
www.EtherAuthority.io
audit@etherauthority.io

# SMART CONTRACT

## Security Audit Report

| | |
|---|---|
| Customer: | Bobo Doge Coin |
| Website: | bobodogecoin.com |
| Platform: | Binance Smart Chain |
| Language: | Solidity |
| Date: | August 9th, 2021 |

# Table of contents




**Email: audit@EtherAuthority.io**

THIS IS SECURITY AUDIT REPORT DOCUMENT AND WHICH MAY CONTAIN INFORMATION WHICH IS CONFIDENTIAL. WHICH INCLUDES ANY POTENTIAL VULNERABILITIES AND MALICIOUS CODES WHICH CAN BE USED TO EXPLOIT THE SOFTWARE. THIS MUST BE REFERRED INTERNALLY AND ONLY SHOULD BE MADE AVAILABLE TO PUBLIC AFTER ISSUES ARE RESOLVED.


**Email: audit@EtherAuthority.io**

# Introduction

EtherAuthority was contracted by the Bobo Doge Coin Token team to perform the Security audit of the Bobo Doge Coin Token  smart contract code. The audit has been performed using manual analysis as well as using automated software tools. This report presents all the findings regarding the audit performed on August 9th, 2021.

**The purpose of this audit was to address the following:**

- Ensure that all claimed functions exist and function correctly.
- Identify any security vulnerabilities that may be present in the smart contract.

# Project Background

Bobo Doge Coin  is a BEP20 standard token smart contract based on Binance Smart Chain (BSC). It has additional features like deflection, liquidity, swapping, burning tokens, etc. BoboDogeCoin is the backbone of its ecosystem.

# Audit scope

| Name | Code Review and Security Analysis Report for  Bobo Doge Coin Token Smart Contract |
|---|---|
| **Platform** | **BSC / Solidity** |
| **File** | BoboDogeCoin.sol |
| **Smart Contract Online Code** | https://bscscan.com/address/0xc5d0187f762a346903240e730919ee549d5145fe#code |
| **File MD5 Hash** | 83F568C086EB1006F03E39485B12E9A8 |
| **Audit Date** | August 9th, 2021 |
| **Revision Date** | August 12th, 2021 |

## Claimed Smart Contract Features

| Claimed Feature Detail | Our Observation |
|---|---|
| Name: Bobo Doge Coin | **YES, This is valid.** |
| Symbol: BoboDoge | **YES, This is valid.** |
| Decimals: 9 | **YES, This is valid.** |
| Token Supply: 5200 Trillion | **YES, This is valid.** |
| No more new tokens generated. | **YES, This is valid.** |
| • 2% tax is collected and distributed to holders for HODLing<br>• 9% buyback and marketing tax is collected and 2% of it is sent for marketing fund and other 7% is used to buyback the tokens<br>• 5% add liquidity | **YES, This is valid. Owner can change this fee percentage anytime. So, It is very important to keep the owner's wallet private key secured.** |

# Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **"Secured"**. These contracts also have owner functions (described in the centralization section below), which does not make everything 100% decentralized. Thus, the owner must execute those smart contract functions as per the business plan.


We used various tools like MythX, Slither and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.
All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in AS-IS section and all identified issues can be found in the Audit overview section.

**We found 0 critical, 0 high, 0 medium and 5 low and some very low level issues.**
**These issues are acknowledged while revisions.**

**Investors Advice:** Technical audit of the smart contract does not guarantee the ethical nature of the project. Any owner controlled functions should be executed by the owner with responsibility. All investors/users are advised to do their due diligence before investing in the project.

# Technical Quick Stats

| Main Category | Subcategory | Result |
|---|---|---|
| Contract Programming | Solidity version not specified | Passed |
| | Solidity version too old | Moderated |
| | Integer overflow/underflow | Passed |
| | Function input parameters lack of check | Passed |
| | Function input parameters check bypass | Passed |
| | Function access control lacks management | Passed |
| | Critical operation lacks event log | Passed |
| | Human/contract checks bypass | Passed |
| | Random number generation/use vulnerability | Passed |
| | Fallback function misuse | Passed |
| | Race condition | Passed |
| | Logical vulnerability | Passed |
| | Features claimed | Passed |
| | Other programming issues | Moderated |
| Code Specification | Function visibility not explicitly declared | Passed |
| | Var. storage location not explicitly declared | Passed |
| | Use keywords/functions to be deprecated | Passed |
| | Other code specification issues | Passed |
| Gas Optimization | “Out of Gas” Issue | Passed |
| | High consumption ‘for/while’ loop | Moderated |
| | High consumption ‘storage’ storage | Passed |
| | Assert() misuse | Passed |
| Business Risk | The maximum limit for mintage not set | Passed |
| | “Short Address” Attack | Passed |
| | “Double Spend” Attack | Passed |

**Overall Audit Result:** **PASSED**

## Code Quality

This audit scope has 1 smart contract. This smart contract also contains Libraries, Smart contracts inherits and Interfaces. These are compact and well written contracts.

The libraries in Bobo Doge Coin are part of its logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned a specific address and its properties / methods can be reused many times by other contracts in the Bobo Doge Coin Token token.

The Bobo Doge Coin Token team has **not** provided scenario and unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Some code parts are **not well** commented on smart contracts.

## Documentation

We were given a Bobo Doge Coin smart contracts code in the form of a BscScn web link. The hash of that code is mentioned above in the table.

As mentioned above, some code parts are **not well** commented. So it is difficult to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Another source of information was its official website https://www.bobodogecoin.com/ which provided rich information about the project architecture and tokenomics.

## Use of Dependencies

As per our observation, the libraries are used in this smart contract infrastructure that are based on well known industry standard open source projects. And their core code blocks are written well.

Apart from libraries, its functions are used in external smart contract calls.

# AS-IS overview

## BoboDogeCoin.sol

### (1) Interface

(a) IERC20

(b) IUniswapV2Factory

(c) IUniswapV2Pair

(d) IUniswapV2Router01

(e) IUniswapV2Router02

### (2) Inherited contracts

(a) Context

(b) Ownable

### (3) Usages

(a) using SafeMath for uint256;

(b) using Address for address;

### (4) Events

(a) event RewardLiquidityProviders(uint256 tokenAmount);

(b) event BuyBackEnabledUpdated(bool enabled);

(c) event SwapAndLiquifyEnabledUpdated(bool enabled);

(d) event SwapAndLiquify(uint256 tokensSwapped, uint256 ethReceived, uint256 tokensIntoLiqudity);

(e) event SwapETHForTokens(uint256 amountIn,address[] path);

(f) event SwapTokensForETH(uint256 amountIn,address[] path);

