HackSafe

Smart Contract Security Audit Report

PhoenixDefi.Finance

November 2022

Security Status

★★★★☆

www.hacksafe.io

Audit Details

Audited project
PhoenixDefi.Finance

Deployer address
0x38a5c7055b960f45f28e932ed366f34e2357803b

Client contacts
PhoenixDefi.Finance Team

Blockchain
Binance smart chain

Website
https://phoenixdefi.finance/

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 PhoenixDefi.Finance to perform an audit of smart contracts:

- https://bscscan.com/token/0xc25D94fc3f8D7bD1d88f89802fe075338F71dEC7#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 24.11.2022

Token Type	: DEFI
Contract name	: PhoenixToken
Contract address	: 0xc25D94fc3f8D7bD1d88f89802fe075338F71dEC7
Total supply	: 43,714,888.597221
Token ticker	: PNIX
Decimals	: 8
Token Holders	: 7,178
Transactions count	: 56,062
Compiler version	: v0.6.12+commit.27d51765
Contract deployer address	: 0x38a5c7055b960f45f28e932ed366f34e2357803b
Owner address	: 0x38a5c7055b960f45f28e932ed366f34e2357803b

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.

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.

Social profiles

Twitter Profile	: https://twitter.com/phoenixdefi
Telegram profile	: https://t.me/PhoenixDefiFinanceCommunity
Coingecko profile	https://www.coingecko.com/en/coins/phoenixdefi-finance/

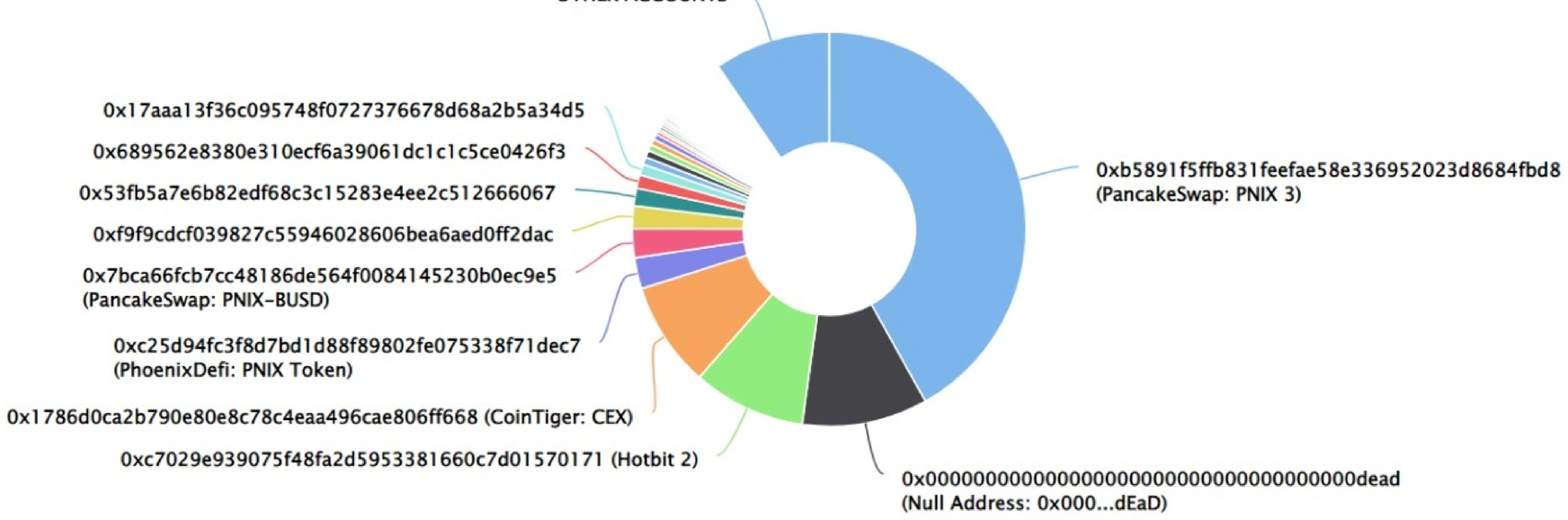
PhoenixDefi.Finance Distribution

The top 100 holders collectively own 90.46% (39,544,592.48 Tokens) of PhoenixDefi.Finance

