RugFreeCoins Audit

Chisai Doge Token Audit

Smart Contract Security Audit

August 07, 2021

Contents

Audit details

Audited project

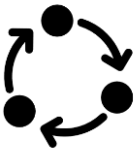
Chisai Doge Token

Contract Address

0x3c25fe68ba725c0dde20cb7ab9fa6ec7b3ed83cb

Client contact

Chisai Doge Team



Blockchain

Binance smart chain

Project website

https://chisaidoge.com/

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 Chisai Doge to perform an audit of the smart contract.

https://bscscan.com/token/0x3c25fe68ba725c0dde20cb7ab9fa6ec7b3ed83cb

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 improve the security posture of the smart contract by remediating the issues that were identified.

About the project

Chisai DogeCoin is a BEP20 token launched on the Binance Blockchain July 19th, 2021 to be a meme project for the community by the community. Chisai's mission is to give back by donating to those in need through reputable charity organizations across the world to make the world a better place through humanitarianism. Each transaction incurs a 10% fee when buying and selling.

Features:

- ❖ The **automatic token reward of 4%** will be distributed among every holder proportional to how many tokens each individual holds.

- ❖ The burn fee of 2% per transaction will be sent to the burn wallet, where the total supply in circulation will be reduced by keeping the market price stable.

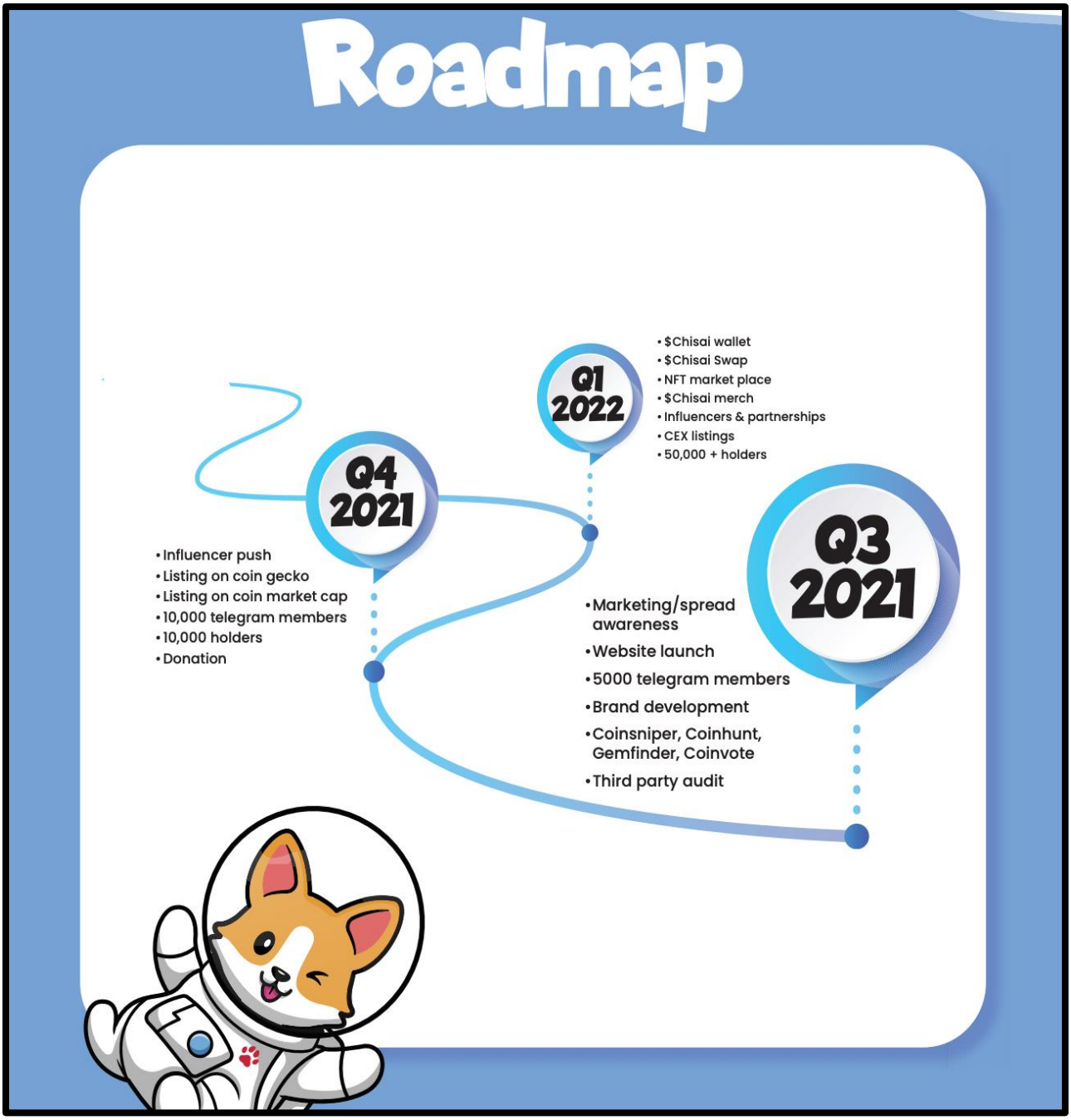
- ❖ The **sustainability fee of 4% marketing and charity** is what allows Chisai Doge to hold the aforementioned promise. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, Chisai Doge will have enough funds for charity and to promote the coin and spend for future development without selling tokens as the traditional way.

