Cyberscope

Audit Report

MrBean Token

July 2023

Network	BSC
Address	0x3EDA68D8357782F47E94abD6f5fd4d9a644831D5
Audited by	© cyberscope

Analysis

Critical Medium Minor / Informative Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	RSW	Redundant Storage Writes	Unresolved
●	IDI	Immutable Declaration Improvement	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L09	Dead Code Elimination	Unresolved

Table of Contents

Review

Contract Name	MrBeanToken
Compiler Version	v0.8.16+commit.07a7930e
Optimization	200 runs
Explorer	https://bscscan.com/address/0x3eda68d8357782f47e94abd6f5fd4d9a644831d5
Address	0x3eda68d8357782f47e94abd6f5fd4d9a644831d5
Network	BSC
Symbol	MRBEANT
Decimals	18
Total Supply	1,000,000,000,000

Audit Updates

Initial Audit	06 Jul 2023

Source Files

Filename	SHA256
MrBeanToken.sol	ae331e0e31d93161074f9c45f4a342e516c801f7212cd7f1357e93cf407db7a7

Findings Breakdown

Critical	0
Medium	0
Minor / Informative	4

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	4	0	0	0

RSW - Redundant Storage Writes

Criticality	Minor / Informative
Location	MrBeanToken.sol#L524
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

The contract modifies the state of the following variables without checking if their current value is the same as the one given as an argument. As a result, the contract performs redundant storage writes.

```
function setMarketingWallet(address payable wallet) external onlyOwner
{
    require(wallet != address(0), "Can not be address(0).");

    marketingWalletAddress = wallet;

    emit UpdatedMarketingWallet(wallet);
}
```

Recommendation

The team is advised to take these segments into consideration and rewrite them so the runtime will be more performant. That way it will improve the efficiency and performance of the source code and reduce the cost of executing it.

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	MrBeanToken.sol#L431
Status	Unresolved

Description

The contract declares state variables that their value is initialized once in the constructor and are not modified afterwards. The `immutable` is a special declaration for this kind of state variables that saves gas when it is defined.

```
uniswapV2Router
```

Recommendation

By declaring a variable as immutable, the Solidity compiler is able to make certain optimizations. This can reduce the amount of storage and computation required by the contract, and make it more gas-efficient.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	MrBeanToken.sol#L273,409,410
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
function WETH() external pure returns (address);
event addLiquidityETH(uint256 eth, uint256 tokens, address receiver);
event transferTokens(uint256 _value, address receiver);
```

Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation
https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

L09 - Dead Code Elimination

Criticality	Minor / Informative
Location	MrBeanToken.sol#L192
Status	Unresolved

Description

In Solidity, dead code is code that is written in the contract, but is never executed or reached during normal contract execution. Dead code can occur for a variety of reasons, such as:

- Conditional statements that are always false.
- Functions that are never called.
- Unreachable code (e.g., code that follows a return statement).

Dead code can make a contract more difficult to understand and maintain, and can also increase the size of the contract and the cost of deploying and interacting with it.

```
function _burn(address account, uint256 amount) internal virtual {
        require(account != address(0), "ERC20: burn from the zero
address");

        uint256 accountBalance = _balances[account];
        require(accountBalance >= amount, "ERC20: burn amount exceeds
balance");
        unchecked {
            _balances[account] = accountBalance - amount;
        }
        _totalSupply -= amount;