Token Total Supply: 43,714,888.60 Token | Total Token Holders: 7,178

PhoenixDefi.Finance Top 100 Token Holders

Source: BscScan.com



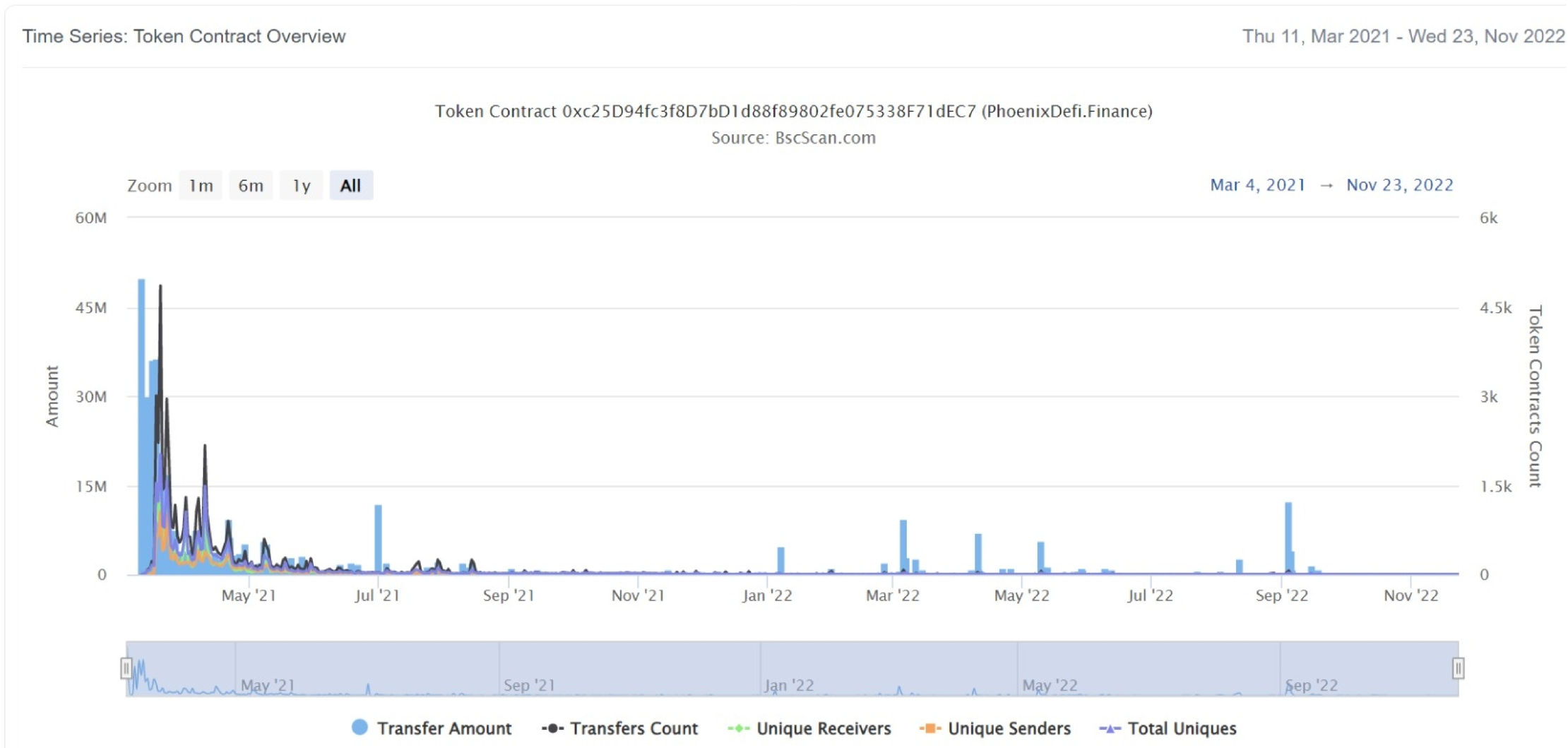
PhoenixDefi.Finance Top 20 Token Holders

