



Smart Contract Security Audit Report

Rupee Token

November 2022

Security Status



www.hacksafe.io



Audit Details



Audited project

Rupee Token



Deployer address

0xb99634cc08ead4989485c1ead5fb864c45ff84b5



Client contacts

Rupee Token Team



Blockchain

Binance smart chain



Website

<https://app.hyruleswap.com/>

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 Rupee Token to perform an audit of smart contracts:

- <https://bscscan.com/token/0x7B0409A3A3f79bAa284035d48E1DFd581d7d7654#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 understood to 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.

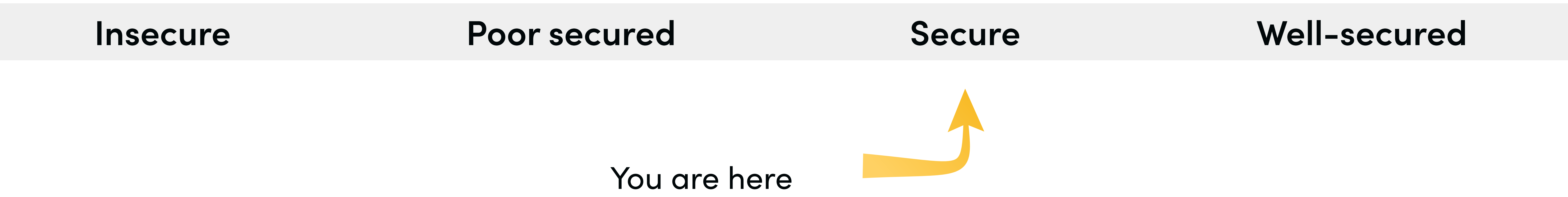
Contract Details

Token contract details for 24.11.2022

Token Type	: UTILITY
Contract name	: RupeeToken
Contract address	: 0x7B0409A3A3f79bAa284035d48E1DFd581d7d7654
Total supply	: 14,937,574.712702
Token ticker	: RUPEE
Decimals	: 18
Token Holders	: 2,320
Transactions count	: 588,637
Compiler version	: v0.6.12+commit.27d51765
Contract deployer address	: 0xb99634cc08ead4989485c1eaa5fb864c45ff84b5
Owner address	: 0xb99634cc08ead4989485c1eaa5fb864c45ff84b5

Audit Summary

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



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.

Social profiles

Telegram profile : <https://t.me/hyruleswap>

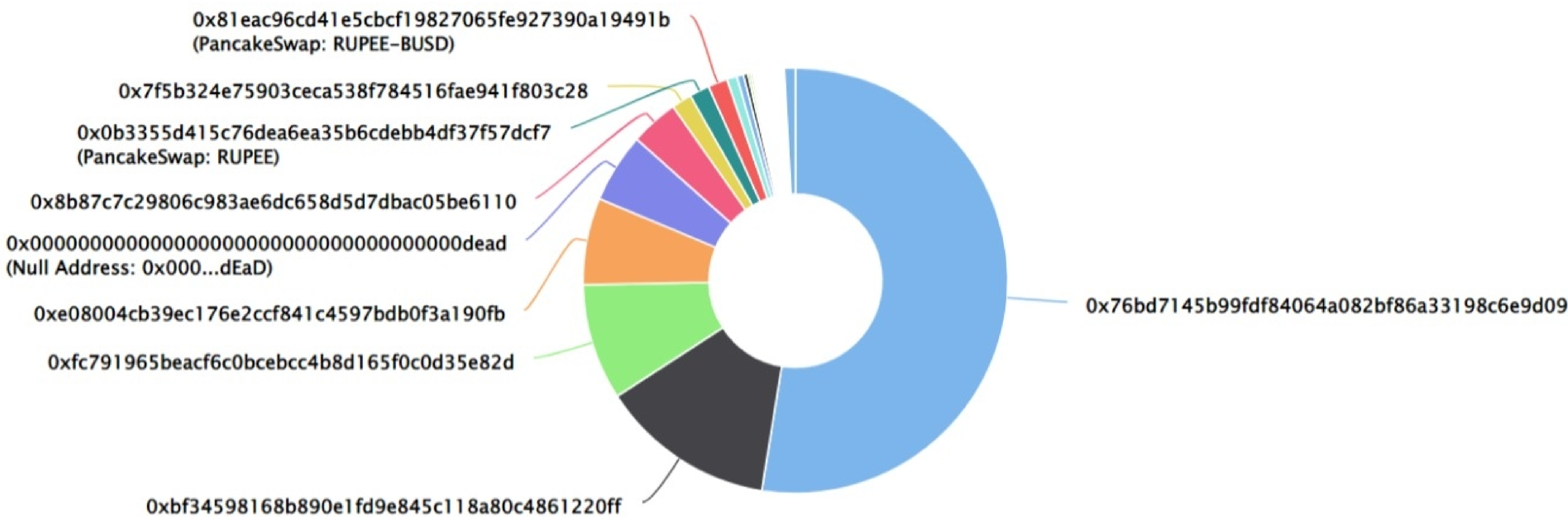
RupeeToken Distribution

💡 The top 100 holders collectively own 99.16% (14,812,126.12 Tokens) of Rupee Token

