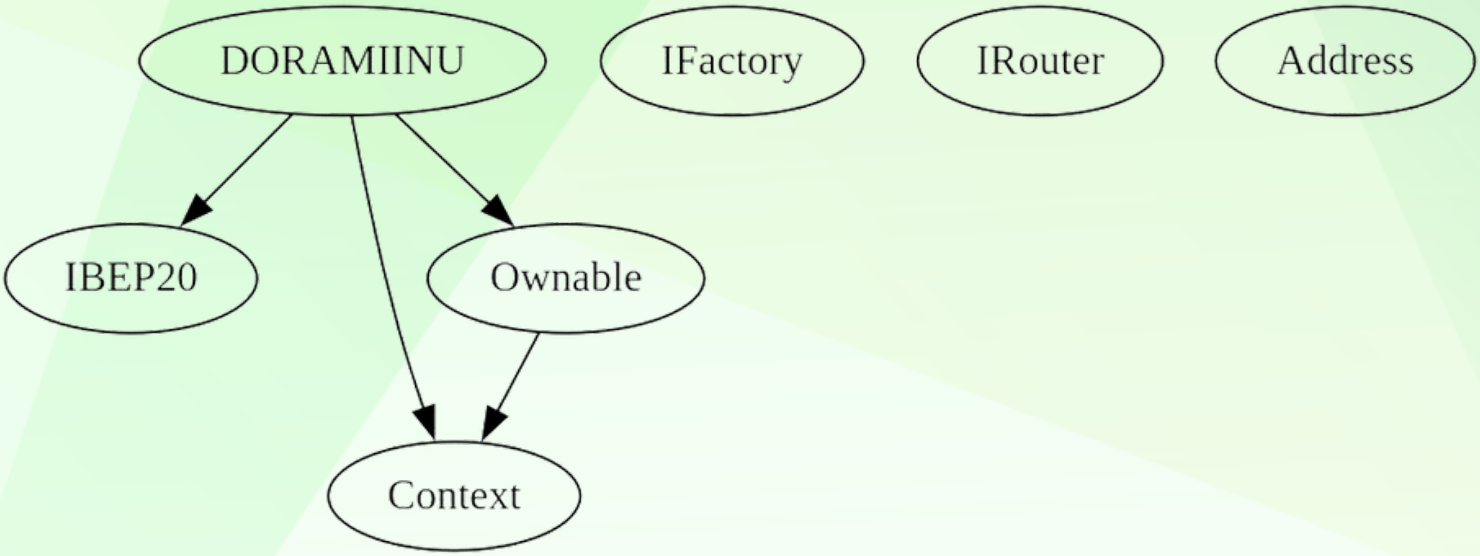
CLASSIFICATION OF RISK

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	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to modify fees (8% buy/sell/transfers)
- Owner must enable trading for investors to be able to trade
- 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**
IBEP20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_setOwner	Private 🔐	🛑	
IFactory	Interface			
└	createPair	External ❗	🛑	NO ❗
IRouter	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
Address	Library			
└	sendValue	Internal 🔒	🛑	
DORAMIINU	Implementation	Context, IBEP20, Ownable		
└	<Constructor>	Public ❗	🛑	NO ❗
└	name	Public ❗		NO ❗
└	symbol	Public ❗		NO ❗
└	decimals	Public ❗		NO ❗
└	totalSupply	Public ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	allowance	Public ❗		NO ❗
└	approve	Public ❗	🛑	NO ❗

