



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Doge 98
\$DOGE98

17/10/2023



TOKEN OVERVIEW

Fees

- Buy fees: 1%
- Sell fees: 1%

Fees privileges

- Can change buy fees up to 70% and sell fees up to 75%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change max tx amount and max wallet amount (with threshold)

Blacklist

- Blacklist function detected

Other privileges

- Can exclude / include from fees
 - Contract owner has to call enableTrading function to enable trade
-

TABLE OF CONTENTS

1

DISCLAIMER

2

INTRODUCTION

3

WEBSITE + SOCIALS

4-5

AUDIT OVERVIEW

6-9

OWNER PRIVILEGES

10

CONCLUSION AND ANALYSIS

11

TOKEN DETAILS

12

DOGE98 ANALYTICS &
TOP 10 TOKEN HOLDERS

13

TECHNICAL DISCLAIMER



DISCLAIMER

The information provided on this analysis document is only for general information and should not be used as a reason to invest.

FreshCoins Team will take no payment for manipulating the results of this audit.

The score and the result will stay on this project page information on our website <https://freshcoins.io>

FreshCoins Team does not guarantees that a project will not sell off team supply, or any other scam strategy (RUG or Honeypot etc)



INTRODUCTION

FreshCoins (Consultant) was contracted by **Doge 98** (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xb91dDc423B73e1E7E3E04cC64bBa7D7E3b2b8071

Network: **Ethereum (ETH)**

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on **17/10/2023**



WEBSITE DIAGNOSTIC

<https://doge98.xyz/>



0-49



50-89



90-100



Performance



Accessibility



Best
Practices



SEO



Progressive
Web App

Socials



Twitter

<https://twitter.com/Doge98erc>



Telegram

<https://t.me/doge98erc>

AUDIT OVERVIEW



Security Score
HIGH RISK
Audit FAIL



Static Scan
Automatic scanning for
common vulnerabilities



ERC Scan
Automatic checks for
ERC's conformance



High



Medium



Low



Optimizations



Informational



No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passed
2	Reentrancy and Cross-function	Passed
3	Front running	Low
4	Timestamp dependence	Passed
5	Integer Overflow and Underflow	Passed
6	Reverted DoS	Passed
7	DoS with block gas limit	Low
8	Methods execution permissions	Passed
9	Exchange rate impact	Passed
10	Malicious Event	Passed
11	Scoping and Declarations	Passed
12	Uninitialized storage pointers	Passed
13	Design Logic	Passed
14	Safe Zeppelin module	Passed

OWNER PRIVILEGES

● Contract owner can't mint tokens after initial contract deploy

● Contract owner can exclude an address from transactions

```
function blacklistAccount(address account, bool isBlacklisted) public onlyOwner {
    _blacklist[account] = isBlacklisted;
}
```

● Contract owner can exclude/include wallet from tax

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    _isExcludedFromFees[account] = excluded;
    emit ExcludeFromFees(account, excluded);
}
```

● Contract owner can exclude/include wallet from max tx limitations

```
function excludeFromMaxTransaction(address updAds, bool isEx) public onlyOwner {
    _isExcludedMaxTransactionAmount[updAds] = isEx;
}
```

● Contract owner can disable transfer delay between transactions

```
function disableTransferDelay() external onlyOwner returns (bool){
    transferDelayEnabled = false;
    return true;
}

_transfer function line 1155

if (transferDelayEnabled){
    if (to != owner() && to != address(uniswapV2Router) && to != address(uniswapV2Pair)){
        require(_holderLastTransferTimestamp[tx.origin] < block.number, "_transfer:: Transfer Delay
enabled. Only one purchase per block allowed.");
        _holderLastTransferTimestamp[tx.origin] = block.number;
    }
}
```

● Contract owner can disable tx limitations

```
function removeLimits() external onlyOwner returns (bool){
    limitsInEffect = false;
    return true;
}
```

● Contract owner has to call `enableTrading` function to enable trade

Please be aware that wallets exempted from fees retain the ability to engage in trading activities even when trading is disabled

```
function enableTrading() external onlyOwner {
    tradingActive = true;
    swapEnabled = true;
    lastLpBurnTime = block.timestamp;
    launchedAt = block.number;
}

_transfer function line 1150

if(!tradingActive){
    require(!_isExcludedFromFees[from] || !_isExcludedFromFees[to], "Trading is not active.");
}
```

● Contract owner can change swap settings

