freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

DODDA
THE FIRST OF ITS KIND

Dodda.io
$DDAT

22/02/2023

TOKEN OVERVIEW

Fees

- Buy fees: 0%
- Sell fees: 0%

Fees privileges

- Can change fees up to 25%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can’t change max tx amount or wallet amount

Blacklist

- No blacklist function

Other privileges

- Can exclude / include wallet from tax

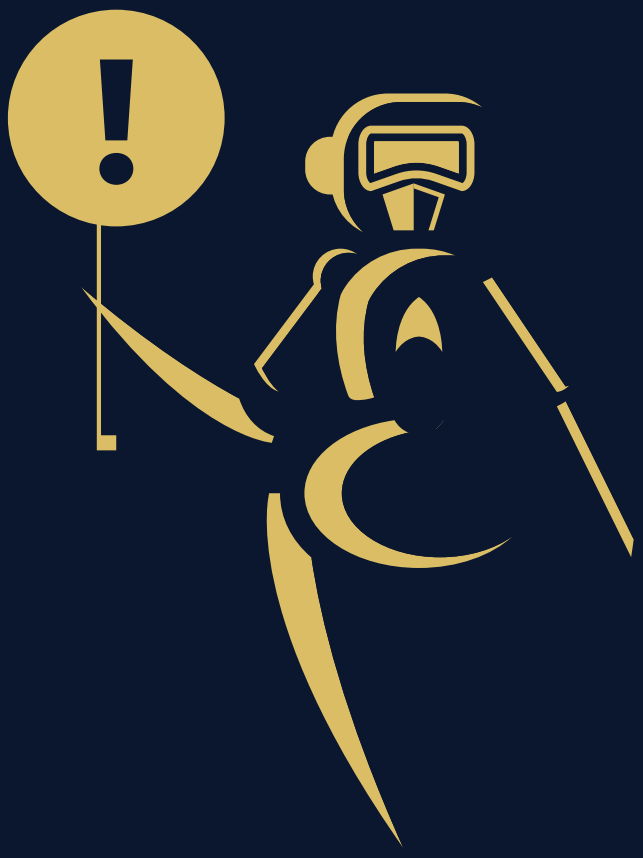
TABLE OF CONTENTS

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 Dodda.io (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x3d0c050fAb51b43D53fcc89AFb91ED268E0A6591

Network: Arbitrum One

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 22/02/2023

WEBSITE DIAGNOSTIC

https://dodda.io/

 0-49

 50-89

 90-100

Performance

Accessibility

Best Practices

SEO

Progressive Web App

Socials

 Twitter

N/A

 Telegram

https://t.me/doddachat

AUDIT OVERVIEW

92

Security Score

98

Static Scan
Automatic scanning for common vulnerabilities

86

ERC Scan
Automatic checks for ERC's conformance

0 High

1 Medium

0 Low

0 Optimizations

0 Informational

No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passed
2	Reentrancy and Cross-function	Passed
3	Front running	Passed
4	Timestamp dependence	Passed
5	Integer Overflow and Underflow	Passed
6	Reverted DoS	Passed
7	DoS with block gas limit	Passed
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't exclude an address from transactions.
- Contract owner can exclude/include wallet from fees

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

- Contract owner can change marketingWallet address

Current value:

marketingWallet : 0x8408d69bd878542c79d6fefa5b4dc88c2aca1e1c

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

- Contract owner can change fees up to 25%

```
function updateFees(uint256 _marketingFee, uint256 _liquidityFee) external onlyOwner {
    MarketingFee = _marketingFee;
    LiquidityFee = _liquidityFee;
    TotalFees = MarketingFee + LiquidityFee;
    require(TotalFees <= 250, "Must keep fees at 25% or less");
}
```

- Contract owner can change swap settings

```
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;
}

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

- 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 no HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	0%
Sell fees:	0%
Max TX:	N/A
Max Sell:	N/A

Honeypot Risk

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

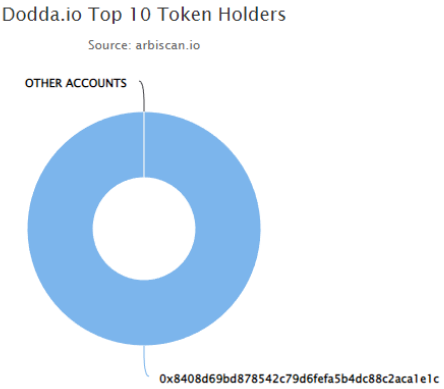
Others

Liquidity:	N/A
Holders:	Clean

DDAT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (100,000,000.00 Tokens) of Dodda.io

Token Total Supply: 100,000,000.00 Token | Total Token Holders: 1



(A total of 100,000,000.00 tokens held by the top 10 accounts from the total supply of 100,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x8408d69bd878542c79d6fefa5b4dc88c2aca1e1c	100,000,000	100.0000%

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.