



RugFreeCoins Audit



Meta Shiba inu Token

Smart Contract Security Audit

February 28, 2022

Contents

Audit details	1
Disclaimer	2
Background	3
About the project	4
Target market and the concept	8
Potential to grow with score points	9
Total Points	9
Contract details	10
Contract code function details	12
Contract description table	14
Security issue checking status	23
Audit conclusion	27

Audit details



Audited project
Meta shiba inu Token



Contract Address
0x13A62150224CE665A95B6D8Ef2f0DAEa9a14a44c



Client contact
Meta Shiba Inu Team



Blockchain
Binance smart chain



Project website
<https://metashibainu.info/>

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Rugfreecoins hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

Rugfreecoins was commissioned by the Meta Shiba Inu Team to perform an audit of the smart contract.

<https://bscscan.com/token/0x13A62150224CE665A95B6D8Ef2f0DAEa9a14a44c>

The focus of this audit is to verify that the smart contract is secure, resilient, and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, long-term sustainability, and as a guide to improving the security posture of the smart contract by remediating the issues that were identified.

About the project

Meta Shiba Inu is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources, which in turn, powers reward combined with the VR entertainment and VR gaming. Each transaction, purchase, and sale incur an 11% fee.

Features

- The **BUSD rewards** will be distributed among every holder proportional to how many tokens each individual holds in values of **6% when buying and selling**.
- The **sustainability fee of 2% when buying and selling for marketing** is what allows Meta Shiba Inu to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, Meta Shiba Inu will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.
- The additional component included under the sustainability section is a **liquidity fee of 1% from buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.
- Meta Shiba Inu has the buyback and burn strategy that a **2% fee in each transaction when buying selling** is getting charged that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Meta Shiba Inu more and more valuable.

Roadmap

66

- ▶ Website Launch
- ▶ Create Token Contract
- ▶ Contract Address Audit
- ▶ Public Sale
- ▶ Pancake Listing
- ▶ TrustWallet Listing
- ▶ Coingecko Listing
- ▶ Coinmarketcap Listing



2022

Phase-01 | Q1


66

- ▶ Start developing Metaverse
- ▶ CEX Listing
- ▶ 3rd party promoter Listing
- ▶ Userbase increment generator program launch
- ▶ 50,000 Holder Target

2022

Phase-02 | Q2


“

- ▶ 100,000 Holder Target
- ▶ Midium Scale marketing launch
- ▶ Top CEX Listing
- ▶ Deploy demo VR [metaverse] based gaming with implementation of \$MSHIB
- ▶ Upgrade Website v2
- ▶ Develop & deploy \$MSHIB powered #metaverse/ VR #Gaming platform
- ▶ A vast massive marketing campaign to gain mainstream attention

2022**Phase-03 | Q2-Q4**

Tokenomics

15% fee when selling

- 6% of trade goes to holders' pockets in BUSD.
- 2% of trade goes to the marketing wallet in BNB.
- 2% of trade goes to the buyback and burn.
- 1% of trade goes to the liquidity pool.

Target market and the concept

Target market

- Anyone who's interested in the Crypto space with long-term investment plans.
- Anyone who's ready to earn a passive income in BUSD by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in taking part with meta Shiba Inu AR entertainment and VR gaming.
- Anyone who's interested in taking part with the future plans of the Meta Shiba Inu token.
- Anyone who's interested in making financial transactions with any other party using Meta Shiba Inu as the currency.

Core concept

The Meta Shiba Inu reward system

6% of each transaction when buying and selling get converted to BUSD and is split amongst all holders. Holders will be eligible to receive tokens every one hour and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The **sustainability fee of 2% when buying and selling for marketing** is what allows Meta Shiba Inu to promote the token and use funds to further the development of the platform. Tokens will be swapped into BNB and will be sent to a marketing wallet. This way, Meta Shiba Inu will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

The liquidity fee of 1% is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Meta Shiba Inu has the buyback and burn strategy that a **2% fee in each transaction when buying and selling** is getting charged that benefits and rewards those who invest long-term. This feature slowly reduces supply making each Meta Shiba Inu more and more valuable.

