HackSafe

Smart Contract Security Audit Report

PulseFeg

January 2023

Security Status

★★★★☆

www.hacksafe.io

Audit Details

Audited project
PulseFeg

Deployer address
0xf1662adf6d4d6de9d39f44e13f19db268ff3c01d

Client contacts
PulseFeg

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 PulseFeg to perform an audit of smart contracts:

- https://bscscan.com/token/0x1E5Dd94A6d7190ab77f834E2cCDF9072597ae4E3#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 06.01.2023

Token Type	: DEFI
Contract name	: PulseFeg
Contract address	: 0x1E5Dd94A6d7190ab77f834E2cCDF9072597ae4E3
Total supply	: 10,000,000,000
Token ticker	: PulseFeg
Decimals	: 9
Token Holders	: 3,914
Transactions count	: 27,815
Buyback fee	: 2
Marketing fee	: 2
Uniswap V2 pair	: 0x370389704399e2f51173dd20f01eb02e3dd3cd00
Compiler version	: v0.8.6+commit.11564f7e
Contract deployer address	: 0xf1662adf6d4d6de9d39f44e13f19db268ff3c01d
Contract owner address	: : 0xf1662adf6d4d6de9d39f44e13f19db268ff3c01d

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.

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 0 low.

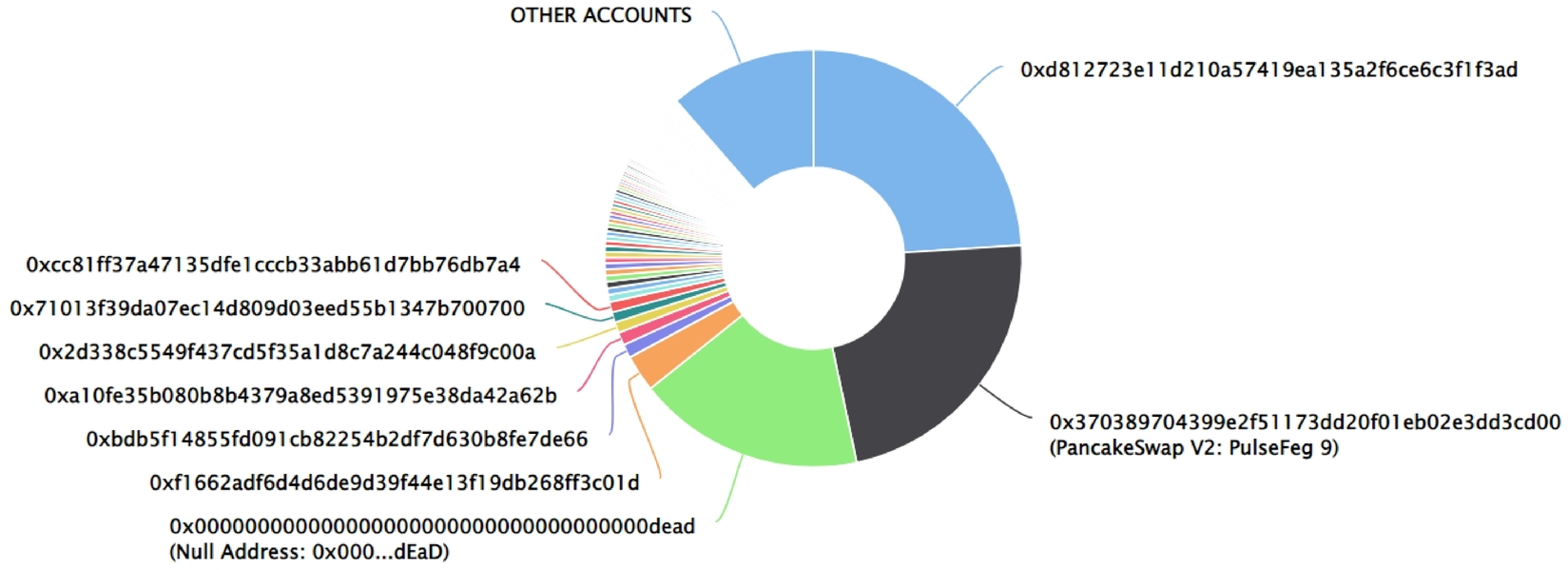
PulseFeg Token Distribution

The top 100 holders collectively own 88.58% (8,858,403,087.64 Tokens) of PulseFeg

