



# Smart Contract Audit

FOR

**BAYMAX INU**

DATED : 4 July 23'



# AUDIT SUMMARY

**Project name –** BAYMAX INU

**Date:** 4 July, 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	2
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/0x68898950B4006C883A34f40ad09529E28CEbC5EA>

---



# Token Information

---

**Token Name :** BAYMAX INU

**Token Symbol:** BAYMAX

**Decimals:** 9

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

**Token Address:**

0x6fbaB74C31C28cF42C9AfDE5F9291980F7928129

**Checksum:**

26e7af51f1e3e374238c8472c8e5bba727c01205

**Owner:**

0x0906b9f092EAe7490669274523B7c249fB5F862a  
(at time of writing the audit)

**Deployer:**

0x0906b9f092EAe7490669274523B7c249fB5F862a

---



# TOKEN OVERVIEW

---

## **Fees:**

Buy Fees: 10%

Sell Fees: 10%

Transfer Fees: 10%

---

**Fees Privilege:** Owner

---

**Ownership:** Owned

---

**Minting:** No mint function

---

**Max Tx Amount/ Max Wallet Amount:** No

---

**Blacklist:** No

---

**Other Privileges:** Initial distribution of the tokens

---

---



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

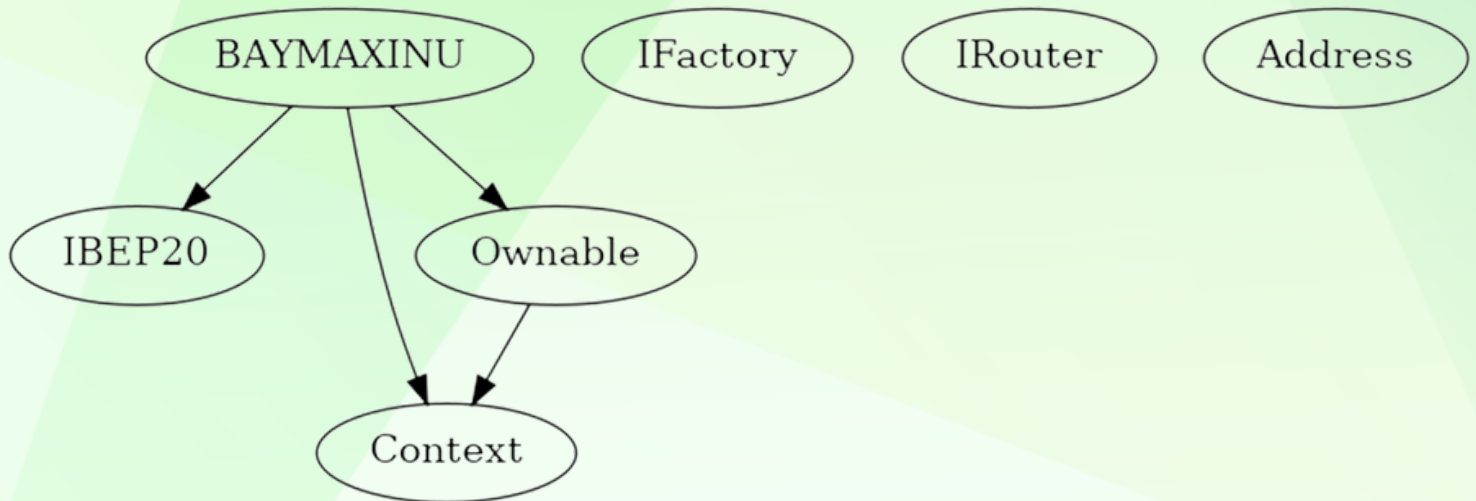
### Found

◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	2



# INHERITANCE TREE

---



# POINTS TO NOTE

---

- owner is not able to change buy/sell/transfer tax (10%)
  - owner is not able to set max buy/sell/transfer/wallet limits
  - owner is not able to blacklist an arbitrary wallet
  - owner is not able to mint new tokens
  - owner is not able to disable trades
-