💡 Token Total Supply: 14,937,574.71 Token | Total Token Holders: 2,320












Rupee Token Top 100 Token Holders

Source: BscScan.com



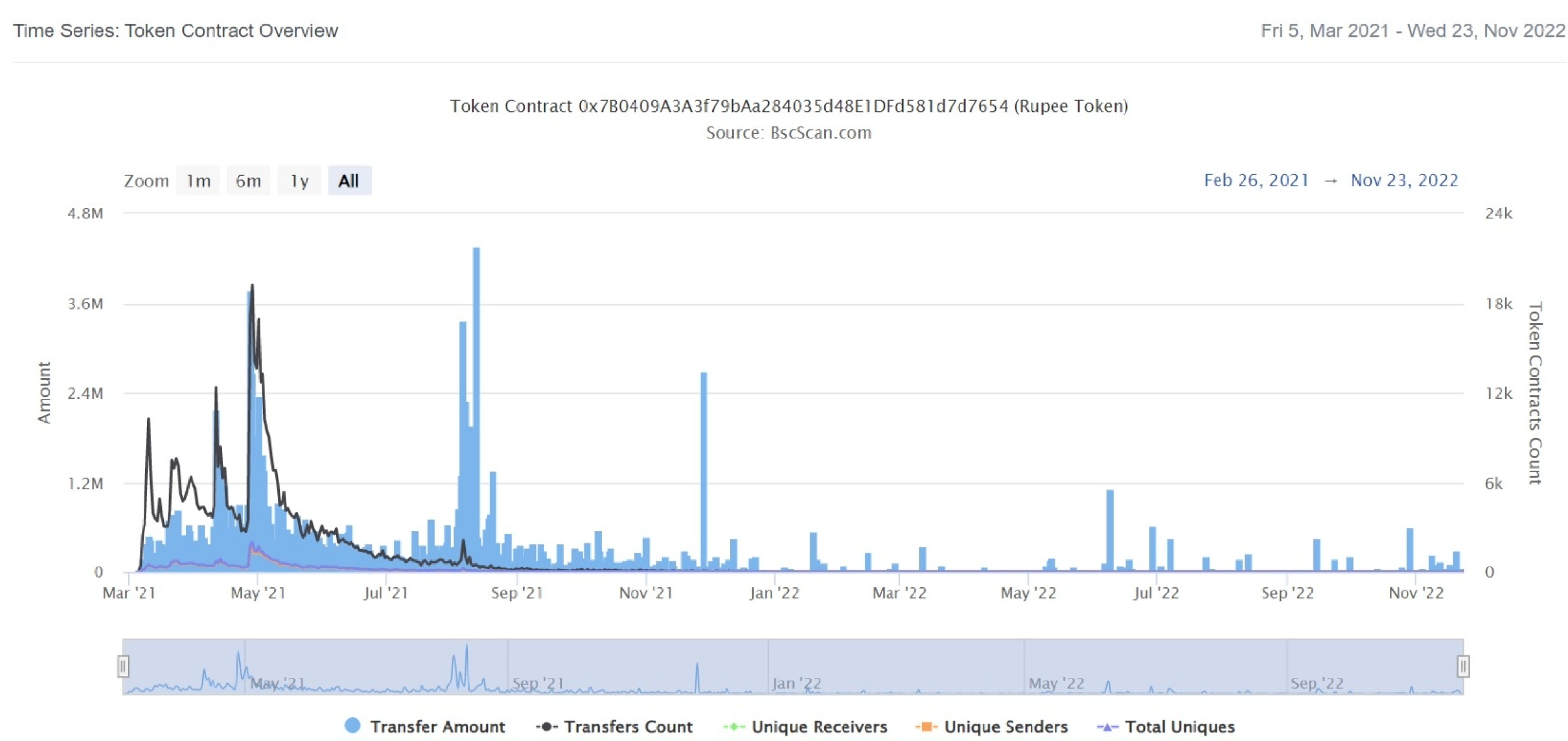
RupeeToken Top 20 Token Holders

(A total of 14,812,126.12 tokens held by the top 100 accounts from the total supply of 14,937,574.71 token)

