HackSafe

Smart Contract Security Audit Report

BitcoinReflect.finance

December 2022

Security Status

★★★★☆

www.hacksafe.io

Audit Details

Audited project
BitcoinReflect.finance

Deployer address
0x7bbf2fb7f137eeb1b9a670436bd9348db9e949ba

Client contacts
BitcoinReflect.finance Team

Blockchain
Binance smart chain

Website
Not provided

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and HackSafe hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Procedure

Step 1 - In-Depth Manual Review

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

Step 2 - Automated Testing

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

Step 3 – Leadership Review

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

Step 4 - Resolution of Issues

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

Step 5 - Published Audit Report

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

Background

HackSafe was commissioned by BitcoinReflect.finance to perform an audit of smart contracts:

- https://bscscan.com/token/0x893F9C19e2b9f7865f6d0f953C38B34A3a6c17cF#code

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contract Details

Token contract details for 01.12.2022

Token Type	: DEFI
Contract name	: BTCR
Contract address	: 0x893F9C19e2b9f7865f6d0f953C38B34A3a6c17cF
Total supply	: 100,000,000,000,000,000
Token ticker	: BTCR
Decimals	: 9
Token Holders	: 6,123
Transactions count	: 38,075
Compiler version	: v0.6.12+commit.27d51765
Contract deployer address	: 0x7bbf2fb7f137eeb1b9a670436bd9348db9e949ba
Owner address	: 0x0000000000000000000000000000000000000000

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **“Secure”.** This token contract does not contain owner control as ownership has been renounced, which do make it fully decentralized as owner does not have control over smart contract.

Insecure	Poor secured	Secure	Well-secured

You are here 

We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 0 high, 1 medium and 1 low.

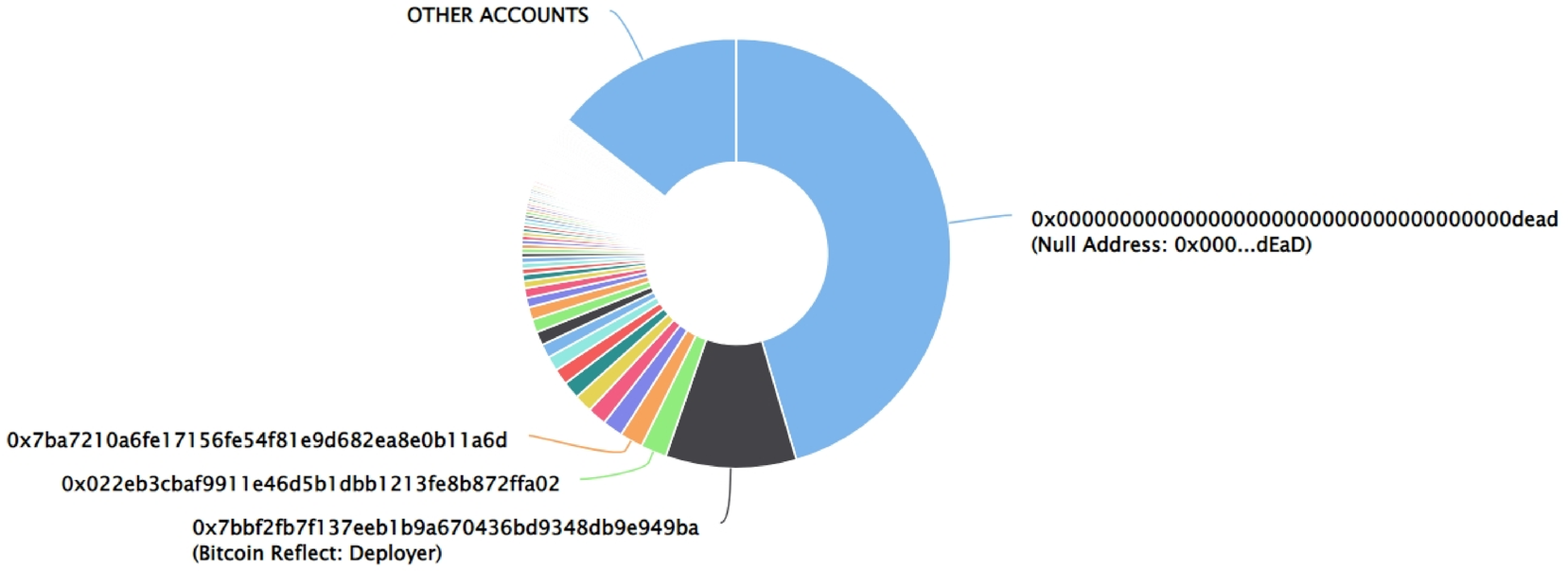
BitcoinReflect.finance Distribution

The top 100 holders collectively own 85.66%
(85,655,937,073,902,000.00 Tokens) of BitcoinReflect.finance

Token Total Supply: 100,000,000,000,000,000.00 Token | Total Token Holders: 6,12:

