



KISHIELD

Security Audit



Deflationary Delfa Token

April 14, 2022

Delfa



Table of Contents

1 Audit Summary

2 Project Overview

2.1 Token Summary

2.2 Main Contract Assessed

3 Smart Contract Vulnerability Checks

4 Contract Ownership

4.1 Privileged Functions

5 Important Notes To The Users

6 Findings Summary

6.1 Classification of Issues

6.1 Findings Table

01 Too many digits

02 Uninitialized local variables

03 Variables could be declared as constant

04 Public function that could be declared external

05 Boolean equality

7 Statistics

7.1 Liquidity

7.2 Token Holders

7.3 Liquidity Holders

8 Liquidity Ownership

9 Disclaimer



Audit Summary

This report has been prepared for Deflationary Delfa Token on the Binance Chain network. KISHIELD provides both client-centered and user-centered examination of the smart contracts and their current status when applicable. This report represents the security assessment made to find issues and vulnerabilities on the source code along with the current liquidity and token holder statistics of the protocol.

A comprehensive examination has been performed, utilizing Cross Referencing, Static Analysis, In-House Security Tools, and line-by-line Manual Review.

The auditing process pays special attention to the following considerations:

- Ensuring contract logic meets the specifications and intentions of the client without exposing the user's funds to risk.
- Testing the smart contracts against both common and uncommon attack vectors.
- Inspecting liquidity and holders statistics to inform the current status to both users and client when applicable.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Verifying contract functions that allow trusted and/or untrusted actors to mint, lock, pause, and transfer assets.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Project Overview

Token Summary

Parameter	Result
Address	0x0bb03a77dafaca79f084f4f65d4722d76aff99d9
Name	Deflationary Delfa
Token Tracker	Deflationary Delfa (DFLA)
Decimals	18
Supply	10,000,000
Platform	Binance Chain
compiler	v0.8.10+commit.fc410830
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://bscscan.com/ address/0x0bb03a77dafaca79f084f4f65d4722d76aff99d9
Url	https://deflationary.finance/

Main Contract Assessed

Name	Contract	Live
Deflationary Delfa	0x0bb03a77dafaca79f084f4f65d4722d76aff99d9	Yes

Smart Contract Vulnerability Checks

Vulnerability	Automatic Scan	Manual Scan	Result
Unencrypted Private Data On-Chain	Complete	Complete	✓ Low / No Risk
Code With No Effects	Complete	Complete	✓ Low / No Risk
Message call with hardcoded gas amount	Complete	Complete	✓ Low / No Risk
Hash Collisions With Multiple Variable Length Arguments	Complete	Complete	✓ Low / No Risk
Unexpected Ether balance	Complete	Complete	✓ Low / No Risk
Presence of unused variables	Complete	Complete	✓ Low / No Risk
Right-To-Left-Override control character (U+202E)	Complete	Complete	✓ Low / No Risk
Typographical Error	Complete	Complete	✓ Low / No Risk
DoS With Block Gas Limit	Complete	Complete	✓ Low / No Risk
Arbitrary Jump with Function Type Variable	Complete	Complete	✓ Low / No Risk
Insufficient Gas Griefing	Complete	Complete	✓ Low / No Risk
Incorrect Inheritance Order	Complete	Complete	✓ Low / No Risk
Write to Arbitrary Storage Location	Complete	Complete	✓ Low / No Risk
Requirement Violation	Complete	Complete	✓ Low / No Risk
Missing Protection against Signature Replay Attacks	Complete	Complete	✓ Low / No Risk
Weak Sources of Randomness from Chain Attributes	Complete	Complete	✓ Low / No Risk

Vulnerability	Automatic Scan	Manual Scan	Result
Authorization through tx.origin	Complete	Complete	✓ Low / No Risk
Delegatecall to Untrusted Callee	Complete	Complete	✓ Low / No Risk
Use of Deprecated Solidity Functions	Complete	Complete	✓ Low / No Risk
Assert Violation	Complete	Complete	✓ Low / No Risk
Reentrancy	Complete	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low / No Risk
Unprotected Ether Withdrawal	Complete	Complete	✓ Low / No Risk
Unchecked Call Return Value	Complete	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	Complete	✓ Low / No Risk
Integer Overflow and Underflow	Complete	Complete	✓ Low / No Risk
Function Default Visibility	Complete	Complete	✓ Low / No Risk

Contract Ownership

The contract ownership of Deflationary Delfa is not currently renounced. The ownership of the contract grants special powers to the protocol creators, making them the sole addresses that can call sensible ownable functions that may alter the state of the protocol.