Potential to grow with score points

1.	Project efficiency	9/10
2.	Project uniqueness	9/10
3	Information quality	8/10
4	Service quality	8/10
5	System quality	8/10
6	Impact on the community	9/10
7	Impact on the business	9/10
8	Preparing for the future	9/10
Total Points		8.625/10

Contract details

Token contract details for 28th February 2022

Contract name	Meta Shiba Inu
Contract address	0x13A62150224CE665A95B6D8Ef2f0DAEa9a14a44c
Token supply	1,000,000,000,000,000
Token ticker	MSHIB
Decimals	9
Token holders	3
Transaction count	6
Dividend tracker	0xf02bbec7679c8b9875e65b1e2e16f89c807e5394
Reward Token	0xe9e7cea3dedca5984780bafc599bd69add087d56
Marketing wallet	0xb6743b1cd4296c0d3c2adb1aa171f7c23d6d3411
Contract deployer address	0x51F2333a56d5AdE9d71cDcfa3BBe2c1E3283774d
Contract's current owner address	Not public

Tokens are distributed as follows:

—— TOKEN SUPPLY

MSHIB Tokenomics

SMART CONTRACT 100%

 0x13A62150224CE665A95B6D8Ef2f0DAEa9a14a44c

MAX SUPPLY 100%

 1,000,000,000,000,000 MSHIB

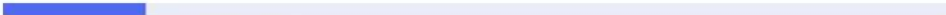
PRESALE AND LIQUIDITY 80%

 Presale and Liquidity by PinkSale. [Liquidity Locked for 12 months]

MARKETING 5%

 Promotions, Giveaways and Airdrops. [Locked for 2 months]

TEAM 15%

 [Locked for 3-6 months]














Contract code function details

































No	Category	Item	Result
1	Coding conventions	BRC20 Token standards	pass
		compile errors	pass
		Compiler version security	pass
		visibility specifiers	pass
		Gas consumption	pass
		SafeMath features	pass
		Fallback usage	pass
		tx.origin usage	pass
		deprecated items	pass
		Redundant code	pass
		Overriding variables	pass
2	Function call audit	Authorization of function call	pass
		Low level function (call/delegate call) security	pass
		Returned value security	pass
		Selfdestruct function security	pass
3	Business security	Access control of owners	pass
		Business logics	pass
		Business implementations	pass
4	Integer overflow/underflow		pass
5	Reentrancy		pass
6	Exceptional reachable state		pass













7	Transaction ordering dependence		pass
8	Block properties dependence		pass
9	Pseudo random number generator (PRNG)		pass
10	DoS (Denial of Service)		pass
11	Token vesting implementation		pass
12	Fake deposit		pass
13	Event security		pass










Contract description table














The below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions, and implementations with their visibility and mutability.

































Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
L	tryAdd	Internal 		
L	trySub	Internal 		
L	tryMul	Internal 		
L	tryDiv	Internal 		
L	tryMod	Internal 		
L	add	Internal 		
L	sub	Internal 		
L	mul	Internal 		
L	div	Internal 		
L	mod	Internal 		
L	sub	Internal 		
L	div	Internal 		
L	mod	Internal 		

Clones	Library			
L	clone	Internal 		
L	cloneDeterministic	Internal 		
L	predictDeterministicAddress	Internal 		
L	predictDeterministicAddress	Internal 		
IUniswapV2Factory	Interface			
L	feeTo	External 		NO 
L	feeToSetter	External 		NO 
L	getPair	External 		NO 
L	allPairs	External 		NO 
L	allPairsLength	External 		NO 
L	createPair	External 		NO 
L	setFeeTo	External 		NO 
L	setFeeToSetter	External 		NO 
IUniswapV2Router01	Interface			
L	factory	External 		NO 
L	WETH	External 		NO 
L	addLiquidity	External 		NO 





