```
function updateSwapEnabled(bool enabled) external onlyOwner(){
    swapEnabled = enabled;
}

function updateSwapTokensAtAmount(uint256 newAmount) external onlyOwner returns (bool){
    require(newAmount >= totalSupply() * 1 / 100000, "Swap amount cannot be lower than 0.001% total supply.");
    require(newAmount <= totalSupply() * 5 / 1000, "Swap amount cannot be higher than 0.5% total supply.");
    swapTokensAtAmount = newAmount;
    return true;
}
```

● Contract owner can change `marketingWallet` and `devWallet` addresses

Current values:

marketingWallet: `0xE09ffc4a9fC6aD1fD94EEdf3bCf7c5330B3327Ce`

devWallet: `0xE09ffc4a9fC6aD1fD94EEdf3bCf7c5330B3327Ce`

```
function updateMarketingWallet(address newMarketingWallet) external onlyOwner {
    emit marketingWalletUpdated(newMarketingWallet, marketingWallet);
    marketingWallet = newMarketingWallet;
}

function updateDevWallet(address newWallet) external onlyOwner {
    emit devWalletUpdated(newWallet, devWallet);
    devWallet = newWallet;
}
```

● Contract owner can change buy fees up to 70% and sell fees up to 75%

```
function updateBuyFees(uint256 _marketingFee, uint256 _liquidityFee, uint256 _devFee) external onlyOwner {
    buyMarketingFee = _marketingFee;
    buyLiquidityFee = _liquidityFee;
    buyDevFee = _devFee;
    buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
    require(buyTotalFees <= 70, "Must keep fees at 70% or less");
}

function updateSellFees(uint256 _marketingFee, uint256 _liquidityFee, uint256 _devFee, uint256 _earlySellLiquidityFee, uint256 _earlySellMarketingFee) external onlyOwner {
    sellMarketingFee = _marketingFee;
    sellLiquidityFee = _liquidityFee;
    sellDevFee = _devFee;
    earlySellLiquidityFee = _earlySellLiquidityFee;
    earlySellMarketingFee = _earlySellMarketingFee;
    sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
    require(sellTotalFees <= 75, "Must keep fees at 75% or less");
}
```

● Contract owner can change max tx amount limitation (with threshold)

```
function updateMaxTxnAmount(uint256 newNum) external onlyOwner {
    require(newNum >= (totalSupply() * 15 / 1000) / 1e18, "Cannot set maxTransactionAmount lower than 0.5%");
    maxTransactionAmount = newNum * (10**18);
}
```

● Contract owner can change max wallet amount limitation (with threshold)

```
function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
    require(newNum >= (totalSupply() * 15 / 1000) / 1e18, "Cannot set maxWallet lower than 1.5%");
    maxWallet = newNum * (10**18);
}
```

● Contract owner can transfer ownership

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

● Contract owner can renounce ownership

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

Recommendation:

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. The risk can be prevented by temporarily locking the contract or renouncing ownership.



CONCLUSION AND ANALYSIS



Smart Contracts within the scope were manually reviewed and analyzed with static tools.



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



Found 3 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	1%
Sell fees:	1%
Max TX:	20,000
Max Sell:	N/A

Honeypot Risk

Ownership:	Owned
Blacklist:	Detected
Modify Max TX:	Detected
Modify Max Sell:	Not detected
Disable Trading:	Not detected

Rug Pull Risk

Liquidity:	N/A
Holders:	11.8% unlocked tokens



DOGE98 TOKEN ANALYTICS

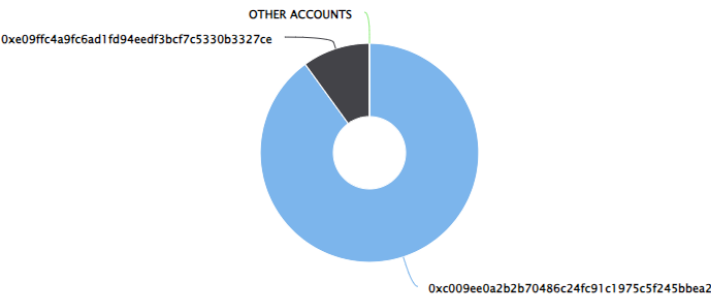
& TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (1,000,000.00 Tokens) of Doge 98

Token Total Supply: 1,000,000.00 Token | Total Token Holders: 2

Doge 98 Top 10 Token Holders

Source: Etherscan.io



(A total of 1,000,000.00 tokens held by the top 10 accounts from the total supply of 1,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xc009Ee...245bbea2	899,999.7234	90.0000%
2	0xE09ffc...0B3327Ce	100,000.2766	10.0000%

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 its vulnerabilities that can lead to hacks. The audit can't guarantee the explicit security of the audited project / smart contract.

