# DOGEFATHER COIN

Token Audit Report

# CONTENTS

## OVERVIEW

This document is a security audit of the **DOGEFATHER COIN** token included with this delivery.

Following files / results are included with this delivery:

1. Original Contract Code
2. Report
3. Contract Overview diagram
4. Custom Test Result

## GLOSSARY

The automated tests are performed against a knowledgebase of commonly known issues and assigned a SEVERITY as per the security issue.

Severity is categorized into three levels starting from 1 up to 3. Higher the number, higher is the threat.

- Severity 1 - Low Severity
- Severity 2 - Medium Severity
- Severity 3 - High Severity

Apart from security issues, notes might also be added to point out a certain functionality.

## CONTRACT OVERVIEW

**Note:** Contract Diagram has been included with this delivery for tracking contract inheritance, functionality calls

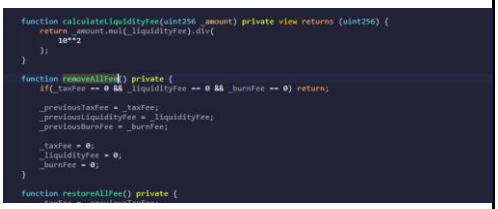
This document details **DOGEFATHER COIN** token findings and recommended solutions. This audit was performed on Aug 04, 2021. We audited a deployed token.

| | |
|---|---|
| **Token Name** | DOGEFATHER COIN |
| **Token Symbol** | DGF |
| **Token Decimal** | 18 |
| **Reward Fee** | 5% |
| **Burn Fee** | 10% |
| **Liqudity Fee** | 10% |
| **Contract Address** | 0x4Bc9846a2BB6fa6502f815F0603C65EF7eFE64Ab |
| **BSCScan Link** | https://bscscan.com/address/0x4Bc9846a2BB6fa6502f815F0603C65EF7eFE64Ab |