BitcoinReflect.finance Top 100 Token Holders

Source: BscScan.com



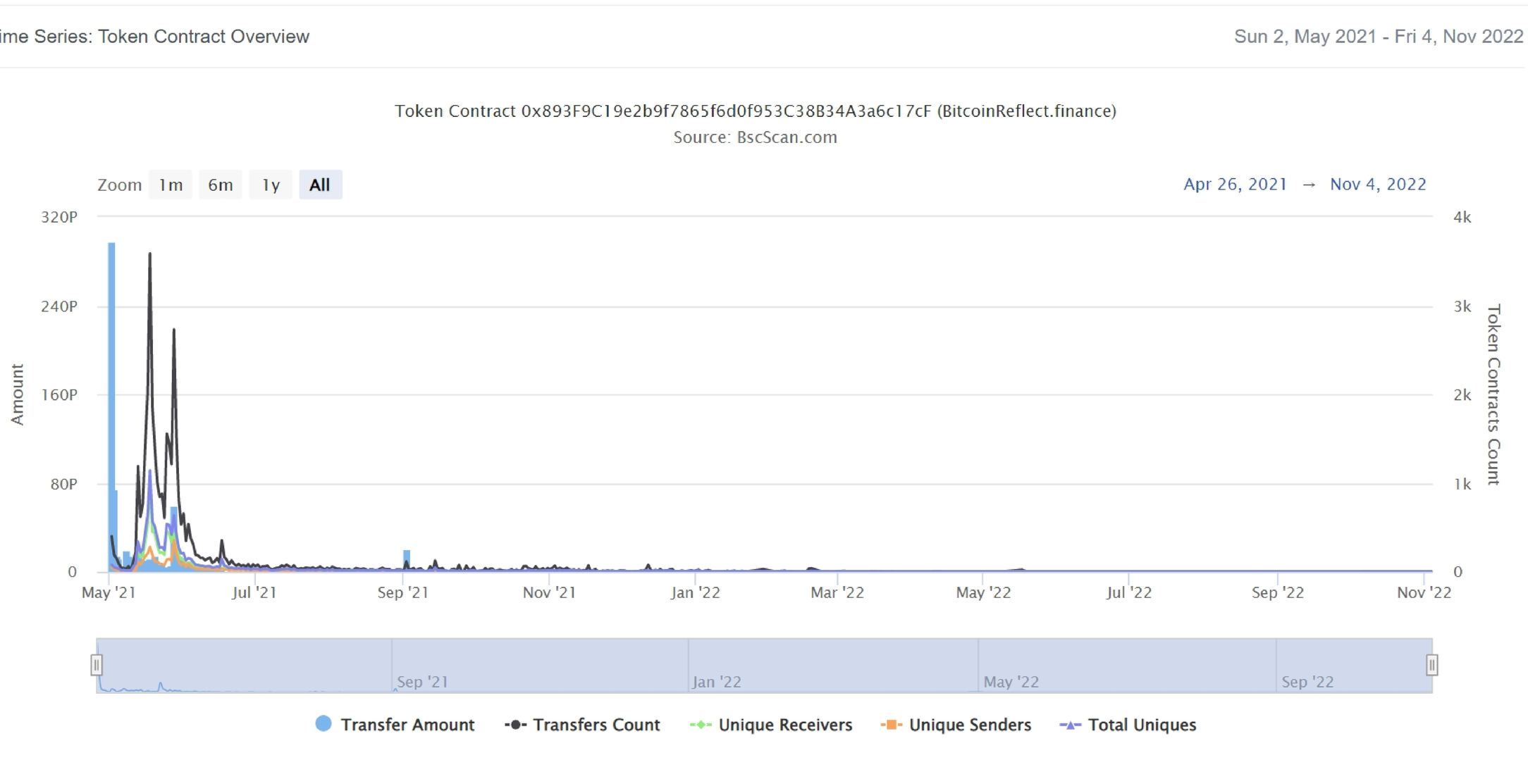
BitcoinReflect.finance Top 20 Token Holders