Tokenomics

10% tax fee when buying and selling

- 4% of every trade goes to holders pockets in tokens.
- 4% of every trade goes to the marketing wallet.
- 2% of every trade goes to the burn wallet.

Roadmap


Roadmap
Q4 2021
• Influencer push
• Listing on coin gecko
• Listing on coin market cap
• 10,000 telegram members
• 10,000 holders
• Donation
Q1 2022
• $Chisai wallet
• $Chisai Swap
• NFT market place
• $Chisai merch
• Influencers & partnerships
• CEX listings
• 50,000 + holders
Q3 2021
• Marketing/spread awareness
• Website launch
• 5000 telegram members
• Brand development
• Coinsniper, Coinhunt, Gemfinder, Coinvote
• Third party audit

Target market and the concept

Target market

- Anyone who’s interested in Crypto space with long term investment plans.
- Anyone who’s ready to earn a passive income by holding tokens.
- Anyone who’s interested in trading tokens.
- Anyone who's interested in supporting a good cause. (Charity)
- Anyone who’s interested in taking part with Chisai Doge future plans.
- Anyone who’s interested in making financial transactions with any other party using **Chisai Doge** as the currency.

Core concept

The reward system

4% of each transaction when buying and selling gets sent amongst all holders in tokens. The holders will be eligible to receive tokens, whenever a transaction occurs, and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

2% of each transaction when buying and selling is getting burnt, where the total supply in circulation is getting reduced with the time, which is able to keep the market price stable.

The fee of 4% marketing and charity is what allows Chisai Doge to use them for charity and to promote the token and use funds to further development of the platform. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, Daddy will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

Anti-whale strategy

The Chisai Doge contract includes a function that stops anyone holding more than 10 trillion tokens.This keeps investors safe from large dumps.

Future Plan

CHISAI SWAP

Chisai SWAP will be a decentralized exchange where users can swap any BEP20 token for another. Chisai Team also plans to add further functionality to ensure the platform is unique in the marketplace.

CHISAI DOGE WALLET

To assist the development of Chisai SWAP, the team is planning to develop a Chisai Wallet which will further provide holders advanced security features to ensure their coins are safe allowing users to spend, receive and trade with peace of mind.

CHISAI APPAREL

High quality merchandise to differentiate from other meme coins which holders can pay using cryptocurrency. The proceeds support further Chisai DogeCoin development and charities.

Potential to grow with score points

1.	Project efficiency	7/10
2.	Project uniqueness	6/10
3	Information quality	6/10
4	Service quality	7/10
5	System quality	7/10
6	Impact on the community	7/10
7	Impact on the business	7/10
8	Preparing for the future	6/10
Total Points		**6.63/10**