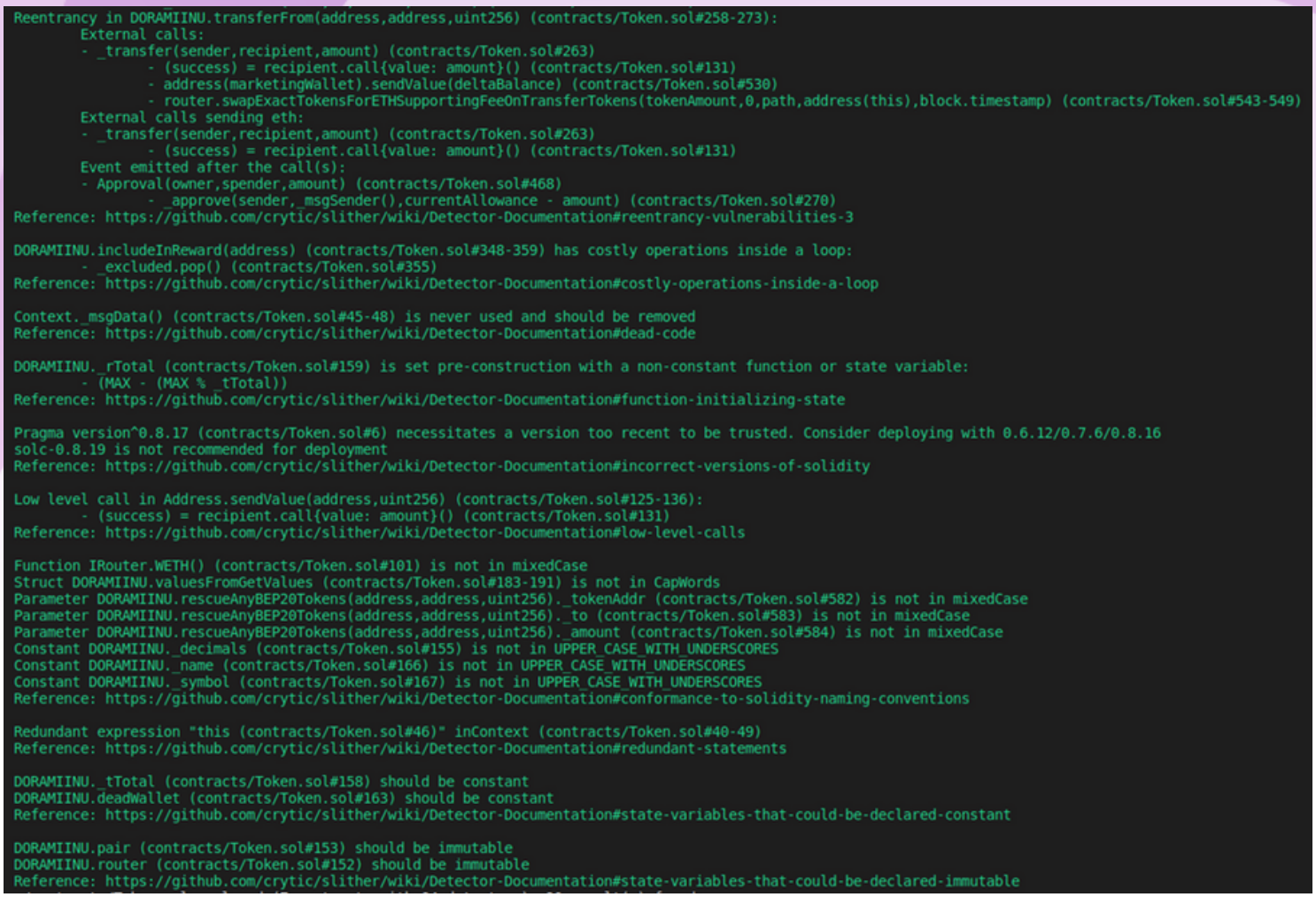
CONTRACT ASSESMENT

└	transferFrom	Public ❗	🛑	NO ❗
└	increaseAllowance	Public ❗	🛑	NO ❗
└	decreaseAllowance	Public ❗	🛑	NO ❗
└	transfer	Public ❗	🛑	NO ❗
└	isExcludedFromReward	Public ❗		NO ❗
└	reflectionFromToken	Public ❗		NO ❗
└	tokenFromReflection	Public ❗		NO ❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	excludeFromFee	Public ❗	🛑	onlyOwner
└	includeInFee	Public ❗	🛑	onlyOwner
└	isExcludedFromFee	Public ❗		NO ❗
└	_reflectRfi	Private 🔐	🛑	
└	_takeMarketing	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap
└	swapTokensForBNB	Private 🔐	🛑	
└	bulkExcludeFee	External ❗	🛑	onlyOwner
└	updateMarketingWallet	External ❗	🛑	onlyOwner
└	updateSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	rescueBNB	External ❗	🛑	onlyOwner
└	rescueAnyBEP20Tokens	Public ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO ❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Reentrancy in DORAMIINU.transferFrom(address,address,uint256) (contracts/Token.sol#258-273):
        External calls:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#263)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#131)
                - address(marketingWallet).sendValue(deltaBalance) (contracts/Token.sol#530)
                - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#543-549)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#263)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#131)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (contracts/Token.sol#468)
                - _approve(sender,_msgSender(),currentAllowance - amount) (contracts/Token.sol#270)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

DORAMIINU.includeInReward(address) (contracts/Token.sol#348-359) has costly operations inside a loop:
        - _excluded.pop() (contracts/Token.sol#355)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#45-48) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

DORAMIINU._rTotal (contracts/Token.sol#159) is set pre-construction with a non-constant function or state variable:
        - (MAX - (MAX % _tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#6) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#125-136):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#131)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#101) is not in mixedCase