### (5) Functions

| Sl. | Functions | Type | Observation | Conclusion |
|---|---|---|---|---|
| 1 | lockTheSwap | modifier | Passed | No Issue |
| 2 | name | read | Passed | No Issue |
| 3 | decimals | read | Passed | No Issue |
| 4 | symbol | read | Passed | No Issue |
| 5 | totalSupply | read | Passed | No Issue |
| 6 | balanceOf | read | Passed | No Issue |
| 7 | transfer | write | Passed | No Issue |

| | | | | |
|---|---|---|---|---|
| **8** | allowance | read | Passed | No Issue |
| **9** | approve | write | Passed | No Issue |
| **10** | transferFrom | read | Passed | No Issue |
| **11** | increaseAllowance | write | Passed | No Issue |
| **12** | decreaseAllowance | write | Passed | No Issue |
| **13** | isExcludedFromReward | read | Passed | No Issue |
| **14** | totalFees | read | Passed | No Issue |
| **15** | minimumTokensBeforeSwapAmount | read | Passed | No Issue |
| **16** | buyBackUpperLimitAmount | read | Passed | No Issue |
| **17** | deliver | write | Passed | No Issue |
| **18** | reflectionFromToken | read | Passed | No Issue |
| **19** | tokenFromReflection | read | Passed | No Issue |
| **20** | excludeFromReward | write | access only Owner | No Issue |
| **21** | includeInReward | external | Possibility of High gas consumption | Refer audit finding section |
| **22** | _approve | write | Passed | No Issue |
| **23** | _transfer | write | Passed | No Issue |
| **24** | callBuyBack | external | access only Owner | No Issue |
| **25** | swapTokens | write | Passed | No Issue |
| **26** | buyBackTokens | write | Passed | No Issue |
| **27** | swapTokensForEth | write | Passed | No Issue |
| **28** | swapETHForTokens | write | Passed | No Issue |
| **29** | addLiquidity | write | Centralized risk possibility | Refer audit finding section |
| **30** | _tokenTransfer | write | Passed | No Issue |
| **31** | _transferStandard | write | Passed | No Issue |
| **32** | _transferToExcluded | write | Passed | No Issue |
| **33** | _transferFromExcluded | write | Passed | No Issue |
| **34** | _transferBothExcluded | write | Passed | No Issue |
| **35** | _reflectFee | write | Passed | No Issue |
| **36** | _getValues | write | Passed | No Issue |
| **37** | _getTValues | read | Passed | No Issue |
| **38** | _getRValues | write | Passed | No Issue |
| **39** | _getRate | read | Passed | No Issue |
| **40** | _getCurrentSupply | read | Possibility of High gas consumption | Refer audit finding section |
| **41** | _takeLiquidity | write | Passed | No Issue |
| **42** | calculateTaxFee | read | Passed | No Issue |
| **43** | calculateLiquidityFee | read | Passed | No Issue |
| **44** | removeAllFee | write | Passed | No Issue |
| **45** | updateWhiteListFee | write | Passed | No Issue |
| **46** | restoreAllFee | write | Passed | No Issue |
| **47** | updateFeeOfWhiteList | external | access only Owner | No Issue |
| **48** | isExcludedFromFee | write | Passed | No Issue |
| **49** | excludeFromFee | write | access only Owner | No Issue |
| **50** | includeInFee | write | access only Owner | No Issue |
| **51** | setTaxFeePercent | external | access only Owner | No Issue |

| | | | | |
|---|---|---|---|---|
| **52** | setLiquidityFeePercent | external | access only Owner | No Issue |
| **53** | setMaxTxAmount | external | access only Owner | No Issue |
| **54** | setNumTokensSellToAddTo Liquidity | external | access only Owner | No Issue |
| **55** | setBuybackUpperLimit | external | access only Owner | No Issue |
| **56** | setMarketingAddress | external | access only Owner | No Issue |
| **57** | setBuyBackAddress | external | access only Owner | No Issue |
| **58** | setSwapAndLiquifyEnabled | write | access only Owner | No Issue |
| **59** | setBuyBackEnabled | write | access only Owner | No Issue |
| **60** | addWhiteList | external | access only Owner | No Issue |
| **61** | removeWhiteList | external | access only Owner | No Issue |
| **62** | prepareForPreSale | external | access only Owner | No Issue |
| **63** | afterPreSale | external | access only Owner | No Issue |
| **64** | transferToAddressETH | write | Passed | No Issue |
| **65** | receive | external | Passed | No Issue |
| **66** | _msgData | internal | Passed | No Issue |
| **67** | _msgSender | internal | Passed | No Issue |
| **68** | owner | read | Passed | No Issue |
| **69** | onlyOwner | modifier | Passed | No Issue |
| **70** | renounceOwnership | write | access only Owner | No Issue |
| **71** | transferOwnership | write | access only Owner | No Issue |
| **72** | getUnlockTime | read | Passed | No Issue |
| **73** | getTime | read | Passed | No Issue |
| **74** | lock | write | access only Owner | No Issue |
| **75** | unlock | write | Possibility of Regain ownership | Refer audit finding section |

# Severity Definitions

| Risk Level | Description |
|---|---|
| Critical | Critical vulnerabilities are usually straightforward to exploit and can lead to token loss etc. |
| High | High-level vulnerabilities are difficult to exploit; however, they also have significant impact on smart contract execution, e.g. public access to crucial |
| Medium | Medium-level vulnerabilities are important to fix; however, they can't lead to tokens lose |
| Low | Low-level vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution |
| Lowest / Code Style / Best Practice | Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can be ignored. |

# Audit Findings

## Critical

No Critical severity vulnerabilities were found.

## High

No High severity vulnerabilities were found.

## Medium

No Medium severity vulnerabilities were found.

## Low

(1) Possible high gas consuming loop

```
function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

If there are so many excluded wallets, then this logic will fail, as it might hit the block's gas limit. If there are very limited exceptions, then this will work, but will cost more gas.

**Resolution**: We suggest excluding limited wallets only.

**Status:** **acknowledged**

(2) Possibility to regain the ownership:

Owner can renounce ownership and make contract without owner but he can regain ownership by following the steps below:

1. Owner calls the lock function in contract to set the current owner as _previousOwner.
2. Owner calls unlock to unlock the contract and set _owner = _previousOwner.
3. Owner called renounceOwnership to leave the contract without the owner.
4. Owner calls unlock to regain ownership.

**Resolution**: We suggest removing these lock/unlock functions as this seems not serving a great purpose. Otherwise, always renounce ownership first before calling the lock function.

**Status:** **acknowledged**

(3) Make variables constant

```
string private _name = "Bobo Doge Coin";
string private _symbol = "BoboDoge";
uint8 private _decimals = 9;
```

name, symbol, decimals. These variables will be unchanged. So, please make it constant. It will save some gas.

**Resolution**: Declare those variables as constant. Just put a constant keyword.

**Status:** acknowledged

(4) Centralized risk in addLiquidity

```
function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {
    // approve token transfer to cover all possible scenarios
    _approve(address(this), address(uniswapV2Router), tokenAmount);

    // add the liquidity
    uniswapV2Router.addLiquidityETH{value: ethAmount}(
        address(this),
        tokenAmount,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        owner(),
        block.timestamp + 360
    );
}
```

In addLiquidity function, the owner gets funds from the Pool. If the private key of the owner's wallet is compromised, then it will create a problem.

**Resolution**: Ideally this can be a governance smart contract. On another hand, the owner can accept this risk and handle the private key very securely.

**Status:** acknowledged

(5) Missing Events: every function which is state changing, must emit an event. There are missing Events log for some functions like:

- excludeFromFee
- includeInFee
- setTaxFeePercent
- setLiquidityFeePercent
- setMaxTxAmount
- setNumTokensSellToAddToLiquidity
- setBuybackUpperLimit
- setMarketingAddress
- setBuyBackAddress
- setSwapAndLiquifyEnabled
- setBuyBackEnabled
- addWhiteList
- removeWhiteList
- deliver.

## Very Low / Discussion / Best practices:

(1) Unused function