Contract details

Token contract details for 07th August 2021

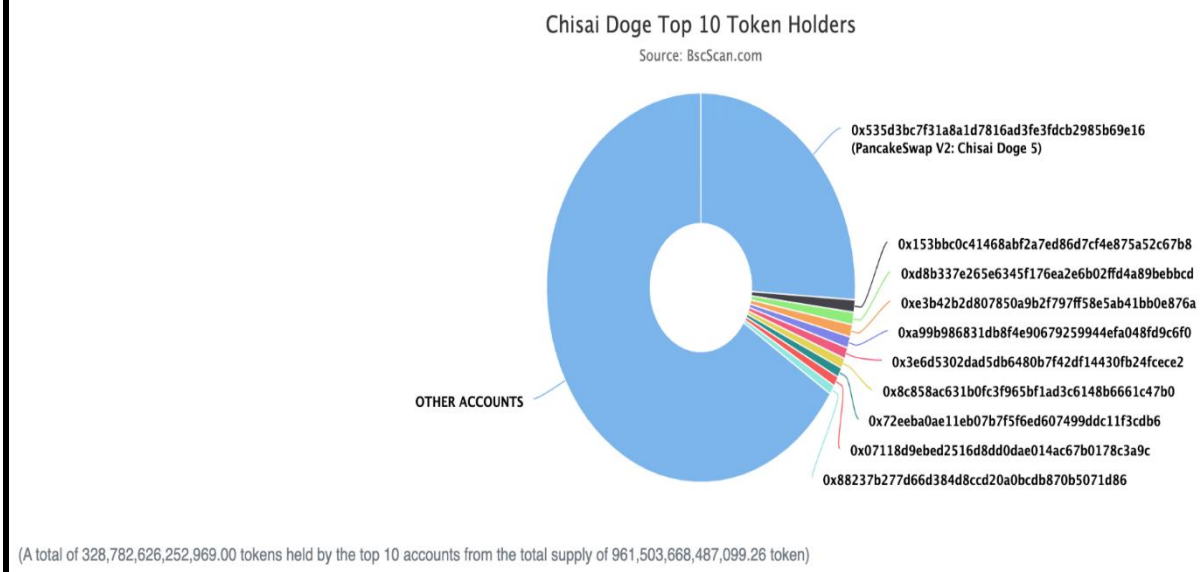
Contract name	Chisai Doge Token
Contract address	0x3c25fe68ba725c0dde20cb7ab9fa6ec7b3ed83cb
Token supply	961,539,483,378,112.27
Token ticker	Chisai Doge
Decimals	9
Token holders	486
Transaction count	2,679
Top 100% holders dominance	75.48%
Marketing address	0x5b40fd0ca37d592bc7e197ca07b12e0c711ff429
Contract deployer address	0x852E13a2fb682A96130FFceD444d31262d5C7b87
Contract's current owner address	0x0000000000000000000000000000000000000000

Top token holders

Top 10 Token Holders

The top 10 holders collectively own 34.19% (328,782,626,252,969.00 Tokens) of Chisai Doge

Token Total Supply: 961,503,668,487,099.26 Token | Total Token Holders: 490



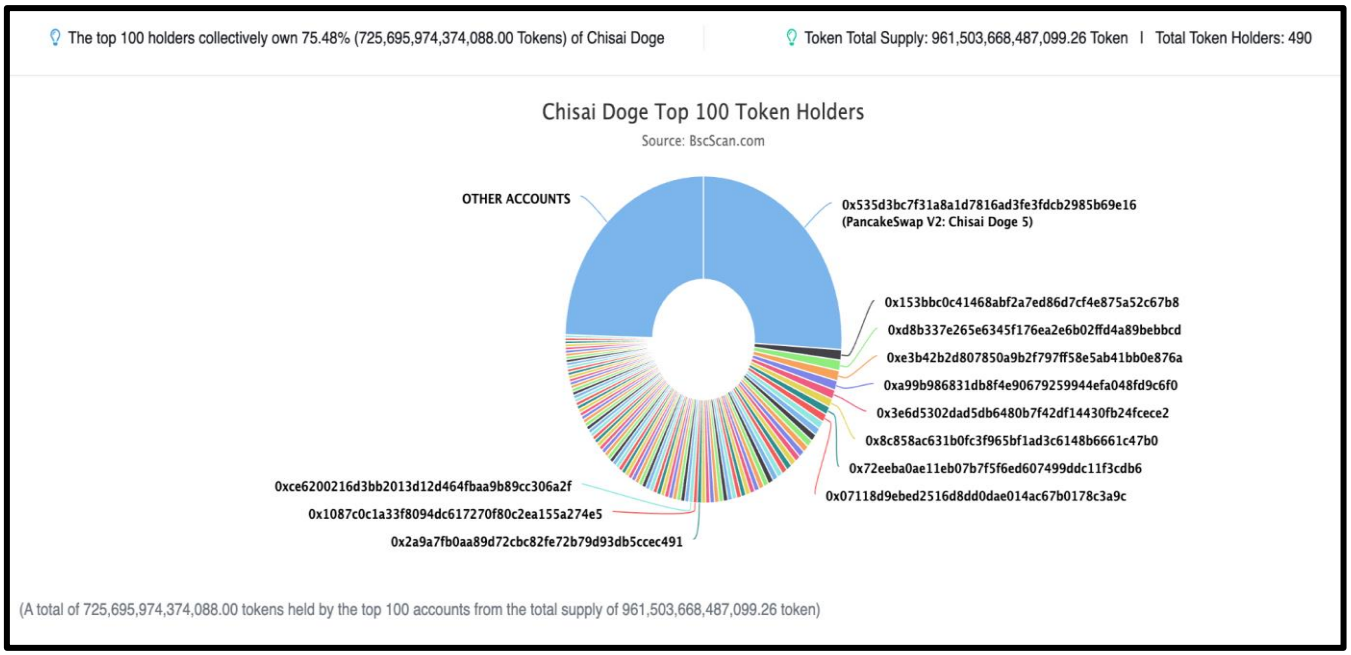
(A total of 328,782,626,252,969.00 tokens held by the top 10 accounts from the total supply of 961,503,668,487,099.26 token)

Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: Chisai Doge 5	249,988,016,947,682.717990466	25.9997%
2	0x153bbc0c41468abf2a7ed86d7cf4e875a52c67b8	9,999,294,966,300.635640144	1.0400%
3	0xd8b337e265e6345f176ea2e6b02ffd4a89bebbcd	9,993,132,354,421.546288762	1.0393%
4	0xe3b42b2d807850a9b2f797ff58e5ab41bb0e876a	9,786,020,282,080.476044886	1.0178%
5	0xa99b986831db8f4e90679259944efa048fd9c6f0	9,000,022,256,987.036963325	0.9360%
6	0x3e6d5302dad5db6480b7f42df14430fb24fcece2	8,768,503,497,240.247736087	0.9120%
7	0x8c858ac631b0fc3f965bf1ad3c6148b6661c47b0	8,200,438,442,172.970072828	0.8529%
8	0x72eeba0ae11eb07b7f5f6ed607499ddc11f3cdb6	7,822,724,415,906.610956098	0.8136%
9	0x07118d9ebed2516d8dd0dae014ac67b0178c3a9c	7,732,936,548,698.93006873	0.8043%
10	0x88237b277d66d384d8ccd20a0bcdb870b5071d86	7,491,536,541,477.323867086	0.7791%

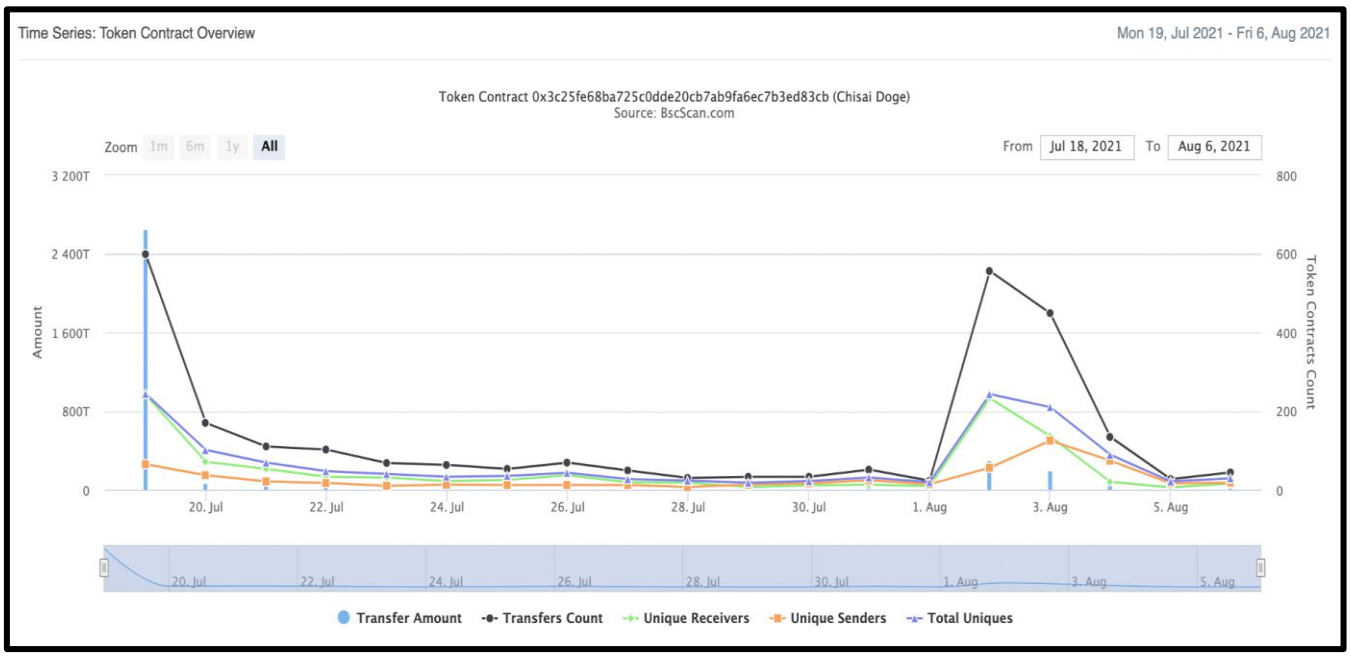
Token distribution

Tokens are distributed as follows:

Top 100 Token Holders



Contract interaction details



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

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 its visibility and mutability.

Contract	**Type**	**Bases**		
└	Function Name	Visibility	Mutability	Modifiers
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
IERC20	**Interface**			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
SafeMath	**Library**			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		

└	mod	Internal 🔒		
Address	**Library**			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
Ownable	**Implementation**	**Context**		
└		Internal 🔒	🛑	
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	getUnlockTime	Public ❗		NO❗
└	getTime	Public ❗		NO❗
IUniswapV2Factory	**Interface**			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗

└	createPair	External ❗	🔘	NO❗
└	setFeeTo	External ❗	🔘	NO❗
└	setFeeToSetter	External ❗	🔘	NO❗
IUniswapV2Pair	**Interface**			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🔘	NO❗