Rank	Address	Quantity (Token)	Percentage
1	 0x76bd7145b99fdf84064a082bf86a33198c6e9d09	7,850,184.410716343361517729	52.5533%
2	 0xbf34598168b890e1fd9e845c118a80c4861220ff	1,990,479.101870143621228093	13.3253%
3	 0xfc791965beacf6c0bcebcc4b8d165f0c0d35e82d	1,319,052.073886801554976712	8.8304%
4	0xe08004cb39ec176e2ccf841c4597bdb0f3a190fb	984,492.709898179741429585	6.5907%
5	Null Address: 0x000...dEaD	788,335.23721701139968154	5.2775%
6	 0x8b87c7c29806c983ae6dc658d5d7dbac05be6110	545,903.493763747441506532	3.6546%
7	 0x7f5b324e75903ceca538f784516fae941f803c28	229,618.475325255110479042	1.5372%
8	 PancakeSwap: RUPEE	228,192.01232268846186652	1.5276%
9	 PancakeSwap: RUPEE-BUSD	220,873.496207992406943768	1.4786%
10	 0x7bc1733c0b4f3e1ae5915e3f587a360ae1fe3310	116,431.255865724473374583	0.7795%
11	 0x343ba9ca372139a33fbe5025fd0e6eabb0de3a2b	76,527.26587859084918515	0.5123%
12	 0xe5085ce5f12516ba3077edf4a3b58c2cc8faabcd	49,656.930058023285603854	0.3324%
13	0xb35a8771c49da5253a15dbca13289ce7f6669804	29,669.525581734096422173	0.1986%
14	 0x8f38f8d770089ec72aec229deca1adc62eb6e8ed	27,868.546498798519285475	0.1866%
15	0xd31c63298aa7018030ee34118f55dbba63d692e0	21,162	0.1417%
16	0x3aac533fdeb9e58846ddcf10feb241b807d3515d	16,829.88763720692472861	0.1127%
17	0xa93a5569090a880f9ba22347f949dfdf77991cba	14,865.598530969949317345	0.0995%
18	0xdb9fd2e1334704526107c07b674cb06025ff2e67	13,500	0.0904%
19	0x1682bd5edce9f449f6c2fbc902548f6830fb20c1	12,595.719504073410766961	0.0843%
20	0x7f2a4c6038f43aa8f634190dcda9c92947591203	11,600.136047438781247028	0.0777%

RupeeToken Distribution

RupeeToken Contract Overview



Contract functions details

RupeeToken.sol

+RupeeToken (BEP20)

- [Pub] mint #
 - modifiers: onlyOwner
- [Ext] delegates
- [Ext] delegates
- [Ext] delegate
- [Ext] delegateBySig
- [Ext] getCurrentVotes
- [Ext] getPriorVotes
- [Int] _delegate
- [Int] _moveDelegates
- [Int] _writeCheckpoint
- [Int] safe32
- [Int] getChainId

BEP20.sol

+BEP20 (Context, IBEP20, Ownable)

- [Pub] <constructor>
- [Ext] getOwner
- [Pub] name
- [Pub] symbol
- [Pub] decimals
- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] transfer
- [Pub] allowance
- [Pub] approve
- [Pub] transferFrom
- [Pub] increaseAllowance
- [Pub] decreaseAllowance
- [Pub] mint #
 - modifiers: onlyOwner
- [Int] _transfer #
- [Int] _mint #
- [Int] _burn #
- [Int] _approve #
- [Int] _burnFrom #

Contract functions details

Ownable.sol

+Ownable (Context)

- [Int] <constructor>
- [Pub] owner
- [Pub] renounceOwnership #
 - modifiers: onlyOwner
- [Pub] transferOwnership #
 - modifiers: onlyOwner

IBEP20.sol

+[Int] IBEP20

- [Ext] totalSupply
- [Ext] decimals
- [Ext] symbol
- [Ext] name
- [Ext] getOwner
- [Ext] balanceOf
- [Ext] transfer
- [Ext] allowance
- [Ext] approve
- [Ext] transferFrom

SafeMath.sol

+ [Lib] SafeMath

- [Int] tryAdd
- [Int] trySub
- [Int] tryMul
- [Int] tryDiv
- [Int] tryMod
- [Int] add
- [Int] sub
- [Int] sub
- [Int] mul
- [Int] div
- [Int] mod
- [Int] sub
- [Int] div
- [Int] mod

Contract functions details

Context.sol

-[Int] _msgSender

-[Int] _msgData

(\$) = 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.	Passed
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.	Medium issue
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. Safe Open Zeppelin contracts implementation and usage.

• Description

The smart contract BEP20.sol has direct imported open zeppelin files, any changes in their contract can affect this smart contract too.

• Recommendation

It is advisable to not direct import smart contract from any github repository.

✓ 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 :

- Rupee Token Contract:
 - Owner can renounce and transfer ownership.
 - Owner can mint tokens.

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 as smart contract ownership has not been renounced. Following are Admin functions:

- `renounceOwnership`
- `transferOwnership`
- `mint`

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.