QuillAudits

Audit Report
May, 2023

For

BabyDoge

Table of Content

Executive Summary 01

Checked Vulnerabilities 03

Techniques and Methods 04

Manual Testing 05

High Severity Issues 05

Medium Severity Issues 05

Low Severity Issues 05

Informational Issues 05

Functional Tests 06

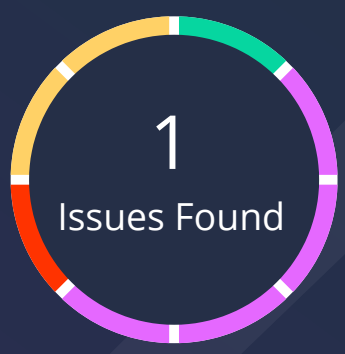
Automated Tests 06

Closing Summary 07

About QuillAudits 08

Executive Summary

Project Name	BabyDoge Burner
Overview	The contract is designed to permit whitelisted users to transfer liquidity pool tokens from the Liquidity pool collector (must have approved the contract), remove the liquidity, and send the BNB to the assigned BNB receiver. It further allows whitelisted users to call a function that burns the BabyDoge token by sending it to the dead wallet. The contract owner have some privileged functions for the functionality of the protocol.
Timeline	24 May, 2023 - 30 May, 2023
Method	Manual Review, Functional Testing, Automated Testing etc.
Language	Solidity
Blockchain	BSC
Scope of Audit	The scope of this audit was to analyze the BabyDoge codebase for quality, security, and correctness. *https://github.com/Baby-doge/BDburner/blob/develop/contracts/BDburner.sol* **Branch:** Main **Commit:** bbbd702e604cac3b6f23dea259f3216e8c567cf9
Contracts in Scope	**BDburner.sol**
Fixed In	NA



High
Medium
Low
Informational

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	0	0	1
Resolved Issues	0	0	0	0

Types of Severities

High

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

These are the issues identified in the initial audit and have been successfully fixed.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.

Checked Vulnerabilities

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- Exception Disorder
- Gasless Send
- Use of tx.origin
- Compiler version not fixed
- Address hardcoded
- Divide before multiply
- Integer overflow/underflow
- Dangerous strict equalities
- Tautology or contradiction
- Return values of low-level calls
- Missing Zero Address Validation
- Private modifier
- Revert/require functions
- Using block.timestamp
- Multiple Sends
- Using SHA3
- Using suicide
- Using throw
- Using inline assembly

Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

Static analysis of smart contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis.

Manual Testing

A. Contract - BDburner.sol

High Severity Issues

No issues found.

Medium Severity Issues

No issues found.

Low Severity Issues

No issues found.

Informational Issues

1. Floating Solidity Version (pragma solidity ^0.8.0)

Description

Contract has a floating solidity pragma version. This is present also in inherited contracts. Locking the pragma helps to ensure that the contract does not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively. The recent solidity pragma version also possesses its own unique bugs.

Remediation

Making the contract use a stable solidity pragma version prevents bugs occurrence that could be ushered in by prospective versions. It is recommended, therefore, to use a fixed solidity pragma version while deploying to avoid deployment with versions that could expose the contract to attack.

Status

Acknowledged

Functional Testing

Some of the tests performed are mentioned below:

- Should successfully burn liquidity pool token and transfer BNB to BNB receiver.
- Should successfully burn Babydoge token derived from the withdrawal from the pair
- Should reverts when non-whitelisted addresses calls any of the privileged functions
- Should revert when there is insufficient liquidity pool token to burn.
- Should revert when there is insufficient Babydoge tokens to transfer to the dead wallet
- Should allow only the owner to recover BNB and other tokens in the contract
- Should allow only the owner should be able to whitelist an address
- Should allow only the owner to remove an address from whitelisted addresses
- Should allow only the owner to assign the bnb receiver address
- Should allow the retrieval of all whitelisted addresses
- Should allow users to know if an account is whitelisted or not

Automated Tests

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Summary

In this report, we have considered the security of BabyDoge. We performed our audit according to the procedure described above.

No Major Issue Found in Contract.

Disclaimer

QuillAudits smart contract audit is not a security warranty, investment advice, or an endorsement of the BabyDoge Platform. This audit does not provide a security or correctness guarantee of the audited smart contracts.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the BabyDoge Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem.

850+
Audits Completed

$16B
Secured

800K
Lines of Code Audited

Follow Our Journey

Audit Report
May, 2023

For

BabyDoge

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com

audits@quillhash.com