        emit Transfer(account, address(0), amount);
    }
```

Recommendation

To avoid creating dead code, it's important to carefully consider the logic and flow of the contract and to remove any code that is not needed or that is never executed. This can help improve the clarity and efficiency of the contract.

Functions Analysis

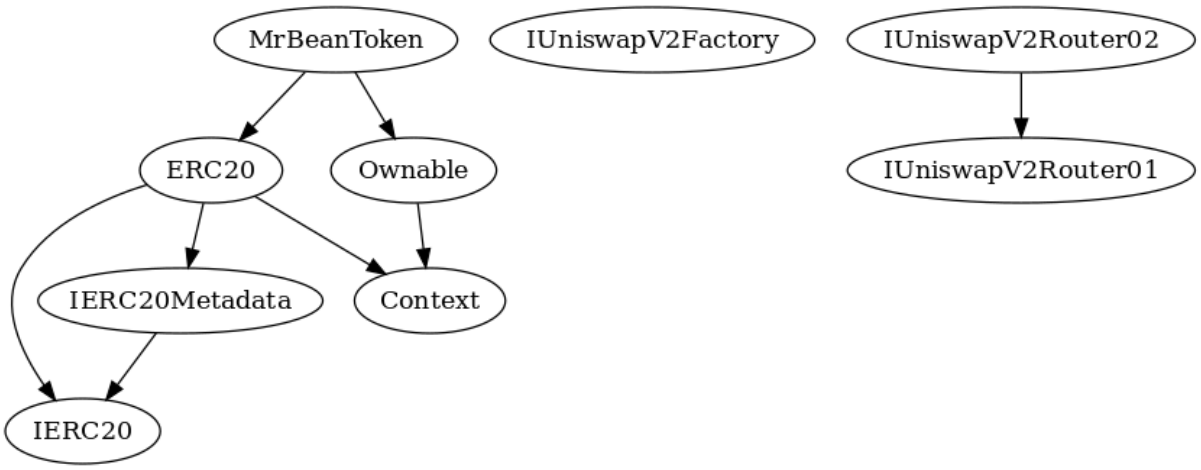
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC20	Implementation	Context, IERC20,		

		IERC20Metadata		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

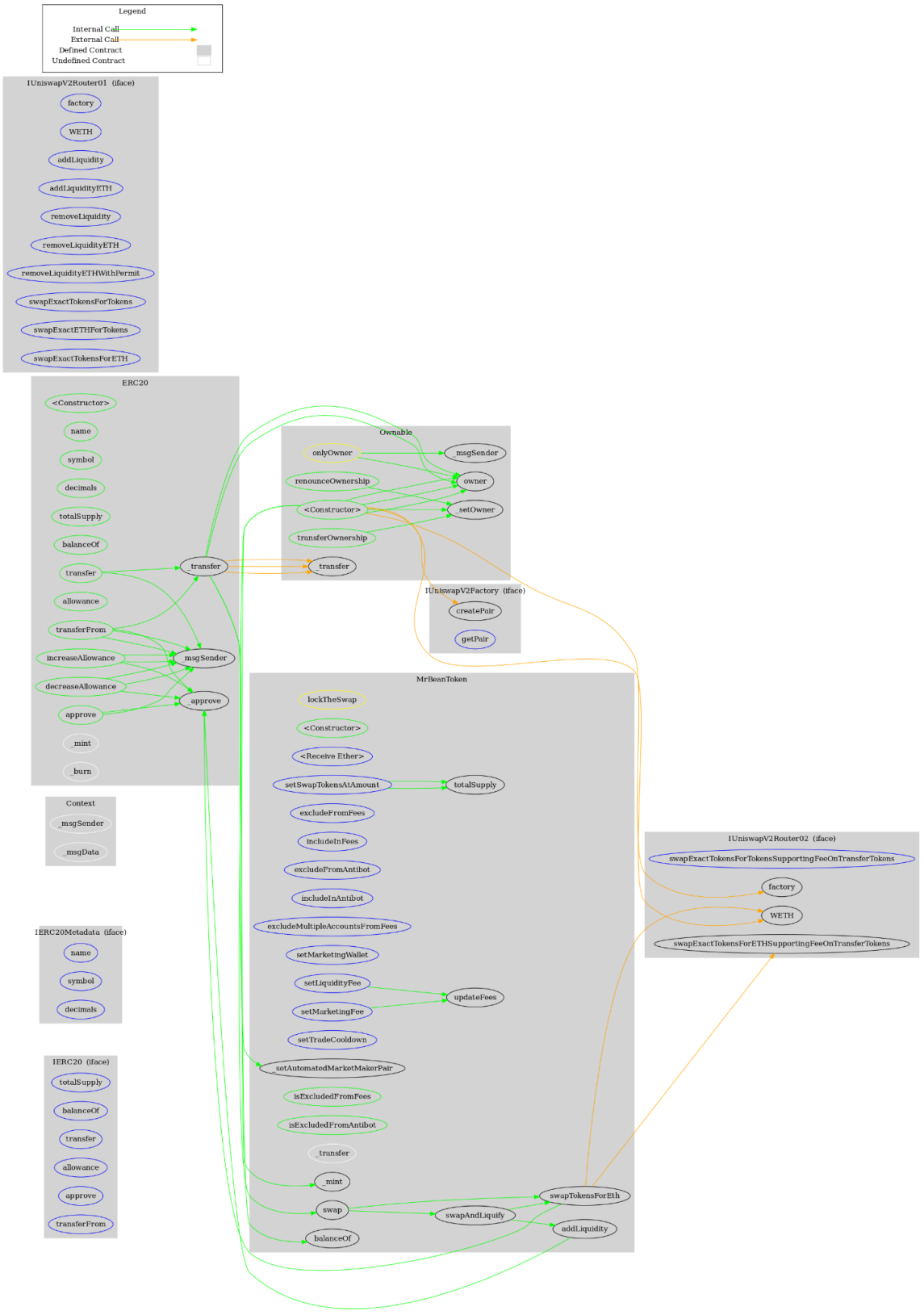
	_setOwner	Private	✓	
IUniswapV2Factory	Interface			
	createPair	External	✓	-
	getPair	External		-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapExactTokensForETH	External	✓	-
IUniswapV2Router02	Interface	IUniswapV2 Router01		
	swapExactTokensForTokensSupporting FeeOnTransferTokens	External	✓	-
	swapExactTokensForETHSupportingFee OnTransferTokens	External	✓	-

MrBeanToken	Implementation	ERC20, Ownable		
		Public	Payable	ERC20
		External	Payable	-
	setSwapTokensAtAmount	External	✓	onlyOwner
	excludeFromFees	External	✓	onlyOwner
	includeInFees	External	✓	onlyOwner
	excludeFromAntibot	External	✓	onlyOwner
	includeInAntibot	External	✓	onlyOwner
	excludeMultipleAccountsFromFees	External	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setLiquidityFee	External	✓	onlyOwner
	setMarketingFee	External	✓	onlyOwner
	setTradeCooldown	External	✓	onlyOwner
	updateFees	Internal	✓	
	_setAutomatedMarketMakerPair	Private	✓	
	isExcludedFromFees	Public		-