```
function transferToAddressETH(address payable recipient, uint256 amount) private {
    recipient.transfer(amount);
}
```

The function transferToAddressETH is not used anywhere. Although this does not create any vulnerability, it's better to remove it to make the code clean.

**Resolution**: Please remove this function.

**Status:** **acknowledged** .

(2) External instead of public:

If any function is not called from inside the smart contract, then it is better to declare it as external instead of public. As it saves some gas as well.

https://ethereum.stackexchange.com/questions/19380/external-vs-public-best-practices

**Status:** **acknowledged** .

## Centralization

These smart contracts have some functions which can be executed by Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble. Following are Admin functions:

- excludeFromReward: The Owner can check that the account is already excluded from reward..
- includeInReward: The Owner can check that the account is already excluded from reward.
- callBuyBack: The Owner can call buyBackTokens function.
- updateFeeOfWhiteList: The Owner can update the fee of the whitelist.
- excludeFromFee: The Owner can check exclude Fee.
- includeInFee: The Owner can include a fee account bool.
- setTaxFeePercent: The Owner can set a tax Fee.
- setLiquidityFeePercent: The Owner can set liquidity Fee.
- setMaxTxAmount: The Owner can set a max tax amount.
- setNumTokensSellToAddToLiquidity: The Owner can set minimum Tokens Before Swap.
- setBuybackUpperLimit: The Owner can set buy Back Limit.
- setMarketingAddress: The Owner can set a marketing address.
- setBuyBackAddress: The Owner can set a buy back address.
- setSwapAndLiquifyEnabled: The Owner can set swap and liquify enable status.
- setBuyBackEnabled: The Owner can set buy back enable status.
- addWhiteList: The Owner can add wallet address in whitelist.
- removeWhiteList: The Owner can remove wallet address from whitelist.
- prepareForPreSale: The Owner can set SwapAndLiquifyEnabled function , tax fee, _liquidityAndBuyBackFee , maxTxAmoun prepare for presale.
- afterPreSale: The Owner can set SwapAndLiquifyEnabled function , tax fee, _liquidityAndBuyBackFee , maxTxAmount after presale.

# Conclusion

We were given a contract code. And we have used all possible tests based on given objects as files. We observed some issues in the smart contracts and those issues are acknowledged in revisions. So, **it's good to go to production**.

Since possible test cases can be unlimited for such smart contracts protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. Smart Contract's high level description of functionality was presented in As-is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.

Security state of the reviewed contract, based on standard audit procedure scope, is **"Secured".**

# Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

**Manual Code Review:**

In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

**Vulnerability Analysis:**

Our audit techniques included manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's web site to get a high level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. While we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation.

**Documenting Results:**

We follow a conservative, transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even though we have not yet verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities. We generally follow a process of first documenting the suspicion with unresolved questions, then confirming the issue through code analysis, live experimentation, or automated tests. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this we analyze the feasibility of an attack in a live system.

**Suggested Solutions:**

We search for immediate mitigations that live deployments can take, and finally we suggest the requirements for remediation engineering for future releases. The mitigation and remediation recommendations should be scrutinized by the developers and deployment engineers, and successful mitigation and remediation is an ongoing collaborative process after we deliver our report, and before the details are made public.

# Disclaimers

## EtherAuthority.io Disclaimer

EtherAuthority team has analyzed this smart contract in accordance with the best industry practices at the date of this report, in relation to: cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment and functionality (performing the intended functions).

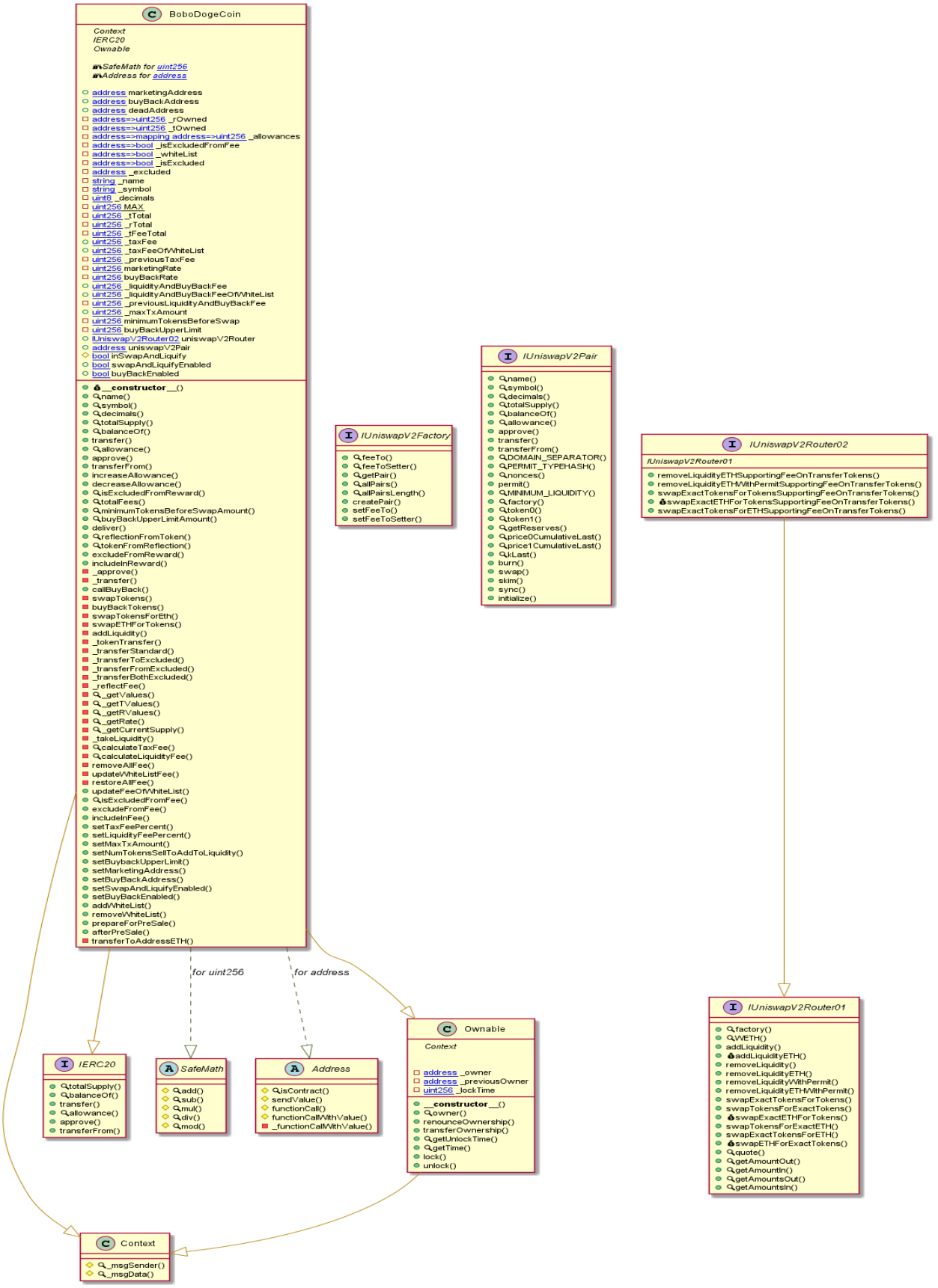
Due to the fact that the total number of test cases are unlimited, the audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only. We also suggest conducting a bug bounty program to confirm the high level of security of this smart contract.

## Technical Disclaimer

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have their own vulnerabilities that can lead to hacks. Thus, the audit can’t guarantee explicit security of the audited smart contracts.

# Appendix

## Code Flow Diagram - Bobo Doge Coin



# Slither Results Log

## Slither log >> BoboDogeCoin.sol