# CONTRACT ASSESMENT

Contract	Type	Bases			
:-----: :-----: :-----: :-----: :-----:					
L	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
**IBEP20**   Interface					
L	totalSupply	External	!		NO !
L	balanceOf	External	!		NO !
L	transfer	External	!		NO !
L	allowance	External	!		NO !
L	approve	External	!		NO !
L	transferFrom	External	!		NO !
**Context**   Implementation					
L	_msgSender	Internal	!		
L	_msgData	Internal	!		
**Ownable**   Implementation   Context					
L	<Constructor>	Public	!		NO !
L	owner	Public	!		NO !
L	renounceOwnership	Public	!		onlyOwner
L	_setOwner	Private	!		
**IFactory**   Interface					
L	createPair	External	!		NO !
**IRouter**   Interface					
L	factory	External	!		NO !
L	WETH	External	!		NO !
L	addLiquidityETH	External	!		NO !
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External	!		NO !
**Address**   Library					
L	sendValue	Internal	!		
**BAYMAXINU**   Implementation   Context, IBEP20, Ownable					
L	<Constructor>	Public	!		NO !
L	name	Public	!		NO !
L	symbol	Public	!		NO !
L	decimals	Public	!		NO !
L	totalSupply	Public	!		NO !
L	balanceOf	Public	!		NO !
L	allowance	Public	!		NO !
L	approve	Public	!		NO !

# CONTRACT ASSESMENT

<sup>L</sup>	transferFrom	Public !	●	NO !
<sup>L</sup>	increaseAllowance	Public !	●	NO !
<sup>L</sup>	decreaseAllowance	Public !	●	NO !
<sup>L</sup>	transfer	Public !	●	NO !
<sup>L</sup>	isExcludedFromReward	Public !		NO !
<sup>L</sup>	reflectionFromToken	Public !		NO !
<sup>L</sup>	tokenFromReflection	Public !		NO !
<sup>L</sup>	excludeFromReward	Public !	●	onlyOwner
<sup>L</sup>	includeInReward	External !	●	onlyOwner
<sup>L</sup>	excludeFromFee	Public !	●	onlyOwner
<sup>L</sup>	includeInFee	Public !	●	onlyOwner
<sup>L</sup>	isExcludedFromFee	Public !		NO !
<sup>L</sup>	\_reflectRfi	Private 🔒	●	
<sup>L</sup>	\_takeMarketing	Private 🔒	●	
<sup>L</sup>	\_getValues	Private 🔒		
<sup>L</sup>	\_getTValues	Private 🔒		
<sup>L</sup>	\_getRValues	Private 🔒		
<sup>L</sup>	\_getRate	Private 🔒		
<sup>L</sup>	\_getCurrentSupply	Private 🔒		
<sup>L</sup>	\_approve	Private 🔒	●	
<sup>L</sup>	\_transfer	Private 🔒	●	
<sup>L</sup>	\_tokenTransfer	Private 🔒	●	
<sup>L</sup>	swapAndLiquify	Private 🔒	●	lockTheSwap
<sup>L</sup>	swapTokensForBNB	Private 🔒	●	
<sup>L</sup>	bulkExcludeFee	External !	●	onlyOwner
<sup>L</sup>	<Receive Ether>	External !	💰	NO !

