RUGDOG

# SMART CONTRACT SECURITY AUDIT

Final report

Plan: Simple

## bnbFROGE

August 2022

rugdog.net

the@rugdog.net

# ✧ CONTENTS

## ◊ INTRODUCTION

Implementation of ERC-20 token standard with fees on transfers. Fees are used to buy back reward token in uniswap pair.

| | |
|---|---|
| Name | bnbFROGE |
| Audit date | 2022-08-29 - 2022-08-29 |
| Language | Solidity |
| Network | Binance Smart Chain |

## ◊ CONTRACTS CHECKED

| Name | Address |
|---|---|
| DividendDistributor | 0x9db56e9f6a1f75f6558f215195a9ff780816321c |
| BuybackBabyToken | 0x9db56e9f6a1f75f6558f215195a9ff780816321c |
| Clones | 0x9db56e9f6a1f75f6558f215195a9ff780816321c |

## ◊ AUDIT PROCESS

The code was audited by the team according to the following order:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual confirmation of all the issues found by the tools

Manual audit

- Thorough manual analysis of smart contracts for security vulnerabilities
- Smart contracts' logic check

## ATTACKS CHECKED

| Title | Check result |
|---|---|
| Unencrypted Private Data On-Chain | ✓ passed |
| Code With No Effects | ✗ failed |
| Message call with hardcoded gas amount | ✓ passed |
| Typographical Error | ✓ passed |
| DoS With Block Gas Limit | ✓ passed |
| Presence of unused variables | ✗ failed |
| Incorrect Inheritance Order | ✓ passed |
| Requirement Violation | ✓ passed |
| Weak Sources of Randomness from Chain Attributes | ✓ passed |
| Shadowing State Variables | ✓ passed |
| Incorrect Constructor Name | ✓ passed |
| Block values as a proxy for time | ✓ passed |
| Authorization through tx.origin | ✓ passed |

| | |
|---|---|
| DoS with Failed Call | ✓ passed |
| Delegatecall to Untrusted Callee | ✓ passed |
| Use of Deprecated Solidity Functions | ✓ passed |
| Assert Violation | ✓ passed |
| State Variable Default Visibility | ✓ passed |
| Reentrancy | ✓ passed |
| Unprotected SELFDESTRUCT Instruction | ✓ passed |
| Unprotected Ether Withdrawal | ✓ passed |
| Unchecked Call Return Value | ✓ passed |
| Floating Pragma | ✓ passed |
| Outdated Compiler Version | ✓ passed |
| Integer Overflow and Underflow | ✓ passed |
| Function Default Visibility | ✓ passed |

## ✧ OVERVIEW OF RELEVANCE LEVELS

**High relevance** Issues of high relevance may lead to losses of users' funds as well as changes of ownership of a contract or possible issues with the logic of the contract.
High-relevance issues require immediate attention and a response from the team.

| | |
|---|---|
| **Medium relevance** | While issues of medium relevance don't pose as high a risk as the high-relevance ones do, they can be just as easily exploited by the team or a malicious user, causing a contract failure and damaging the project's reputation in the process. Usually, these issues can be fixed if the contract is redeployed.<br>Medium-relevance issues require a response from the team. |
| **Low relevance** | Issues of low relevance don't pose high risks since they can't cause damage to the functionality of the contract. However, it's still recommended to consider fixing them. |

# ISSUES

## High relevance issues

### 1. Excessive owner's rights (BuybackBabyToken)

a. The owner can exclude any address from dividends reception;

b. The owner can update the swapThreshold variable with a wrong value. This may halt the distribution of the fees for a long period of time. Enabling back swaps and liquidity adding may lead to the token price wrecking if the contract's balance is comparable to a pair reserves.

### 2. No error handling (BuybackBabyToken)

_transferFrom() and swapBack() functions contain empty try/catch blocks in case of reward token transfer failure. Therefore, users' shares may become unfair and inconsistent.

## Medium relevance issues

### 1. Swaps with 100% slippage (BuybackBabyToken)

swapExactTokensForETHSupportingFeeOnTransferTokens() and swapExactETHForTokensSupportingFeeOnTransferTokens() functions call router with 100% slippage. The transactions sent from this contract may be front-runned resulting in swaps with an undesired rate (sandwich attacks).

## Low relevance issues

### 1. Lack of events (DividendDistributor)

No events are emittied in setShare(), deposit(), distributeDividend(), setDistributionCriteria()

### 2. Gas optimisation (DividendDistributor)

a. rewardToken, router, _token should be marked immutable;

b. dividendsPerShareAccuracyFactor shoud be const;

### 3. Redundant code (BuybackBabyToken)

Setter function and access modifactor for buyBackers are not applyied anywhere.

### 4. Lack of events (BuybackBabyToken)

No events are emitted in setFeeReceivers(), setSwapBackSettings(), setTargetLiquidity(), setDistributorSettings(), setFees(), setIsFeeExempt().

### 5. Gas optimization (BuybackBabyToken)

_name, _symbol, _totalSupply, router, pair should be marked as immutable.

### 6. Redundunt code (Clones)

The library is not used in the contract.

## ✧ CONCLUSION

bnbFROGE DividendDistributor, BuybackBabyToken, Clones contracts were audited. 2 high, 1 medium, 6 low relevance issues were found.

# ✧ DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability)set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without RugDog prior written consent.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts RugDog to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

RUGDOG

# WOOF!

rugdog.net

the@rugdog.net