Token Total Supply: 10,000,000,000.00 Token | Total Token Holders: 3,914

PulseFeg Top 100 Token Holders

Source: BscScan.com



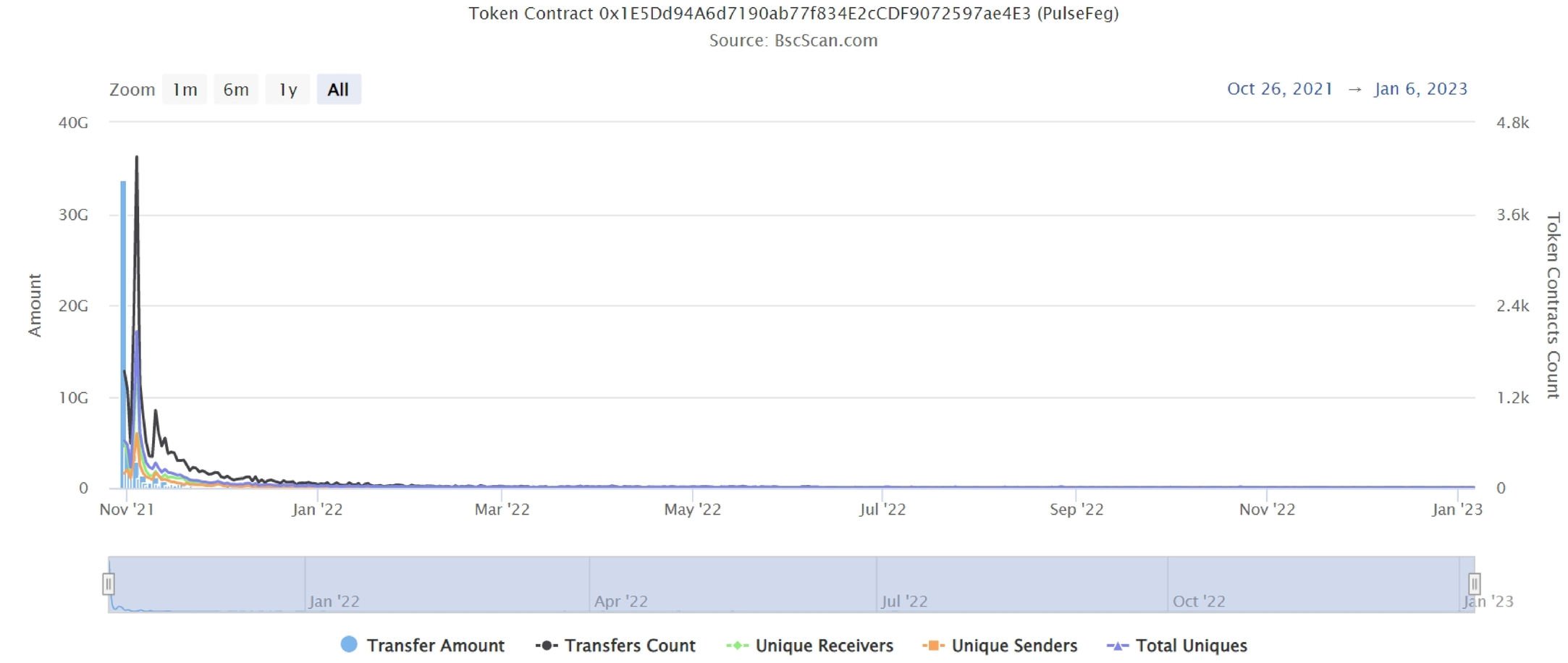
PulseFeg Top 20 Token Holders

(A total of 8,858,403,087.64 tokens held by the top 100 accounts from the total supply of 10,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xd812723e11d210a57419ea135a2f6ce6c3f1f3ad	2,400,000,000	24.0000%
2	PancakeSwap V2: PulseFeg 9	2,267,962,003.28190677	22.6796%
3	Null Address: 0x000...dEaD	1,761,697,113.732713541	17.6170%
4	0xf1662adf6d4d6de9d39f44e13f19db268ff3c01d	284,743,431.566209718	2.8474%
5	0xbdb5f14855fd091cb82254b2df7d630b8fe7de66	105,479,493.723617199	1.0548%
6	0xa10fe35b080b8b4379a8ed5391975e38da42a62b	100,053,255.020731105	1.0005%
7	0x2d338c5549f437cd5f35a1d8c7a244c048f9c00a	84,160,905.129478057	0.8416%
8	0x71013f39da07ec14d809d03eed55b1347b700700	80,000,000	0.8000%
9	0xcc81ff37a47135dfe1cccb33abb61d7bb76db7a4	79,957,738.49249479	0.7996%
10	0xd68b776b54b47b7a29e17151eb0e3938243d64ac	54,381,262.50220316	0.5438%
11	0xf0aa1c82de6546f2dd432612068d1b7575e50e8e	51,409,550.55275106	0.5141%
12	0xc2697196d08265850738d69e8bb0de45a301c6fc	51,241,164.935450557	0.5124%
13	0x669467c367c7ae0b8609dc27df9f2827940a12b4	48,034,060.484847167	0.4803%
14	0x22fc23e65f612a6190ff44204e5307849fffec99	47,939,239.699499461	0.4794%
15	0x82932d8ee68b65f4047bedcb87ba4214bbdc07ff	46,663,918.14325017	0.4666%
16	0x95ec1ad3ccfde4b083825701f125ff6ecb1e7aea	45,295,326.371945581	0.4530%
17	0x6452e3f906cf11f3a7f5b75ce7b3e44eec0be83b	44,718,778.40094747	0.4472%
18	0xdd4cd7e82d4be5bb81043fe37fd2616cf75f03c6	41,932,532.311635314	0.4193%
19	0x3f529523984f91908c551b4ac9432e1049888fd0	40,011,999.674502615	0.4001%
20	0x33e1381737e8e05ddb17d5236b382825686c6b69	37,100,419.66480611	0.3710%