```
INFO:Detectors:
Reentrancy in BoboDogeCoin._transfer(address,address,uint256) (BoboDogeCoin.sol#614-653):
        External calls:
        - swapTokens(contractTokenBalance) (BoboDogeCoin.sol#632)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        External calls sending eth:
        - swapTokens(contractTokenBalance) (BoboDogeCoin.sol#632)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
        State variables written after the call(s):
        - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
                - _liquidityAndBuyBackFee = _previousLiquidityAndBuyBackFee (BoboDogeCoin.sol#892)
                - _liquidityAndBuyBackFee = 0 (BoboDogeCoin.sol#877)
                - _liquidityAndBuyBackFee = _liquidityAndBuyBackFeeOfWhiteList (BoboDogeCoin.sol#887)
        - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
                - _rOwned[address(this)] = _rOwned[address(this)].add(rLiquidity) (BoboDogeCoin.sol#853)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (BoboDogeCoin.sol#778)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (BoboDogeCoin.sol#769)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (BoboDogeCoin.sol#799)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (BoboDogeCoin.sol#789)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (BoboDogeCoin.sol#770)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (BoboDogeCoin.sol#780)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (BoboDogeCoin.sol#790)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (BoboDogeCoin.sol#801)
        - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
                - _rTotal = _rTotal.sub(rFee) (BoboDogeCoin.sol#808)
        - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
                - _tOwned[address(this)] = _tOwned[address(this)].add(tLiquidity) (BoboDogeCoin.sol#855)
                - _tOwned[sender] = _tOwned[sender].sub(tAmount) (BoboDogeCoin.sol#788)
                - _tOwned[sender] = _tOwned[sender].sub(tAmount) (BoboDogeCoin.sol#798)
                - _tOwned[recipient] = _tOwned[recipient].add(tTransferAmount) (BoboDogeCoin.sol#779)
                - _tOwned[recipient] = _tOwned[recipient].add(tTransferAmount) (BoboDogeCoin.sol#800)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities
INFO:Detectors:
BoboDogeCoin.addLiquidity(uint256,uint256) (BoboDogeCoin.sol#731-744) ignores return value by uniswapV2Router.addLiquidityETH{value: ethA
mount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDogeCoin.sol#736-743)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
BoboDogeCoin.allowance(address,address).owner (BoboDogeCoin.sol#516) shadows:
        - Ownable.owner() (BoboDogeCoin.sol#155-157) (function)
BoboDogeCoin._approve(address,address,uint256).owner (BoboDogeCoin.sol#606) shadows:
        - Ownable.owner() (BoboDogeCoin.sol#155-157) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
BoboDogeCoin.setMarketingAddress(address)._marketingAddress (BoboDogeCoin.sol#932) lacks a zero-check on :
                - marketingAddress = address(_marketingAddress) (BoboDogeCoin.sol#933)
BoboDogeCoin.setBuyBackAddress(address)._buyBackAddress (BoboDogeCoin.sol#936) lacks a zero-check on :
                - buyBackAddress = address(_buyBackAddress) (BoboDogeCoin.sol#937)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in BoboDogeCoin._transfer(address,address,uint256) (BoboDogeCoin.sol#614-653):
        External calls:
        - swapTokens(contractTokenBalance) (BoboDogeCoin.sol#632)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        External calls sending eth:
        - swapTokens(contractTokenBalance) (BoboDogeCoin.sol#632)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
        State variables written after the call(s):
        - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
                - _previousLiquidityAndBuyBackFee = _liquidityAndBuyBackFee (BoboDogeCoin.sol#874)
                - _previousLiquidityAndBuyBackFee = _liquidityAndBuyBackFee (BoboDogeCoin.sol#884)
        - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
                - _previousTaxFee = _taxFee (BoboDogeCoin.sol#873)
                - _previousTaxFee = _taxFee (BoboDogeCoin.sol#883)
        - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
```

```
                - _previousTaxFee = _taxFee (BoboDogeCoin.sol#883)
        - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
                - _tFeeTotal = _tFeeTotal.add(tFee) (BoboDogeCoin.sol#809)
        - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
                - _taxFee = _previousTaxFee (BoboDogeCoin.sol#891)
                - _taxFee = 0 (BoboDogeCoin.sol#876)
                - _taxFee = _taxFeeOfWhiteList (BoboDogeCoin.sol#886)
Reentrancy in BoboDogeCoin.constructor() (BoboDogeCoin.sol#476-488):
        External calls:
        - uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this),_uniswapV2Router.WETH()) (BoboDogeCoin.s
ol#480)
        State variables written after the call(s):
        - _isExcludedFromFee[owner()] = true (BoboDogeCoin.sol#484)
        - _isExcludedFromFee[address(this)] = true (BoboDogeCoin.sol#485)
        - uniswapV2Router = _uniswapV2Router (BoboDogeCoin.sol#482)
Reentrancy in BoboDogeCoin.swapTokens(uint256) (BoboDogeCoin.sol#659-685):
        External calls:
        - swapTokensForEth(marketingAmount,marketingAddress) (BoboDogeCoin.sol#669)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(buyBackAmount,buyBackAddress) (BoboDogeCoin.sol#673)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        State variables written after the call(s):
        - swapTokensForEth(buyBackAmount,buyBackAddress) (BoboDogeCoin.sol#673)
                - _allowances[owner][spender] = amount (BoboDogeCoin.sol#610)
Reentrancy in BoboDogeCoin.swapTokens(uint256) (BoboDogeCoin.sol#659-685):
        External calls:
        - swapTokensForEth(marketingAmount,marketingAddress) (BoboDogeCoin.sol#669)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(buyBackAmount,buyBackAddress) (BoboDogeCoin.sol#673)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(swap2BnbTokenAmount,address(address(this))) (BoboDogeCoin.sol#680)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        State variables written after the call(s):
```