└	transfer	External ❗	🔘	NO❗
└	transferFrom	External ❗	🔘	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🔘	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗

└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	burn	External ❗	🔘	NO❗
└	swap	External ❗	🔘	NO❗
└	skim	External ❗	🔘	NO❗
└	sync	External ❗	🔘	NO❗
└	initialize	External ❗	🔘	NO❗
IUniswapV2Router01	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidity	External ❗	🔘	NO❗
└	addLiquidityETH	External ❗	💸	NO❗
└	removeLiquidity	External ❗	🔘	NO❗
└	removeLiquidityETH	External ❗	🔘	NO❗
└	removeLiquidityWithPermit	External ❗	🔘	NO❗
└	removeLiquidityETHWithPermit	External ❗	🔘	NO❗
└	swapExactTokensForTokens	External ❗	🔘	NO❗
└	swapTokensForExactTokens	External ❗	🔘	NO❗
└	swapExactETHForTokens	External ❗	💸	NO❗
└	swapTokensForExactETH	External ❗	🔘	NO❗
└	swapExactTokensForETH	External ❗	🔘	NO❗

└	swapETHForExactTokens	External ❗	💵	NO❗
└	quote	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗
IUniswapV2Router02	**Interface**	**IUniswapV2Router01**		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔘	NO❗
ChisaiDoge	**Implementation**	**Context, IERC20, Ownable**		
└		Public ❗	🔘	NO❗
└	setWalletlimit	External ❗	🔘	onlyOwner
└	name	Public ❗		NO❗

└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🔘	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🔘	NO❗
└	transferFrom	Public ❗	🔘	NO❗
└	increaseAllowance	Public ❗	🔘	NO❗
└	decreaseAllowance	Public ❗	🔘	NO❗
└	isExcludedFromReward	Public ❗		NO❗
└	totalFees	Public ❗		NO❗
└	totalBurn	Public ❗		NO❗
└	minimumTokensBeforeSwapAmount	Public ❗		NO❗
└	deliver	Public ❗	🔘	NO❗
└	reflectionFromToken	Public ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	excludeFromReward	Public ❗	🔘	onlyOwner
└	includeInReward	External ❗	🔘	onlyOwner
└	_approve	Private 🔐	🔘	
└	_transfer	Private 🔐	🔘	
└	swapAndLiquify	Private 🔐	🔘	lockTheSwap
└	swapTokensForEth	Private 🔐	🔘	

