DOGEFATHER COIN

Token Audit Report

**TABLE OF CONTENTS**

[Overview 3](#_Toc78999139)

[Glossary 3](#_Toc78999140)

[Contract Overview 4](#_Toc78999141)

[Audit Result 5](#_Toc78999142)

[Severity Definitions 7](#_Toc78999143)

[Audit Findings 8](#_Toc78999144)

[Conclusion 8](#_Toc78999145)

# 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

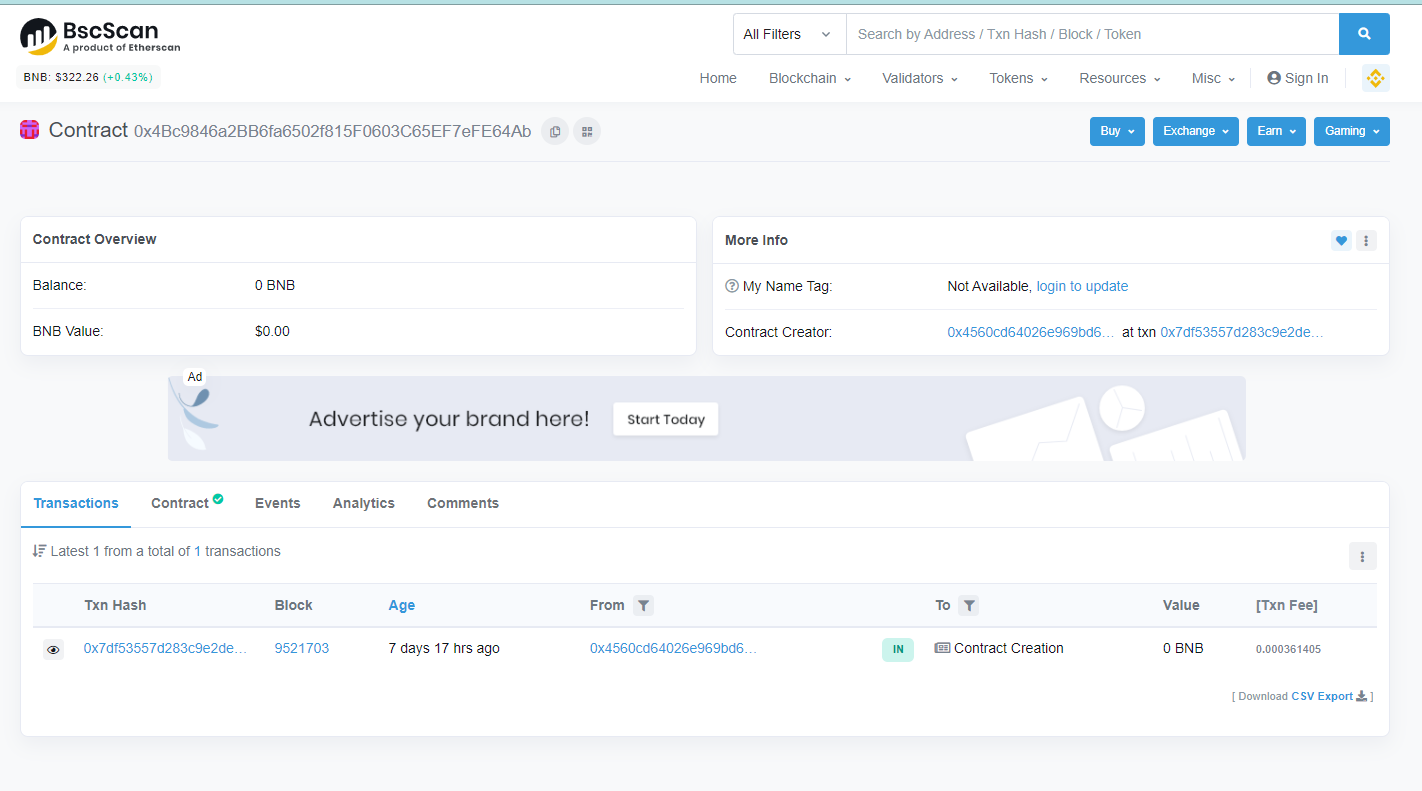
**Note:** Contract Diagram has been included with this delivery for tracking contract inheritance, functionality calls

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 |



# Severity Definitions

|  |  |
| --- | --- |
| **Risk Level** | **Description** |
| **Critical** | Critical vulnerabilities are usually straightforward to exploit and can lead to lost tokens etc. |
| **High** | High level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g. public access to crucial functions. |
| **Medium** | Medium level vulnerabilities are important to fix; however, they cannot lead to lost tokens. |
| **Low** | Low level vulnerabilities are most related to outdated, unused etc. These code snippets cannot have a significant impact on execution. |
| **Lowest Code Style/ Best Practice** | Lowest level vulnerabilities, code style violations and information statements cannot affect smart contract execution and can be ignored. |

# Audit Findings

* Critical

No Critical severity vulnerabilities were found.

* High

No high severity vulnerabilities were found.

* Medium

No medium severity vulnerabilities were found.

* Low

No low severity vulnerabilities were found.

* Very Low

No very low severity vulnerabilities were found

# Conclusion

This Token is exact in all part. All codes and functions are checked enough. As a result, all things are correct.

There are no severity vulnerabilities.