```
(BoboDogeCoin.sol#703-709)
        State variables written after the call(s):
        - swapTokensForEth(swap2BnbTokenAmount,address(address(this))) (BoboDogeCoin.sol#680)
                - _allowances[owner][spender] = amount (BoboDogeCoin.sol#610)
Reentrancy in BoboDogeCoin.swapTokens(uint256) (BoboDogeCoin.sol#659-685):
        External calls:
        - swapTokensForEth(marketingAmount,marketingAddress) (BoboDogeCoin.sol#669)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(buyBackAmount,buyBackAddress) (BoboDogeCoin.sol#673)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(swap2BnbTokenAmount,address(address(this))) (BoboDogeCoin.sol#680)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - addLiquidity(swap2BnbTokenAmount,swapBnbAmount) (BoboDogeCoin.sol#684)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
        External calls sending eth:
        - addLiquidity(swap2BnbTokenAmount,swapBnbAmount) (BoboDogeCoin.sol#684)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
        State variables written after the call(s):
        - addLiquidity(swap2BnbTokenAmount,swapBnbAmount) (BoboDogeCoin.sol#684)
                - _allowances[owner][spender] = amount (BoboDogeCoin.sol#610)
Reentrancy in BoboDogeCoin.transferFrom(address,address,uint256) (BoboDogeCoin.sol#525-529):
        External calls:
        - _transfer(sender,recipient,amount) (BoboDogeCoin.sol#526)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (BoboDogeCoin.sol#526)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
        State variables written after the call(s):
        - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (BoboDogeC
```

```
        State variables written after the call(s):
        - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (BoboDogeC
oin.sol#527)
                - _allowances[owner][spender] = amount (BoboDogeCoin.sol#610)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in BoboDogeCoin._transfer(address,address,uint256) (BoboDogeCoin.sol#614-653):
        External calls:
        - swapTokens(contractTokenBalance) (BoboDogeCoin.sol#632)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        External calls sending eth:
        - swapTokens(contractTokenBalance) (BoboDogeCoin.sol#632)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
        Event emitted after the call(s):
        - Transfer(sender,recipient,tTransferAmount) (BoboDogeCoin.sol#773)
                - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
        - Transfer(sender,recipient,tTransferAmount) (BoboDogeCoin.sol#783)
                - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
        - Transfer(sender,recipient,tTransferAmount) (BoboDogeCoin.sol#793)
                - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
        - Transfer(sender,recipient,tTransferAmount) (BoboDogeCoin.sol#804)
                - _tokenTransfer(from,to,amount,takeFee) (BoboDogeCoin.sol#652)
Reentrancy in BoboDogeCoin.constructor() (BoboDogeCoin.sol#476-488):
        External calls:
        - uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this),_uniswapV2Router.WETH()) (BoboDogeCoin.s
ol#480)
        Event emitted after the call(s):
        - Transfer(address(0),_msgSender(),_tTotal) (BoboDogeCoin.sol#487)
Reentrancy in BoboDogeCoin.swapETHForTokens(uint256) (BoboDogeCoin.sol#714-729):
        External calls:
        - uniswapV2Router.swapExactETHForTokensSupportingFeeOnTransferTokens{value: amount}(0,path,deadAddress,block.timestamp.add(300))
(BoboDogeCoin.sol#721-726)
        Event emitted after the call(s):
        - SwapETHForTokens(amount,path) (BoboDogeCoin.sol#728)
```

```
        Event emitted after the call(s):
        - SwapETHForTokens(amount,path) (BoboDogeCoin.sol#728)
Reentrancy in BoboDogeCoin.swapTokens(uint256) (BoboDogeCoin.sol#659-685):
        External calls:
        - swapTokensForEth(marketingAmount,marketingAddress) (BoboDogeCoin.sol#669)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(buyBackAmount,buyBackAddress) (BoboDogeCoin.sol#673)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (BoboDogeCoin.sol#611)
                - swapTokensForEth(buyBackAmount,buyBackAddress) (BoboDogeCoin.sol#673)
        - SwapTokensForETH(tokenAmount,path) (BoboDogeCoin.sol#711)
                - swapTokensForEth(buyBackAmount,buyBackAddress) (BoboDogeCoin.sol#673)
Reentrancy in BoboDogeCoin.swapTokens(uint256) (BoboDogeCoin.sol#659-685):
        External calls:
        - swapTokensForEth(marketingAmount,marketingAddress) (BoboDogeCoin.sol#669)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(buyBackAmount,buyBackAddress) (BoboDogeCoin.sol#673)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(swap2BnbTokenAmount,address(address(this))) (BoboDogeCoin.sol#680)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (BoboDogeCoin.sol#611)
                - swapTokensForEth(swap2BnbTokenAmount,address(address(this))) (BoboDogeCoin.sol#680)
        - SwapTokensForETH(tokenAmount,path) (BoboDogeCoin.sol#711)
                - swapTokensForEth(swap2BnbTokenAmount,address(address(this))) (BoboDogeCoin.sol#680)
Reentrancy in BoboDogeCoin.swapTokens(uint256) (BoboDogeCoin.sol#659-685):
        External calls:
        - swapTokensForEth(marketingAmount,marketingAddress) (BoboDogeCoin.sol#669)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(buyBackAmount,buyBackAddress) (BoboDogeCoin.sol#673)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
```

```
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - swapTokensForEth(swap2BnbTokenAmount,address(address(this))) (BoboDogeCoin.sol#680)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        - addLiquidity(swap2BnbTokenAmount,swapBnbAmount) (BoboDogeCoin.sol#684)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
        External calls sending eth:
        - addLiquidity(swap2BnbTokenAmount,swapBnbAmount) (BoboDogeCoin.sol#684)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (BoboDogeCoin.sol#611)
                - addLiquidity(swap2BnbTokenAmount,swapBnbAmount) (BoboDogeCoin.sol#684)
Reentrancy in BoboDogeCoin.swapTokensForEth(uint256,address) (BoboDogeCoin.sol#694-712):
        External calls:
        - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360) (BoboDog
eCoin.sol#703-709)
        Event emitted after the call(s):
        - SwapTokensForETH(tokenAmount,path) (BoboDogeCoin.sol#711)
Reentrancy in BoboDogeCoin.transferFrom(address,address,uint256) (BoboDogeCoin.sol#525-529):
        External calls:
        - _transfer(sender,recipient,amount) (BoboDogeCoin.sol#526)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
                - uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,recipient,block.timestamp + 360)
(BoboDogeCoin.sol#703-709)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (BoboDogeCoin.sol#526)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp + 360) (BoboDog
eCoin.sol#736-743)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (BoboDogeCoin.sol#611)
                - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (B
oboDogeCoin.sol#527)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
```

