HackSafe

Smart Contract Security Audit Report

Pup Token

September 2022

Security Status

★★★★½

www.hacksafe.io

Audit Details

Audited project
Pup Token

Deployer address
0x098350D31E0ee2912e21C2B679113ce7E64e148e

Client contacts
Pup Token Team

Blockchain
Polygon

Website
https://pup.polypup.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 Pup Token to perform an audit of smart contract:

- https://polygonscan.com/address/0xcfe2cf35d2bdde84967e67d00ad74237e234ce59#code

The purpose of the audit was to achieve the

- Ensutre 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 21.09.2022

Token Type	: ERC20
Contract name	: PupToken
Contract address	: 0xcFe2cF35D2bDDE84967e67d00aD74237e234CE59
Compiler version	: v0.6.12+commit.27d51765
Total supply	: 31,658.731487
Token ticker	: PUP
Decimals	: 18
Token holders	: 888
Transactions count	: 968,393
Contract deployer address	: 0x098350D31E0ee2912e21C2B679113ce7E64e148e
Owner address	: 0xcc7e7c9fc775d25176e9bfc5a400edac212aa81c

Social profiles

Twitter Profile	: https://twitter.com/PolyPup1
Telegram profile	: https://t.me/PolyPupFarm
Coingecko profile	: https://www.coingecko.com/en/coins/polypup/

Claimed Smart Contract Features

Claimed Feature Detail

Tokenomics :

- **Name** : Pup Token
- **Symbol** : PUP
- **Decimals** : 18
- **Protocol** : ERC20
- **Total supply** : 31,658.731487
- **Contract address** : 0xcFe2cF35D2bDDE84967e67d00aD74237e234CE59

Our Observation

YES, this is valid.

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, 0 medium and 1 low and some very low-level issues.