PulseFeg Token Distribution

PulseFeg Contract Overview



Contract functions details

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

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

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

+[Int] IUniswapV2Factory
    -[Ext] createPair #

+[Int] IUniswapV2Router01
    -[Ext] factory
    -[Ext] WETH
    -[Ext] swapETHForExactTokens ($)

+[Int] IUniswapV2Router02 (IUniswapV2Router01)
    -[Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
    -[Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+PulseFeg (Context, IERC20, Ownable)
    -[Pub] <Constructor >#
    -[Pub] name
    -[Pub] symbol
    -[Pub] decimals
    -[Pub] totalSupply
    -[Pub] balanceOf
    -[Pub] transfer #
    -[Pub] allowance
    -[Pub] approve #

Contract functions details

-[Pub] transferFrom #
-[Pub] increaseAllowance #
-[Pub] decreaseAllowance #
-[Pub] isExcludedFromReward
-[Pub] totalFees
-[Pub] minimumTokensBeforeSwapAmount
-[Pub] buybackThresholdAmount
-[Pub] deliver #
-[Pub] reflectionFromToken
-[Pub] tokenFromReflection
-[Pub] excludeFromReward #
  - modifiers: onlyOwner
-[Ext] includeInReward #
  - modifiers: onlyOwner
-[Pvr] _approve #
-[Pvr] _transfer #
-[Pvr] swapTokens #
  - modifiers: lockTheSwap
-[Pvr] buyBackTokens #
  - modifiers: lockTheSwap
-[Pvr] swapTokensForEth #
-[Pvr] swapETHForTokens #
-[Pvr] _tokenTransfer #
-[Pvr] _transferStandard #
-[Pvr] _transferToExcluded #
-[Pvr] _transferFromExcluded #
-[Pvr] _transferBothExcluded #
-[Pvr] _reflectFee #
-[Pvr] _getValues
-[Pvr] _getTValues
-[Pvr] _getRValues
-[Pvr] _getRate
-[Pvr] _getCurrentSupply
-[Pvr] _takeSwap #
-[Pvr] calculateTaxFee
-[Pvr] calculateSwapFee
-[Pvr] removeAllFee #
-[Pvr] restoreAllFee #

Contract functions details

-[Pub] isExcludedFromFee
-[Pub] excludeFromFee #
  - modifiers: onlyOwner
-[Pub] includeInFee #
  - modifiers: onlyOwner
-[Ext] setTaxFeePercent #
  - modifiers: onlyOwner
-[Ext] setSwapFeePercent #
  - modifiers: onlyOwner
-[Ext] setMarketingFee #
  - modifiers: onlyOwner
-[Ext] setMarketingAddress #
  - modifiers: onlyOwner
-[Ext] setSellFees #
  - modifiers: onlyOwner
-[Ext] setMaxTxAmount #
  - modifiers: onlyOwner
-[Ext] setNumTokensSellToAddToBuyback #
  - modifiers: onlyOwner
-[Ext] setSwapUpperLimit #
  - modifiers: onlyOwner
-[Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
-[Pub] setSwapEnabled #
  - modifiers: onlyOwner
-[Ext] transferToAddress #
  - modifiers: onlyOwner
-[Ext] <Fallback>($)

($) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Compiler error	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	Passed

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

- **Issue:**
- 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 addresses list.
- **Recommendation**

  Check that the excluded array length is not too big.

Low Severity Issues

No low severity issue found.

Centralization

Owner Privileges :

- PulseFeg Contract:
    - Owner can exclude from the 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.

Conclusion

Smart contract contains 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.