```
INFO:Detectors:
Ownable.unlock() (BoboDogeCoin.sol#190-195) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp > _lockTime,Contract is locked until 7 days) (BoboDogeCoin.sol#192)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.isContract(address) (BoboDogeCoin.sol#85-94) uses assembly
        - INLINE ASM (BoboDogeCoin.sol#92)
Address._functionCallWithValue(address,bytes,uint256,string) (BoboDogeCoin.sol#122-139) uses assembly
        - INLINE ASM (BoboDogeCoin.sol#131-134)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Address._functionCallWithValue(address,bytes,uint256,string) (BoboDogeCoin.sol#122-139) is never used and should be removed
Address.functionCall(address,bytes) (BoboDogeCoin.sol#105-107) is never used and should be removed
Address.functionCall(address,bytes,string) (BoboDogeCoin.sol#109-111) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (BoboDogeCoin.sol#113-115) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (BoboDogeCoin.sol#117-120) is never used and should be removed
Address.isContract(address) (BoboDogeCoin.sol#85-94) is never used and should be removed
Address.sendValue(address,uint256) (BoboDogeCoin.sol#96-102) is never used and should be removed
BoboDogeCoin.transferToAddressETH(address,uint256) (BoboDogeCoin.sol#974-976) is never used and should be removed
Context._msgData() (BoboDogeCoin.sol#11-14) is never used and should be removed
SafeMath.mod(uint256,uint256) (BoboDogeCoin.sol#73-75) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (BoboDogeCoin.sol#77-80) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
BoboDogeCoin._tTotal (BoboDogeCoin.sol#423) is set pre-construction with a non-constant function or state variable:
        - 5200000000 * 10 ** 6 * 10 ** _decimals
BoboDogeCoin._rTotal (BoboDogeCoin.sol#425) is set pre-construction with a non-constant function or state variable:
        - (MAX - (MAX % _tTotal))
BoboDogeCoin._previousTaxFee (BoboDogeCoin.sol#431) is set pre-construction with a non-constant function or state variable:
        - _taxFee
BoboDogeCoin._previousLiquidityAndBuyBackFee (BoboDogeCoin.sol#437) is set pre-construction with a non-constant function or state variabl
e:
        - _liquidityAndBuyBackFee
BoboDogeCoin._maxTxAmount (BoboDogeCoin.sol#439) is set pre-construction with a non-constant function or state variable:
        - 52000000 * 10 ** 6 * 10 ** _decimals
BoboDogeCoin.minimumTokensBeforeSwap (BoboDogeCoin.sol#440) is set pre-construction with a non-constant function or state variable:
        - 200000 * 10 ** 6 * 10 ** _decimals
```

```
        - 52000000 * 10 ** 6 * 10 ** _decimals
BoboDogeCoin.minimumTokensBeforeSwap (BoboDogeCoin.sol#440) is set pre-construction with a non-constant function or state variable:
        - 200000 * 10 ** 6 * 10 ** _decimals
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state-variables
INFO:Detectors:
Pragma version^0.8.4 (BoboDogeCoin.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (BoboDogeCoin.sol#96-102):
        - (success) = recipient.call{value: amount}() (BoboDogeCoin.sol#100)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (BoboDogeCoin.sol#122-139):
        - (success,returndata) = target.call{value: weiValue}(data) (BoboDogeCoin.sol#125)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (BoboDogeCoin.sol#230) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (BoboDogeCoin.sol#231) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (BoboDogeCoin.sol#247) is not in mixedCase
Function IUniswapV2Router01.WETH() (BoboDogeCoin.sol#266) is not in mixedCase
Parameter BoboDogeCoin.calculateTaxFee(uint256)._amount (BoboDogeCoin.sol#858) is not in mixedCase
Parameter BoboDogeCoin.calculateLiquidityFee(uint256)._amount (BoboDogeCoin.sol#864) is not in mixedCase
Parameter BoboDogeCoin.setNumTokensSellToAddToLiquidity(uint256)._minimumTokensBeforeSwap (BoboDogeCoin.sol#924) is not in mixedCase
Parameter BoboDogeCoin.setMarketingAddress(address)._marketingAddress (BoboDogeCoin.sol#932) is not in mixedCase
Parameter BoboDogeCoin.setBuyBackAddress(address)._buyBackAddress (BoboDogeCoin.sol#936) is not in mixedCase
Parameter BoboDogeCoin.setSwapAndLiquifyEnabled(bool)._enabled (BoboDogeCoin.sol#940) is not in mixedCase
Parameter BoboDogeCoin.setBuyBackEnabled(bool)._enabled (BoboDogeCoin.sol#945) is not in mixedCase
Parameter BoboDogeCoin.addWhiteList(address)._account (BoboDogeCoin.sol#950) is not in mixedCase
Parameter BoboDogeCoin.removeWhiteList(address)._account (BoboDogeCoin.sol#955) is not in mixedCase
Variable BoboDogeCoin._taxFee (BoboDogeCoin.sol#428) is not in mixedCase
Variable BoboDogeCoin._taxFeeOfWhiteList (BoboDogeCoin.sol#429) is not in mixedCase
Variable BoboDogeCoin._liquidityAndBuyBackFee (BoboDogeCoin.sol#435) is not in mixedCase
Variable BoboDogeCoin._liquidityAndBuyBackFeeOfWhiteList (BoboDogeCoin.sol#436) is not in mixedCase
Variable BoboDogeCoin._maxTxAmount (BoboDogeCoin.sol#439) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (BoboDogeCoin.sol#12)" inContext (BoboDogeCoin.sol#6-15)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (BoboDogeCoin.sol#12)" inContext (BoboDogeCoin.sol#6-15)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (BoboDogeCoin.sol#271) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (BoboDogeCoin.sol#272)
Variable BoboDogeCoin._liquidityAndBuyBackFeeOfWhiteList (BoboDogeCoin.sol#436) is too similar to BoboDogeCoin.updateFeeOfWhiteList(uint256,uint256).liquidityAndBuyBackFeeOfWhiteList_ (BoboDogeCoin.sol#895)
Variable BoboDogeCoin._taxFeeOfWhiteList (BoboDogeCoin.sol#429) is too similar to BoboDogeCoin.updateFeeOfWhiteList(uint256,uint256).taxFeeOfWhiteList_ (BoboDogeCoin.sol#895)
Variable BoboDogeCoin._transferBothExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#797) is too similar to BoboDogeCoin._transferBothExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#797)
Variable BoboDogeCoin._transferBothExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#797) is too similar to BoboDogeCoin._transferToExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#777)
Variable BoboDogeCoin._transferBothExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#797) is too similar to BoboDogeCoin._getTValues(uint256).tTransferAmount (BoboDogeCoin.sol#821)
Variable BoboDogeCoin._transferStandard(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#768) is too similar to BoboDogeCoin._transferFromExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#787)
Variable BoboDogeCoin._transferToExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#777) is too similar to BoboDogeCoin._transferToExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#777)
Variable BoboDogeCoin._transferToExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#777) is too similar to BoboDogeCoin._getTValues(uint256).tTransferAmount (BoboDogeCoin.sol#821)
Variable BoboDogeCoin._getValues(uint256).rTransferAmount (BoboDogeCoin.sol#814) is too similar to BoboDogeCoin._transferFromExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#787)
Variable BoboDogeCoin._transferStandard(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#768) is too similar to BoboDogeCoin._transferStandard(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#768)
Variable BoboDogeCoin._transferFromExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#787) is too similar to BoboDogeCoin._transferFromExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#787)
Variable BoboDogeCoin._transferBothExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#797) is too similar to BoboDogeCoin._transferStandard(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#768)
Variable BoboDogeCoin._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (BoboDogeCoin.sol#829) is too similar to BoboDogeCoin._transferToExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#777)
Variable BoboDogeCoin._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (BoboDogeCoin.sol#829) is too similar to BoboDogeCoin._getTValues(uint256).tTransferAmount (BoboDogeCoin.sol#821)
Variable BoboDogeCoin._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (BoboDogeCoin.sol#829) is too similar to BoboDogeCoin._transferFromExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#787)
```