The current owner is the address 0x11ac05cfd520c3fa2d08c3ba24413cfb8d67a72e which can be viewed from:
[HERE](#)

The owner wallet has the power to call the functions displayed on the privileged functions chart below, if the owner wallet is compromised this privileges could be exploited.

We recommend the team to renounce ownership at the right timing if possible, or gradually migrate to a timelock with governing functionalities in respect of transparency and safety considerations.

Important Notes To The Users:

- The owner cannot mint tokens after initial deployment.
- The owner cannot stop Trading.
- The owner cannot change the fees.
- The owner cannot change the max tx amount.
- Team states that the contract lock mechanism is to protect users for a potential price rebase with a malicious LP.
- Once the owner renounces ownership of the contract, none of the following are applicable.
- The owner can remove BNB and tokens stuck in the contract
- The owner can exclude addresses from tax.
- The owner can lock contracts for a period of time, and change the lock time, this may include the pair and router contracts.
- The owner can change pair address.
- The owner can remove locks.
- No high-risk Exploits/Vulnerabilities Were Found in token Source Code other than owner privileges.

Audit Passed



Findings Summary

Classification of Issues

Severity	Description
● High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency
● Medium	Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
● Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
● Info	Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

Findings

Severity	Found
● High	0
● Medium	0
● Low	0
● Info	5
Total	5

Findings

Too many digits

ID	Severity	Contract	Function
01	● Informational	Deflationary Delfa	Variables INITIAL_SUPPLY

Description

Literals with many digits are difficult to read and review.

Recommendation

Make use of scientific notation, use underscores, and/or use ether suffix.

Uninitialized local variables

ID	Severity	Contract	Function
02	● Informational	Deflationary Delfa	function _rLockBalanceOf()

Description

Variable total.

Recommendation

Initialize all the variables. If a variable is meant to be initialized to zero, explicitly set it to zero to improve code readability.

Variables could be declared as constant

ID	Severity	Contract	Function
03	● Informational	Deflationary Delfa	variables BURN_TAX, DEAD, WBNB

Description

Gas Optimization. Variables that are never changed could be declared as constant.

Recommendation

We recommend declaring those variables as constant.

Public function that could be declared external

ID	Severity	Contract	Function
04	● Informational	Deflationary Delfa	Functions authorize, unauthorize, transferOwnership, lockBalanceOf, viewBatches, getCirculatingSupply, rescueToken

Description

Gas Optimization. Public function that could be declared external

Recommendation

Public functions that are never called by the contract should be declared external to save gas.

Boolean equality

ID	Severity	Contract	Function
05	● Informational	Deflationary Delfa	Functions transfer() transferFrom()

Description

Statement 'excludedFromTax[sender] == true' Boolean constants can be used directly and do not need to be compare to true or false.

Recommendation

Remove the equality to the boolean constant.

Privileged Functions (onlyOwner)

Function Name	Parameters	Visibility
authorize	address adr	public
unauthorize	address adr	public
transferOwnership	address adr	public
setLockContract	address _contractAddress, uint256 _timestamp	external
changeUnlockTimestamp	address _contractAddress, uint256 _timestamp	external
setLP	address _address	external
removeBatches	address _address	external
clearStuckBalance	none	external
rescueToken	address tokenAddress, uint256 tokens	public

Statistics

Liquidity Info

Parameter	Result
Pair Address	0xF6Ec380545098fD9a7fccF383C3710c11DC32803
DFLA Reserves	0.28 DFLA
BNB Reserves	0.00 BNB
Liquidity Value	\$0 USD

Token (DFLA) Holders Info

Parameter	Result
DFLA Percentage Burnt	0.00%
DFLA Amount Burnt	0 DFLA
Top 10 Percentage Own	99.9%
Top 10 Amount Owned	9,999,946.014 DFLA
Top 10 Aprox Value	\$0 USD

LP (DFLA/BNB) Holders Info

Parameter	Result
DFLA/BNB % Burnt	0.00%
DFLA/BNB Amount Burnt	0 DFLA
Top 10 Percentage Owned	0.00%
Top 10 Amount Owned	0 DFLA
Locked Tokens Percentage	0.00%
Locked Tokens Amount	0 DFLA

* All the data displayed above was taken on-chain at block 16942937

* The tokens on industry-standard burn wallets are not included on the top 10 wallets calculations

Liquidity Ownership

The token does not have liquidity at the moment of the audit, block 16942937

KISHIELD



Disclaimer

KISHIELD has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocacy for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and KISHIELD is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will KISHIELD or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

The assessment services provided by KISHIELD is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.