L	addLiquidityETH	External ¶		NO¶
L	removeLiquidity	External ¶		NO¶
L	removeLiquidityETH	External ¶		NO¶
L	removeLiquidityWithPermit	External ¶		NO¶
L	removeLiquidityETHWithPermit	External ¶		NO¶
L	swapExactTokensForTokens	External ¶		NO¶
L	swapTokensForExactTokens	External ¶		NO¶
L	swapExactETHForTokens	External ¶		NO¶
L	swapTokensForExactETH	External ¶		NO¶
L	swapExactTokensForETH	External ¶		NO¶
L	swapETHForExactTokens	External ¶		NO¶
L	quote	External ¶		NO¶
L	getAmountOut	External ¶		NO¶
L	getAmountIn	External ¶		NO¶
L	getAmountsOut	External ¶		NO¶
L	getAmountsIn	External ¶		NO¶
IUniswapV2Router02	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External ¶		NO¶

L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ¶		NO ¶
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ¶		NO ¶
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ¶		NO ¶
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ¶		NO ¶
IERC20Extended	Interface			
L	totalSupply	External ¶		NO ¶
L	decimals	External ¶		NO ¶
L	symbol	External ¶		NO ¶
L	name	External ¶		NO ¶
L	balanceOf	External ¶		NO ¶
L	transfer	External ¶		NO ¶
L	allowance	External ¶		NO ¶
L	approve	External ¶		NO ¶
L	transferFrom	External ¶		NO ¶
IPinkAntiBot	Interface			
L	setTokenOwner	External ¶		NO ¶
L	onPreTransferCheck	External ¶		NO ¶



Auth	Implementation			
L		Public ¶		NO¶
L	authorize	Public ¶		onlyOwner
L	unauthorize	Public ¶		onlyOwner
L	isOwner	Public ¶		NO¶
L	isAuthorized	Public ¶		NO¶
L	transferOwnership	Public ¶		onlyOwner
IDividendDistri butor	Interface			
L	setDistributionCriteria	External ¶		NO¶
L	setShare	External ¶		NO¶
L	deposit	External ¶		NO¶
L	process	External ¶		NO¶
DividendDistrib utor	Implementation	IDividendD istributor		
L		Public ¶		NO¶
L	setDistributionCriteria	External ¶		onlyToken
L	setShare	External ¶		onlyToken
L	deposit	External ¶		onlyToken
L	process	External ¶		onlyToken

L	shouldDistribute	Internal 		
L	distributeDividend	Internal 		
L	claimDividend	External 		NO 
L	getUnpaidEarnings	Public 		NO 
L	getCumulativeDividends	Internal 		
L	addShareholder	Internal 		
L	removeShareholder	Internal 		
BaseToken	Implementation			
AntiBotBuybackBabyToken	Implementation	IERC20Extended, Auth, BaseToken		
L		Public 		Auth
L	_initializeFees	Internal 		
L	_initializeLiquidityBuyBack	Internal 		
L	setEnableAntiBot	External 		authorized
L		External 		NO 
L	totalSupply	External 		NO 
L	decimals	External 		NO 
L	symbol	External 		NO 
L	name	External 		NO 

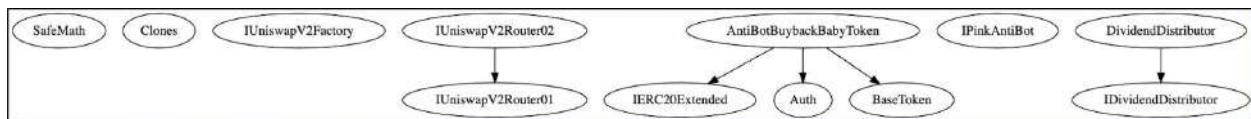
L	balanceOf	Public 🔒		NO 🔒
L	allowance	External 🔒		NO 🔒
L	approve	Public 🔒	🔒	NO 🔒
L	approveMax	External 🔒	🔒	NO 🔒
L	transfer	External 🔒	🔒	NO 🔒
L	transferFrom	External 🔒	🔒	NO 🔒
L	_transferFrom	Internal 🔒	🔒	
L	_basicTransfer	Internal 🔒	🔒	
L	shouldTakeFee	Internal 🔒		
L	getTotalFee	Public 🔒		NO 🔒
L	getMultipliedFee	Public 🔒		NO 🔒
L	takeFee	Internal 🔒	🔒	
L	shouldSwapBack	Internal 🔒		
L	swapBack	Internal 🔒	🔒	swapping
L	shouldAutoBuyback	Internal 🔒		
L	triggerZeusBuyback	External 🔒	🔒	authorized
L	clearBuybackMultiplier	External 🔒	🔒	authorized
L	triggerAutoBuyback	Internal 🔒	🔒	
L	buyTokens	Internal 🔒	🔒	swapping
L	setAutoBuybackSettings	External 🔒	🔒	authorized