```
_transferToExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#777)
Variable BoboDogeCoin._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (BoboDogeCoin.sol#829) is too similar to BoboDogeCoin._getTValues(uint256).tTransferAmount (BoboDogeCoin.sol#821)
Variable BoboDogeCoin._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (BoboDogeCoin.sol#829) is too similar to BoboDogeCoin._transferFromExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#787)
Variable BoboDogeCoin._transferToExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#777) is too similar to BoboDogeCoin._transferStandard(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#768)
Variable BoboDogeCoin._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (BoboDogeCoin.sol#829) is too similar to BoboDogeCoin._transferBothExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#797)
Variable BoboDogeCoin._transferBothExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#797) is too similar to BoboDogeCoin._transferFromExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#787)
Variable BoboDogeCoin._transferStandard(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#768) is too similar to BoboDogeCoin._getTValues(uint256).tTransferAmount (BoboDogeCoin.sol#821)
Variable BoboDogeCoin._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (BoboDogeCoin.sol#829) is too similar to BoboDogeCoin._getValues(uint256).tTransferAmount (BoboDogeCoin.sol#813)
Variable BoboDogeCoin._transferBothExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#797) is too similar to BoboDogeCoin._getValues(uint256).tTransferAmount (BoboDogeCoin.sol#813)
Variable BoboDogeCoin._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (BoboDogeCoin.sol#829) is too similar to BoboDogeCoin._transferStandard(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#768)
Variable BoboDogeCoin._transferToExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#777) is too similar to BoboDogeCoin._transferFromExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#787)
Variable BoboDogeCoin._transferToExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#777) is too similar to BoboDogeCoin._getValues(uint256).tTransferAmount (BoboDogeCoin.sol#813)
Variable BoboDogeCoin._transferToExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#777) is too similar to BoboDogeCoin._transferBothExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#797)
Variable BoboDogeCoin._transferFromExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#787) is too similar to BoboDogeCoin._transferBothExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#797)
Variable BoboDogeCoin._transferFromExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#787) is too similar to BoboDogeCoin._transferStandard(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#768)
Variable BoboDogeCoin.reflectionFromToken(uint256,bool).rTransferAmount (BoboDogeCoin.sol#572) is too similar to BoboDogeCoin._transferStandard(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#768)
Variable BoboDogeCoin._transferStandard(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#768) is too similar to BoboDogeCoin._getValues(uint256).tTransferAmount (BoboDogeCoin.sol#813)
Variable BoboDogeCoin.reflectionFromToken(uint256,bool).rTransferAmount (BoboDogeCoin.sol#572) is too similar to BoboDogeCoin._transferToExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#777)
Variable BoboDogeCoin._getValues(uint256).rTransferAmount (BoboDogeCoin.sol#814) is too similar to BoboDogeCoin._getValues(uint256).tTransferAmount (BoboDogeCoin.sol#813)
Variable BoboDogeCoin.reflectionFromToken(uint256,bool).rTransferAmount (BoboDogeCoin.sol#572) is too similar to BoboDogeCoin._getTValues
```

```
ransferBothExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#797)
Variable BoboDogeCoin.reflectionFromToken(uint256,bool).rTransferAmount (BoboDogeCoin.sol#572) is too similar to BoboDogeCoin._transferBothExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#797)
Variable BoboDogeCoin._transferFromExcluded(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#787) is too similar to BoboDogeCoin._getValues(uint256).tTransferAmount (BoboDogeCoin.sol#813)
Variable BoboDogeCoin._transferStandard(address,address,uint256).rTransferAmount (BoboDogeCoin.sol#768) is too similar to BoboDogeCoin._transferToExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#777)
Variable BoboDogeCoin._getValues(uint256).rTransferAmount (BoboDogeCoin.sol#814) is too similar to BoboDogeCoin._transferBothExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#797)
Variable BoboDogeCoin._getValues(uint256).rTransferAmount (BoboDogeCoin.sol#814) is too similar to BoboDogeCoin._transferStandard(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#768)
Variable BoboDogeCoin._getValues(uint256).rTransferAmount (BoboDogeCoin.sol#814) is too similar to BoboDogeCoin._transferToExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#777)
Variable BoboDogeCoin.reflectionFromToken(uint256,bool).rTransferAmount (BoboDogeCoin.sol#572) is too similar to BoboDogeCoin._getValues(uint256).tTransferAmount (BoboDogeCoin.sol#813)
Variable BoboDogeCoin._getValues(uint256).rTransferAmount (BoboDogeCoin.sol#814) is too similar to BoboDogeCoin._getTValues(uint256).tTransferAmount (BoboDogeCoin.sol#821)
Variable BoboDogeCoin.reflectionFromToken(uint256,bool).rTransferAmount (BoboDogeCoin.sol#572) is too similar to BoboDogeCoin._transferFromExcluded(address,address,uint256).tTransferAmount (BoboDogeCoin.sol#787)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
BoboDogeCoin.prepareForPreSale() (BoboDogeCoin.sol#960-965) uses literals with too many digits:
        - _maxTxAmount = 5200000000 * 10 ** 6 * 10 ** _decimals (BoboDogeCoin.sol#964)
BoboDogeCoin.afterPreSale() (BoboDogeCoin.sol#967-972) uses literals with too many digits:
        - _maxTxAmount = 52000000 * 10 ** 6 * 10 ** _decimals (BoboDogeCoin.sol#971)
BoboDogeCoin.slitherConstructorVariables() (BoboDogeCoin.sol#400-981) uses literals with too many digits:
        - deadAddress = 0x000000000000000000000000000000000000dEaD (BoboDogeCoin.sol#406)
BoboDogeCoin.slitherConstructorVariables() (BoboDogeCoin.sol#400-981) uses literals with too many digits:
        - _tTotal = 5200000000 * 10 ** 6 * 10 ** _decimals (BoboDogeCoin.sol#423)
BoboDogeCoin.slitherConstructorVariables() (BoboDogeCoin.sol#400-981) uses literals with too many digits:
        - _maxTxAmount = 52000000 * 10 ** 6 * 10 ** _decimals (BoboDogeCoin.sol#439)
BoboDogeCoin.slitherConstructorVariables() (BoboDogeCoin.sol#400-981) uses literals with too many digits:
        - minimumTokensBeforeSwap = 200000 * 10 ** 6 * 10 ** _decimals (BoboDogeCoin.sol#440)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
BoboDogeCoin._decimals (BoboDogeCoin.sol#420) should be constant
BoboDogeCoin._name (BoboDogeCoin.sol#418) should be constant
BoboDogeCoin._symbol (BoboDogeCoin.sol#419) should be constant
```