Struct DORAMIINU.valuesFromGetValues (contracts/Token.sol#183-191) is not in CapWords
Parameter DORAMIINU.rescueAnyBEP20Tokens(address,address,uint256)._tokenAddr (contracts/Token.sol#582) is not in mixedCase
Parameter DORAMIINU.rescueAnyBEP20Tokens(address,address,uint256)._to (contracts/Token.sol#583) is not in mixedCase
Parameter DORAMIINU.rescueAnyBEP20Tokens(address,address,uint256)._amount (contracts/Token.sol#584) is not in mixedCase
Constant DORAMIINU._decimals (contracts/Token.sol#155) is not in UPPER_CASE_WITH_UNDERSCORES
Constant DORAMIINU._name (contracts/Token.sol#166) is not in UPPER_CASE_WITH_UNDERSCORES
Constant DORAMIINU._symbol (contracts/Token.sol#167) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#46)" inContext (contracts/Token.sol#40-49)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

DORAMIINU._tTotal (contracts/Token.sol#158) should be constant
DORAMIINU.deadWallet (contracts/Token.sol#163) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

DORAMIINU.pair (contracts/Token.sol#153) should be immutable
DORAMIINU.router (contracts/Token.sol#152) 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

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0xb80e1489f5bddc0dd6760df139e39fdf5c59b5f7813ed381a29fdfc383c2e903

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xee196268677676589140351253 96a863644fbefe9ffb6689e45b19eaf969f0b6

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x573a1d73cb8d2c446107082e75 326b5b87e13bb577f4b9b9d174757abe8bf241

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x2232868a6b6bc2e0075c44112c b2c0c1f713f82841fea7997e3ea4c419e494e3

5- Buying when not excluded from fees (8% tax) (passed):
https://testnet.bscscan.com/tx/0x22cead8b31b193ddada3230396 0fef77dd9e7d2f3023c4b53ce96525f204af8a

6- Selling when not excluded from fees (8% tax) (passed):
https://testnet.bscscan.com/tx/0xd889c78c47b1bb44c291bb446d ebdb9ea8f6dba12a1c3ea1a0438486e092507a

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (up to 8% tax) (passed):
https://testnet.bscscan.com/tx/0x85a86fdb639bd76d914e7b4dcf7dc028d2387c8dd468cef466d1de8e34d206e2

8- Internal swap (passed):
Marketing wallet received BNB
https://testnet.bscscan.com/address/0x6869c95cdef9e5598fa5323ff56d521e66a3db50#internaltx

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