(A total of 85,655,937,073,902,000.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Null Address: 0x000...dEaD	45,482,681,834,114,200.845560793	45.4827%
2	Bitcoin Reflect: Deployer	9,846,741,752,445,350.695079685	9.8467%
3	0x022eb3cbaf9911e46d5b1dbb1213fe8b872ffa02	1,958,627,327,222,040.767170296	1.9586%
4	0x7ba7210a6fe17156fe54f81e9d682ea8e0b11a6d	1,735,812,336,237,650.435463754	1.7358%
5	0x68900fa8d06ecb9a87bbf9438dd12a84fa1b59d3	1,538,212,781,550,960.960214043	1.5382%
6	0x0c8cb89d99cff4fae3f62a673448682329cf6d57	1,432,898,933,139,860.684006554	1.4329%
7	0x7a0f7f7692b78049901436864fb7586a79431f1b	1,363,130,987,879,330.161081526	1.3631%
8	0xc438a115ffc125feecf8a081b5eb2fb8d9d7168e	1,304,315,548,811,000.680366729	1.3043%
9	0x20a86148fce6096e64ac9aef5355485600491ed3	1,197,271,802,936,970.773079276	1.1973%
10	0x6b1ecb203d82fec24f8b684f7906ef98ec9abf0a	1,116,995,994,156,810.640134907	1.1170%
11	0x0bc716da0c86086608cf72d27ea926a85d67837a	1,023,082,665,924,150.155022499	1.0231%
12	0xed6e835dd031b3e53e647aee5d513546af6e713a	1,019,461,821,479,110.766307047	1.0195%
13	0xf3232104dce8bcd5c7d804e6ae84957e4e821060	994,608,946,182,222.864864396	0.9946%
14	0x626d78be54ef823529a68f4a72057c8a536beba1	934,425,716,662,090.916939482	0.9344%
15	0x5957140844f33a7357f0ce4fc0710dc40090a87e	734,131,580,797,121.428271666	0.7341%
16	0xaaeefeddfb5ca96119d1d4cacb98c18d99e6c78f	732,528,738,768,554.319857958	0.7325%
17	0xd0c6b5917913490fb2d70ccc290385fb1c94692a	517,742,192,595,399.41440026	0.5177%
18	0x6585a29fb9daed5631bcb1b201b7f3527d1fb6bd	514,869,911,621,962.912701686	0.5149%
19	0x424c0b3d6d251445c140d0d71786786c36499f71	434,035,905,411,329.51119137	0.4340%
20	0xa249933e8d1e4f95dbc3b90a2cc1b4b587e0befc	413,213,386,734,129.800047133	0.4132%

BitcoinReflect.finance Distribution

BitcoinReflect.finance Overview



Contract functions details

+Context
    -[Int] _msgSender
    -[Int] _msgData

+[Int] IERC20
    -[Ext] totalSupply
    -[Ext] balanceOf
    -[Ext] transfer
    -[Ext] allowance
    -[Ext] approve
    -[Ext] transferFrom

+[Lib] SafeMath
    -[Int] add
    -[Int] sub
    -[Int] sub
    -[Int] mul
    -[Int] div
    -[Int] mod
    -[Int] mod

+[Lib] Address
    -[Int] isContract
    -[Int] sendValue
    -[Int] functionCall
    -[Int] functionCall
    -[Int] functionCallWithValue
    -[Int] functionCallWithValue
    -[Int] _functionCallWithValue

+Ownable (Context)
    -[Int]< constructor>
    -[Pub] owner
    -[Pub] renounceOwnership #
      -modifiers: onlyOwner
    -[Pub] transferOwnership #
      -modifiers: onlyOwner

Contract functions details

+BTCR (Context, IERC20, Ownable)
- -[Pub] <constructor> #
- -[Pub] name
- -[Pub] symbol
- -[Pub] decimals
- -[Pub] totalSupply
- -[Pub] balanceOf
- -[Pub] transfer #
- -[Pub] transferFrom #
- -[Pvt] _transfer #
- -[Pub] allowance
- -[Pub] approve #
- -[Pub] increaseAllowance #
- -[Pub] decreaseAllowance #
- -[Pub] isExcluded
- -[Pub] totalFees
- -[Pub] reflect
- -[Pub] reflectionFromToken
- -[Pub] tokenFromReflection
- -[Ext] excludeAccount #
  - -modifiers: onlyOwer
- -[Ext] includeAccount #
  - -modifiers: onlyOwner
- -[Pvt] _approve #
- -[Pvt] _getUValues
- -[Pvt] _transferStandard #
- -[Pvt] _transferToExcluded #
- -[Pvt] _transferFromExcluded #
- -[Pvt] _transferBothExcluded #
- -[Pvt] _reflectFee #
- -[Pvt] _getValues
- -[Pvt] _getTValues
- -[Pvt] _getRValues
- -[Pvt] _getRate
- -[Pvt] _getCurrentSupply

($) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Passed
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Medium issue
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Passed
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed
20.	Too old version	Low issue

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
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 can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

Critical Severity Issues

No critical severity issue found.

High Severity Issues

No high severity issue found.

Medium Severity Issues

One medium severity issue found.

1. Out of gas limit

- **Description**

  The function includeInReward() uses the loop to find and remove addresses from the _excluded list. Function will be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

  The function _getCurrentSupply also uses the loop for evaluating total supply. It also could be aborted with OUT_OF_GAS exception if there will be a long excluded addresseses list

- **Recommendation**

  Use EnumerableSet instead of array or do not use long arrays.

Low Severity Issues

One low severity issue found.

1. Old compiler version

- **Description**

  Contract has been deployed using too old solidity version.

- **Recommendation**

  It is advisable to deploy contract using any of the latest version of solidity

Centralization

Owner Privileges :

- BitcoinReflect.finance Contract:
    - Owner can renounce and transfer ownership.
    - Owner can exclude and include account.

This smart contract has some functions which can be executed by the admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble but smart contract ownership has been renounced. Following are Admin functions:

- renounceOwnership
- transferOwnership
- excludeAccount
- includeAccount

Conclusion

Smart contract contains low and medium severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

HackSafe note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.