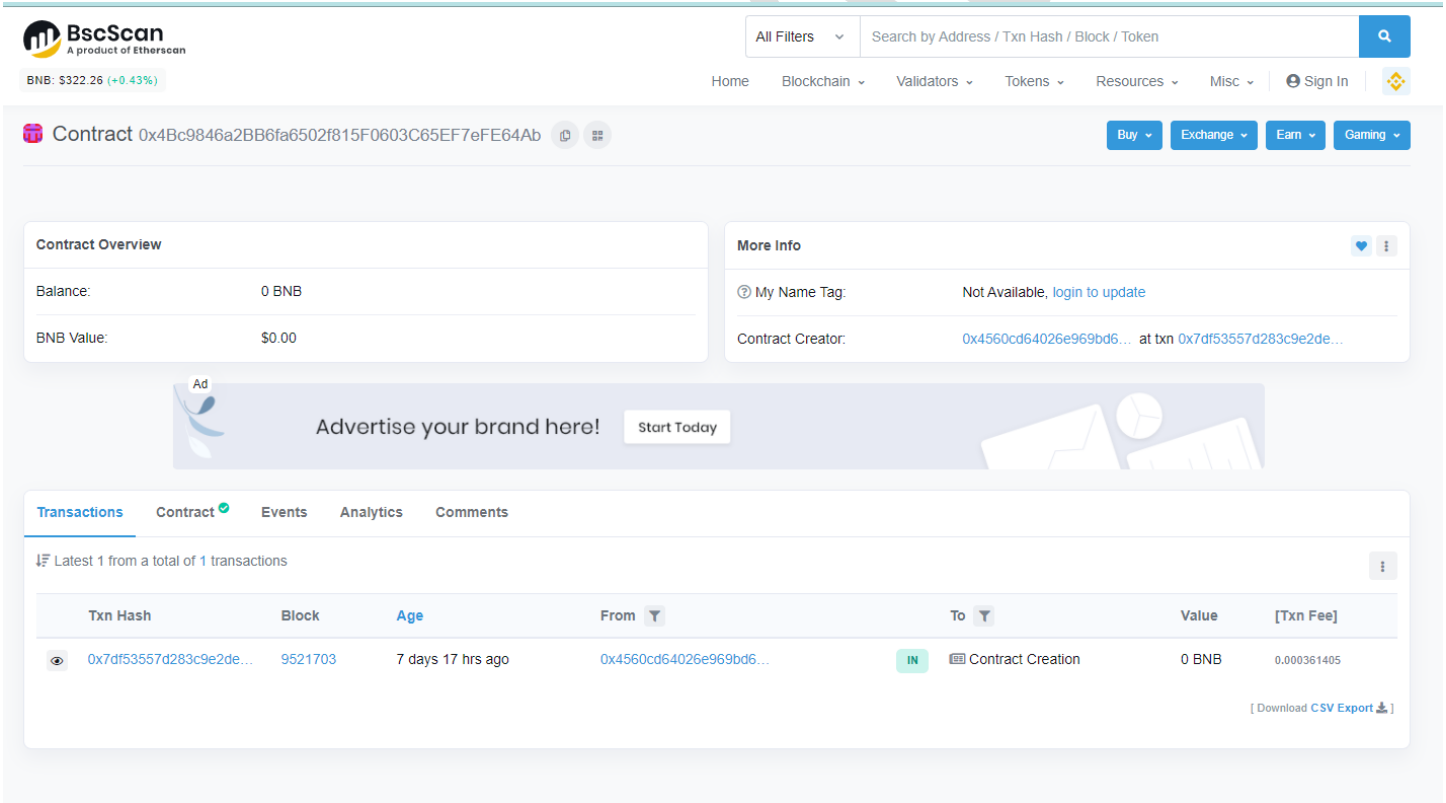
# AUDIT RESULT

| NO. | Audited Items | Function's role of Token | Audit Result |
|---|---|---|---|
| **1** | Overflow Audit |  | Passed |
| **2** | Safe Design Audit | Checking if the design of contract is safe | Passed |
| **3** | Gas Optimization Audit | Checking if All Gas is optimized | Passed |
| **4** | Design Logic Audit | Checking if the logic of full contract | Passed |
| **5** | Malicious Event Log Audit | Checking if Malicious event log will be displayed | Passed |
| **6** | Uninitialized Storage Pointers Audit | Checking if uninitialized storage exists | Passed |
| **7** | Arithmetic Accuracy Deviation Audit | Checking if the all calculating is success | Passed |
| **8** | Compiler errors | Checking if compile is success | Passed |
| **9** | Private user data leaks | Checking if user data is safe | Passed |
| **10** | Possibly delays in data delivery | Checking if there is delay when data delivery | Passed |
| **11** | Methods execution permissions | Checking if the permission is allowed to exact user | Passed |
| **12** | _transfer() | Sending token between two accounts | Passed |
| **13** | addLiquidity() | Addition Liqudity(It calls frequently in farming and pools) | Passed |
| **14** | setTaxFeePercent () | Setting Tax fee | Passed |
| **15** | setLiquidityFeePercent() | Setting Liqudity fee | Passed |
| **16** | setBurnFeePercent () | Setting Burn Fee | Passed |

| | | | |
|---|---|---|---|
| **17** | removeAllFee () | Remove All Fees | Passed |
| **18** | restoreAllFee () | Restore all Fees after removing fees | Passed |
| **19** | _getCurrentSupply () | Getting current Supply | Passed |







## High Severity Issues

- None.

## Medium Severity Issues

- Issue:

  Reward Fee as 5%, Burn Fee as 10%, Liqudity Fee as 10% were settee.

  So, it can be changeable any time.

  But people don't often focus on fee. They will see it at beginning time at once.

  People may not know if the liquidity fee is changed

- Comments:

  The All Fees should be static value. So, people will always be paid the same fee

## Low Severity Issues

- None.

## CONCLUSION

A Medium severity issue found. These issues are not so serious problems. If some fees are getting higher so much than beginning, people will not happy with it.

The all things that related with Fee should be transparent.