(A total of 39,544,592.48 tokens held by the top 100 accounts from the total supply of 43,714,888.60 token)

Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap: PNIX 3	18,317,420.68862539	41.9020%
2	Null Address: 0x000...dEaD	4,509,689.17514913	10.3161%
3	Hotbit 2	4,046,540.59895801	9.2567%
4	CoinTiger: CEX	3,779,555.53550179	8.6459%
5	PhoenixDefi: PNIX Token	1,109,953.24178198	2.5391%
6	PancakeSwap: PNIX-BUSD	1,055,725.44694626	2.4150%
7	0xf9f9cdcf039827c55946028606bea6aed0ff2dac	795,998.16745172	1.8209%
8	0x53fb5a7e6b82edf68c3c15283e4ee2c512666067	633,013.13038087	1.4480%
9	0x689562e8380e310ecf6a39061dc1c1c5ce0426f3	484,811.79099634	1.1090%
10	0x17aaa13f36c095748f0727376678d68a2b5a34d5	386,526.6755274	0.8842%
11	PancakeSwap: PNIXS-PNIX	274,610.1531108	0.6282%
12	0x2fb3ab885db765538f9c13e5a52ae6465901efce	248,557.4437813	0.5686%
13	PancakeSwap: ETH-PNIX	232,160.76933437	0.5311%
14	0x55e4e18868bed48930dfddece76048337c43c38d	207,453.84010892	0.4746%
15	0x1486032b33da31bd62aa0db9591218a3e9c67f0a	201,686.17007993	0.4614%
16	0xd39522d5230694ff8dcf0a66831293308d393cac	115,776.58907672	0.2648%
17	0x8557e5a03f87e4f666e1cad080a1f46303943713	102,807.77349115	0.2352%
18	0xb7af0c8a698470ce49a4a305c269dadee08ee161	99,926.39473865	0.2286%
19	0xde807f8597ae0011b3bfe7540948eefa9ee2ba48	97,136.64353578	0.2222%
20	0x499c6965bea3a127271215b01cb3766ecf1aa139	96,528.3914784	0.2208%

PhoenixDefi.Finance Distribution

PhoenixDefi.Finance Contract Overview



Contract functions details

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

+[Int] IBEP20
    -[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] div
    -[Int] mod
    -[Int] mod

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

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

Contract functions details

+PhoenixToken (Context, IBEP20, Ownable)
-[Pub] <constructor>
-[Pub] name
-[Pub] symbol
-[Pub] decimals
-[Pub] totalSupply
-[Pub] balanceOf
-[Pub] transfer
-[Pub] allowance
-[Pub] approve
-[Pub] transferFrom
-[Pub] increaseAllowance
-[Pub] decreaseAllowance
-[Pub] isExcluded
-[Pub] totalFees
-[Pub] totalBurn
-[Pub] deliver #
-[Pub] reflectionFromToken #
-[Pub] tokenFromReflection #
-[Ext] excludeAccount #
-[Ext] includeAccount #
-[Pvt] _approve #
-[Pvt] _transfer #
-[Pvt] _transferStandard #
-[Pvt] _transferToExcluded #
-[Pvt] _transferFromExcluded #
-[Pvt] _transferBothExcluded #
-[Pvt] _reflectFee
-[Pvt] _getValues
-[Pvt] _getTValues
-[Pvt] _getRValues
-[Pvt] _getRate
-[Pvt] _getCurrentSupply
-[Pub] _getTaxFee
-[Pub] _getBurnFee
-[Pub] _getMaxTxAmount
-[Ext] _setTaxFee #
  -modifiers: onlyOwner
-[Ext] _setBurnFee #
  -modifiers: onlyOwner

($) = 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 smart contract has functions which has used for includeAccount, _getCurrentSupply. Large length of _excluded can cause an error of out of gas for these two functions.

- **Recommendation**

  It is advisable to either remove for loop or use smaller length to avoid the gas limit error while transaction.

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 :

- PhoenixDefi.Finance Contract:
    - Owner can renounce and transfer ownership.
    - Owner can exclude and include account.
    - Owner can set tax and burn fee.

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
- excludeAccount
- includeAccount
- _setTaxFee
- _setBurnFee

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.