## ### Legend

Symbol	Meaning
●	Function can modify state
💰	Function is payable



# STATIC ANALYSIS

```
Reentrancy in BAYMAXINU.transferFrom(address,address,uint256) (contracts/Token.sol#212-220):
  External calls:
    - _transfer(sender,recipient,amount) (contracts/Token.sol#213)
      - (success) = recipient.call{value: amount}{} (contracts/Token.sol#97)
      - address(marketingWallet).sendValue(deltaBalance) (contracts/Token.sol#417)
      - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#430-436)
  External calls sending eth:
    - _transfer(sender,recipient,amount) (contracts/Token.sol#213)
      - (success) = recipient.call{value: amount}{} (contracts/Token.sol#97)
  Event emitted after the call(s):
    - Approval(owner,spender,amount) (contracts/Token.sol#371)
      - _approve(sender,_msgSender(),currentAllowance - amount) (contracts/Token.sol#217)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

BAYMAXINU.includeInReward(address) (contracts/Token.sol#271-282) has costly operations inside a loop:
  - _excluded.pop() (contracts/Token.sol#278)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#31-34) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

BAYMAXINU._rTotal (contracts/Token.sol#122) 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-0.8.20 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#94-99):
  - (success) = recipient.call{value: amount}{} (contracts/Token.sol#97)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#73) is not in mixedCase
Struct BAYMAXINU.valuesFromGetValues (contracts/Token.sol#146-154) is not in CapWords
Constant BAYMAXINU._decimals (contracts/Token.sol#118) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BAYMAXINU._name (contracts/Token.sol#129) is not in UPPER_CASE_WITH_UNDERSCORES
Constant BAYMAXINU._symbol (contracts/Token.sol#130) 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#32)" inContext (contracts/Token.sol#26-35)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

BAYMAXINU._tTotal (contracts/Token.sol#121) should be constant
BAYMAXINU.deadWallet (contracts/Token.sol#126) should be constant
BAYMAXINU.marketingWallet (contracts/Token.sol#127) should be constant
BAYMAXINU.swapTokensAtAmount (contracts/Token.sol#124) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

BAYMAXINU.pair (contracts/Token.sol#116) should be immutable
BAYMAXINU.router (contracts/Token.sol#115) 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/0xce8c8f21efb501a49179b841fb433d9351e5a73dd4abe7018f7f755426ab223f>

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

<https://testnet.bscscan.com/tx/0x4d577bdf92b231cde0e2937bfd47748c7db165d72efdafae2c35660218a32b0b>

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

<https://testnet.bscscan.com/tx/0x0afc7fc5e04900e7c5272298898184a5bb813e0ad0618f161a7cd5f43fc05dd6>

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

<https://testnet.bscscan.com/tx/0x4d82cd53d3bc3d289467fd05666fe6062119b2636f83c88fe4e79ca5a4653a5b>

## 5- Buying when not excluded from fees (10% tax) (passed):

<https://testnet.bscscan.com/tx/0xa2321fdf5b6bd6710317bf9bc50a77611af4b1bd657b3046c6a0b3ad7eedf39>

## 6- Selling when not excluded from fees (10% tax) (passed):

<https://testnet.bscscan.com/tx/0xadc9d203fe6acde4c43e75fb9bd6b8c05a026796640bd8ba2c56f278fe136058>

---



# FUNCTIONAL TESTING

---

## **7- Transferring when not excluded from fees (10% tax) (passed):**

<https://testnet.bscscan.com/tx/0x85b13a99b978a09ee4d9dd36d469079f6670c9c5dc8e0dce524be2af4e0094de>

## **8- Internal swap (passed):**

- BNB fee sent to marketing wallet

- Rewards distributed

<https://testnet.bscscan.com/tx/0xadc9d203fe6acde4c43e75fb9bd6b8c05a026796640bd8ba2c56f278fe136058>

---



# MANUAL TESTING

---

## Missing Logic – Stuck ETH and Tokens

Severity: **Informational**

Status: Open

### Overview:

Contract has no function to withdraw stuck ETH or ERC20 tokens. If ETH or ERC20 sent to contract by mistake, there wont be any ways to withdraw those funds.

### Suggestion

Implement a function to be able to withdraw stuck ETH and ERC20 tokens from the contract (by owner)



# MANUAL TESTING

---

## Missing Logic – Immutable tax

Severity: **Informational**

Status: Open

### Overview:

Fees are immutable and owner is not able to change fees later.

### Suggestion

Its suggested to implement a function for updating fees depending on different market conditions in a safe range.

$0 \leq \text{total buy fees} \leq 10$

$0 \leq \text{total sell fees} \leq 10$

$0 \leq \text{total transfer fees} \leq 10$

<https://docs.pinksale.finance/important/safu-contract>

---



# 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 specialize 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://t.me/Audit_Ace)**



**[https://twitter.com/auditace\\_](https://twitter.com/auditace_)**



**<https://github.com/Audit-Ace>**

---