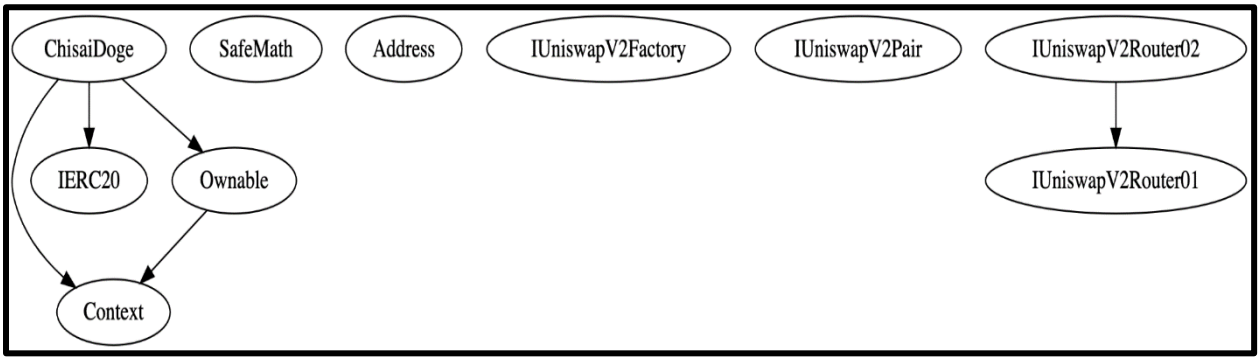
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeLiquidity	Private 🔐	🛑	
└	calculateTaxFee	Private 🔐		
└	calculateBurnFee	Private 🔐		
└	calculateLiquidityFee	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	restoreAllFee	Private 🔐	🛑	
└	isExcludedFromFee	Public ❗		NO❗
└	excludeFromFee	Public ❗	🛑	onlyOwner
└	includeInFee	Public ❗	🛑	onlyOwner
└	setTaxFeePercent	External ❗	🛑	onlyOwner
└	setBurnFeePercent	External ❗	🛑	onlyOwner
└	setDonationFeePercent	External ❗	🛑	onlyOwner

L	setMaxTxPercent	External		onlyOwner
L	setNumTokensSellToAddToLiquidity	External		onlyOwner
L	setSwapAndLiquifyEnabled	Public		onlyOwner
L	TransferMarketingETH	Private		
L		External		NO

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

Inheritance Hierarchy



Security issue checking status

- **High severity issues**
  No medium severity issues found.

- **Medium severity issues**
  No medium severity issues found.

- **Low severity issues**
  No low severity issues found.

Owner privileges

Ownership has been renounced and the owner doesn't have any privileges and has no authority to make any changes now.

Ownership privileges owner had

- The owner can transfer and renounce the ownership.

```
ftrace | funcSig
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
}

ftrace | funcSig
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

- The owner can change the max wallet limit.

```
ftrace | funcSig
function setWalletlimit(uint256 _walletlimit) external onlyOwner() {
    walletlimit = _walletlimit;
}
```

- The owner can include/exclude wallets from rewards.

```
ftrace | funcSig
function excludeFromReward(address account) public onlyOwner() {
    // require(account != 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D, 'We can not exclude Uniswap router.');
    require(!_isExcluded[account], "Account is already excluded");
    if(_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

ftrace | funcSig
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;
        }
    }
}
```

❖ The owner can include/exclude wallets from fees.

```
ftrace | funcSig
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

ftrace | funcSig
function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

❖ The owner can change the tax fee, burn Fee and marketing fee.

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}

ftrace | funcSig
function setBurnFeePercent(uint256 burnFee) external onlyOwner() {
    _burnFee = burnFee;
}

ftrace | funcSig
function setDonationFeePercent(uint256 DonationFee) external onlyOwner() {
    _donationFee = DonationFee;
}
```

❖ The owner can change the maximum transaction amount.

```
ftrace | funcSig
function setMaxTxPercent(uint256 maxTxPercent, uint256 maxTxDecimals) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**(uint256(maxTxDecimals) + 2)
    );
}
```

Audit conclusion

While conducting the audit of the Chisai Doge Token smart contract, it was observed that there is nothing alarming with the code.