L	setBuybackMultiplierSettings	External 		authorized
L	setIsDividendExempt	External 		authorized
L	setIsFeeExempt	External 		authorized
L	setBuyBacker	External 		authorized
L	setFees	Public 		authorized
L	_setFees	Internal 		
L	setFeeReceivers	External 		authorized
L	setSwapBackSettings	External 		authorized
L	setTargetLiquidity	External 		authorized
L	setDistributionCriteria	External 		authorized
L	setDistributorSettings	External 		authorized
L	getCirculatingSupply	Public 		NO 
L	getLiquidityBacking	Public 		NO 
L	isOverLiquified	Public 		NO 

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**

No high severity issues found

- **Medium severity issues**

No medium severity issues found

- **Low severity issues**

- ❖ Cannot exclude wallets from buy fee

```
ftrace | funcSig  
function shouldTakeFee(address sender↑) internal view returns (bool) {  
    return !isFeeExempt[sender↑];  
}
```

Owner privileges

- ❖ The owner can manually trigger buyback.

```
ftrace | funcSig
function triggerZeusBuyback(uint256 amount↑, bool triggerBuybackMultiplier↑)
    external
    authorized
{
    buyTokens(amount↑, DEAD);
    if (triggerBuybackMultiplier↑) {
        buybackMultiplierTriggeredAt = block.timestamp;
        emit BuybackMultiplierActive(buybackMultiplierLength);
    }
}
```

- ❖ The owner can clear buy back multiplier.

```
ftrace | funcSig
function clearBuybackMultiplier() external authorized {
    buybackMultiplierTriggeredAt = 0;
}
```

- ❖ The owner can exclude wallets from rewards.

```
ftrace | funcSig
function setIsDividendExempt(address holder↑, bool exempt↑)
    external
    authorized
{
    require(holder↑ != address(this) && holder↑ != pair);
    isDividendExempt[holder↑] = exempt↑;
    if (exempt↑) {
        distributor.setShare(holder↑, 0);
    } else {
        distributor.setShare(holder↑, balances[holder↑]);
    }
}
```

- ❖ The owner can include/exclude wallets from fees.

```
ftrace | funcSig
function setIsFeeExempt(address holder↑, bool exempt↑) external authorized {
    isFeeExempt[holder↑] = exempt↑;
}
```

- ❖ The owner can change all fees maximum up to 25%.

```
ftrace | funcSig
function setFees(
    uint256 _liquidityFee↑,
    uint256 _buybackFee↑,
    uint256 _reflectionFee↑,
    uint256 _marketingFee↑,
    uint256 _feeDenominator↑
) public authorized {
    _setFees(
        _liquidityFee↑,
        _buybackFee↑,
        _reflectionFee↑,
        _marketingFee↑,
        _feeDenominator↑
    );
}
```

- ❖ The owner can change liquidity and marketing fee receiver.

```
ftrace | funcSig
function setFeeReceivers(
    address _autoLiquidityReceiver↑,
    address _marketingFeeReceiver↑
) external authorized {
    autoLiquidityReceiver = _autoLiquidityReceiver↑;
    marketingFeeReceiver = _marketingFeeReceiver↑;
}
```

- ❖ The owner can enable/disable swap back and can change swap point.

```
ftrace | funcSig
function setSwapBackSettings(bool _enabled↑, uint256 _amount↑)
{
    external
    authorized
{
    swapEnabled = _enabled↑;
    swapThreshold = _amount↑;
}
```

- ❖ The owner can change the minimum reward period and minimum amount.

```
ftrace | funcSig
function setDistributionCriteria(
    uint256 _minPeriod↑,
    uint256 _minDistribution↑
) external authorized {
    distributor.setDistributionCriteria(_minPeriod↑, _minDistribution↑);
}
ftrace | funcSig
```


Audit conclusion

While conducting the audit of the Meta Shiba Inu smart contract, it was observed that there is nothing alarming with the code and it only contains a low severity issue.