	isExcludedFromAntibot	Public		-
	_transfer	Internal	✓	
	swap	Private	✓	lockTheSwap
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IUniswapV2Router01 (iface)
factory
WETH
addLiquidity
addLiquidityETH
removeLiquidity
removeLiquidityETH
removeLiquidityETHWithPermit
swapExactTokensForTokens
swapExactETHForTokens
swapExactTokensForETH
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transfer
allowance
transferFrom
increaseAllowance
decreaseAllowance
approve
_mint
_burn
_transfer
msgSender
_approve
Ownable
onlyOwner
renounceOwnership
<Constructor>
transferOwnership
msgSender
owner
setOwner
_transfer
IUniswapV2Factory (iface)
createPair
getPair
MrBeanToken
lockTheSwap
<Constructor>
<Receive Ether>
setSwapTokensAtAmount
totalSupply
excludeFromFees
includeInFees
excludeFromAntibot
includeInAntibot
excludeMultipleAccountsFromFees
setMarketingWallet
setLiquidityFee
setMarketingFee
updateFees
setTradeCooldown
setAutomatedMarketMakerPair
isExcludedFromFees
isExcludedFromAntibot
_transfer
_mint
swap
balanceOf
swapAndLiquify
swapTokensForEth
addLiquidity
IUniswapV2Router02 (iface)
swapExactTokensForTokensSupportingFeeOnTransferTokens
factory
WETH
swapExactTokensForETHSupportingFeeOnTransferTokens
Context
_msgSender
msgData
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom

Summary

MrBean Token contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. MrBean Token is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 20% fees. Additionally, the contract has an antibot mechanism.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io