```
BoboDogeCoin._symbol (BoboDogeCoin.sol#419) should be constant
BoboDogeCoin.buyBackRate (BoboDogeCoin.sol#434) should be constant
BoboDogeCoin.deadAddress (BoboDogeCoin.sol#406) should be constant
BoboDogeCoin.marketingRate (BoboDogeCoin.sol#433) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (BoboDogeCoin.sol#164-167)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (BoboDogeCoin.sol#169-173)
getUnlockTime() should be declared external:
        - Ownable.getUnlockTime() (BoboDogeCoin.sol#175-177)
getTime() should be declared external:
        - Ownable.getTime() (BoboDogeCoin.sol#179-181)
lock(uint256) should be declared external:
        - Ownable.lock(uint256) (BoboDogeCoin.sol#183-188)
unlock() should be declared external:
        - Ownable.unlock() (BoboDogeCoin.sol#190-195)
name() should be declared external:
        - BoboDogeCoin.name() (BoboDogeCoin.sol#490-492)
symbol() should be declared external:
        - BoboDogeCoin.symbol() (BoboDogeCoin.sol#494-496)
decimals() should be declared external:
        - BoboDogeCoin.decimals() (BoboDogeCoin.sol#498-500)
totalSupply() should be declared external:
        - BoboDogeCoin.totalSupply() (BoboDogeCoin.sol#502-504)
transfer(address,uint256) should be declared external:
        - BoboDogeCoin.transfer(address,uint256) (BoboDogeCoin.sol#511-514)
allowance(address,address) should be declared external:
        - BoboDogeCoin.allowance(address,address) (BoboDogeCoin.sol#516-518)
approve(address,uint256) should be declared external:
        - BoboDogeCoin.approve(address,uint256) (BoboDogeCoin.sol#520-523)
transferFrom(address,address,uint256) should be declared external:
        - BoboDogeCoin.transferFrom(address,address,uint256) (BoboDogeCoin.sol#525-529)
increaseAllowance(address,uint256) should be declared external:
        - BoboDogeCoin.increaseAllowance(address,uint256) (BoboDogeCoin.sol#531-534)
decreaseAllowance(address,uint256) should be declared external:
        - BoboDogeCoin.decreaseAllowance(address,uint256) (BoboDogeCoin.sol#536-539)
```

```
decreaseAllowance(address,uint256) should be declared external:
        - BoboDogeCoin.decreaseAllowance(address,uint256) (BoboDogeCoin.sol#536-539)
isExcludedFromReward(address) should be declared external:
        - BoboDogeCoin.isExcludedFromReward(address) (BoboDogeCoin.sol#541-543)
totalFees() should be declared external:
        - BoboDogeCoin.totalFees() (BoboDogeCoin.sol#545-547)
minimumTokensBeforeSwapAmount() should be declared external:
        - BoboDogeCoin.minimumTokensBeforeSwapAmount() (BoboDogeCoin.sol#549-551)
buyBackUpperLimitAmount() should be declared external:
        - BoboDogeCoin.buyBackUpperLimitAmount() (BoboDogeCoin.sol#553-555)
deliver(uint256) should be declared external:
        - BoboDogeCoin.deliver(uint256) (BoboDogeCoin.sol#557-564)
reflectionFromToken(uint256,bool) should be declared external:
        - BoboDogeCoin.reflectionFromToken(uint256,bool) (BoboDogeCoin.sol#566-575)
excludeFromReward(address) should be declared external:
        - BoboDogeCoin.excludeFromReward(address) (BoboDogeCoin.sol#583-591)
isExcludedFromFee(address) should be declared external:
        - BoboDogeCoin.isExcludedFromFee(address) (BoboDogeCoin.sol#900-902)
excludeFromFee(address) should be declared external:
        - BoboDogeCoin.excludeFromFee(address) (BoboDogeCoin.sol#904-906)
includeInFee(address) should be declared external:
        - BoboDogeCoin.includeInFee(address) (BoboDogeCoin.sol#908-910)
setBuyBackEnabled(bool) should be declared external:
        - BoboDogeCoin.setBuyBackEnabled(bool) (BoboDogeCoin.sol#945-948)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:BoboDogeCoin.sol analyzed (10 contracts with 75 detectors), 147 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
root@server:/chetan/gaza/mycontracts#
```

# Solidity static analysis

**BoboDogeCoin.sol**



**Email: audit@EtherAuthority.io**

SOLIDITY STATIC ANALYSIS

contracts/BoboDogeCoin.sol

Security

**Transaction origin:**

INTERNAL ERROR in module Transaction origin: can't convert undefined to object
Pos: not available

**Check-effects-interaction:**

INTERNAL ERROR in module Check-effects-interaction: can't convert undefined to object
Pos: not available

**Inline assembly:**

INTERNAL ERROR in module Inline assembly: can't convert undefined to object
Pos: not available

**Block timestamp:**

INTERNAL ERROR in module Block timestamp: can't convert undefined to object
Pos: not available

**Low level calls:**

INTERNAL ERROR in module Low level calls: can't convert undefined to object
Pos: not available

**Selfdestruct:**

INTERNAL ERROR in module Selfdestruct: can't convert undefined to object
Pos: not available

Gas & Economy

**This on local calls:**

INTERNAL ERROR in module This on local calls: can't convert undefined to object
Pos: not available

**Delete dynamic array:**

INTERNAL ERROR in module Delete dynamic array: can't convert undefined to object
Pos: not available

**For loop over dynamic array:**

INTERNAL ERROR in module For loop over dynamic array: can't convert undefined to object
Pos: not available

**Ether transfer in loop:**

INTERNAL ERROR in module Ether transfer in loop: can't convert undefined to object
Pos: not available


## Solhint Linter

### BoboDogeCoin.sol

```
contracts/BoboDogeCoin.sol:22:1: Error: Compiler version ^0.8.4 does not satisfy the r
semver requirement

contracts/BoboDogeCoin.sol:144:51: Error: Avoid using low level calls.

contracts/BoboDogeCoin.sol:150:17: Error: Avoid using inline assembly. It is acceptable
only in rare cases

contracts/BoboDogeCoin.sol:168:5: Error: Explicitly mark visibility in function (Set
ignoreConstructors to true if using solidity >=0.7.0)

contracts/BoboDogeCoin.sol:199:16: Error: Avoid to make time-based decisions in your
business logic
```

```
contracts/BoboDogeCoin.sol:205:21: Error: Avoid to make time-based decisions in your business logic

contracts/BoboDogeCoin.sol:211:17: Error: Avoid to make time-based decisions in your business logic

contracts/BoboDogeCoin.sol:253:5: Error: Function name must be in mixedCase

contracts/BoboDogeCoin.sol:254:5: Error: Function name must be in mixedCase

contracts/BoboDogeCoin.sol:270:5: Error: Function name must be in mixedCase

contracts/BoboDogeCoin.sol:291:5: Error: Function name must be in mixedCase

contracts/BoboDogeCoin.sol:427:1: Error: Contract has 29 states declarations but allowed no more than 15

contracts/BoboDogeCoin.sol:473:5: Error: Explicitly mark visibility of state

contracts/BoboDogeCoin.sol:503:5: Error: Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)

contracts/BoboDogeCoin.sol:735:13: Error: Avoid to make time-based decisions in your business logic

contracts/BoboDogeCoin.sol:752:13: Error: Avoid to make time-based decisions in your business logic

contracts/BoboDogeCoin.sol:769:13: Error: Avoid to make time-based decisions in your business logic

contracts/BoboDogeCoin.sol:1006:32: Error: Code contains empty blocks
```




**Email: audit@EtherAuthority.io**