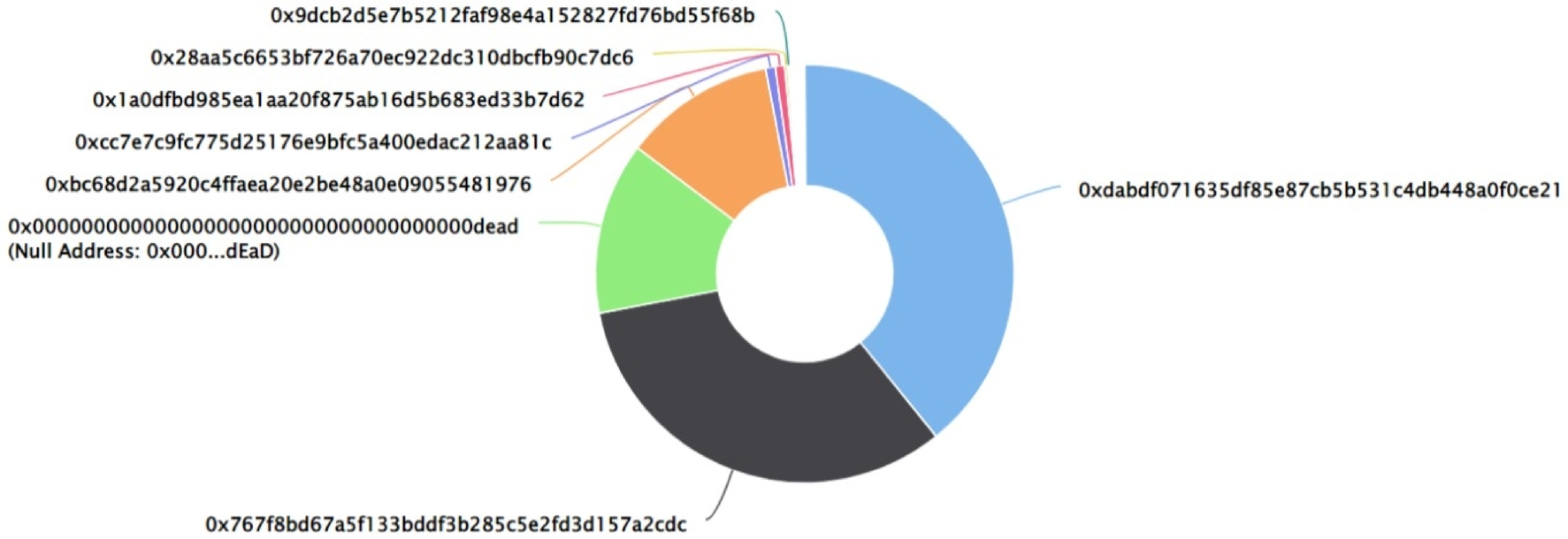
PupToken Distribution

The top 100 holders collectively own 99.98% (31,651.80 Tokens) of Pup Token

Token Total Supply: 31,658.73 Token | Total Token Holders: 888

Pup Token Top 100 Token Holders

Source: polygonscan.com



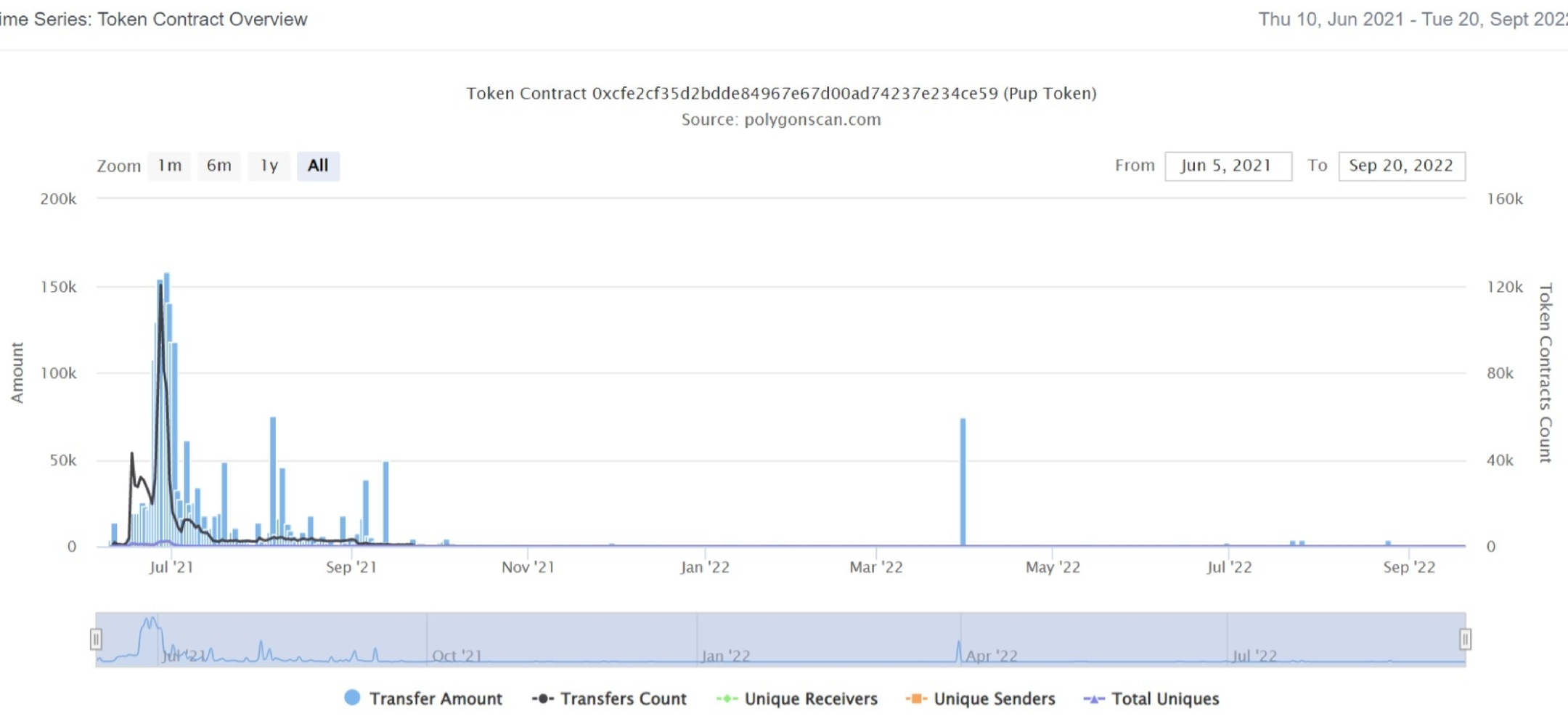
PupToken Top 20 Token Holders

(A total of 31,651.80 tokens held by the top 100 accounts from the total supply of 31,658.73 token)

Rank	Address	Quantity (Token)	Percentage
1	0xdabdf071635df85e87cb5b531c4db448a0f0ce21	12,409.580647184784936205	39.1980%
2	0x767f8bd67a5f133bddf3b285c5e2fd3d157a2cdc	10,380.66364124828408211	32.7893%
3	Null Address: 0x000...dEaD	4,206.821492112494920085	13.2880%
4	0xbc68d2a5920c4ffaea20e2be48a0e09055481976	3,710.515473133617141013	11.7204%
5	0xcc7e7c9fc775d25176e9bfc5a400edac212aa81c	237.024876634028828605	0.7487%
6	0x1a0dfbd985ea1aa20f875ab16d5b683ed33b7d62	229.046275578404851771	0.7235%
7	0x28aa5c6653bf726a70ec922dc310dbcfb90c7dc6	69.514945251593964909	0.2196%
8	0x9dcb2d5e7b5212faf98e4a152827fd76bd55f68b	42.755497916119026021	0.1351%
9	0x0d6ff4e5a446c3e7ae84c838bbf5e3b6b2e26093	38.654339716708856388	0.1221%
10	0x5c61658a350ed2d78096a016766776c18d0655af	38.131383766642398736	0.1204%
11	0x150fb0cfa5bf3d4023ba198c725b6dcbc1577f21	31.538607279277829507	0.0996%
12	0x3cefb8333a1e822029d4e731950797ea09a8e505	22.17212525756291792	0.0700%
13	0x1784e73fe44a31ef8a9695bb5e0976437b09436f	21.97106647659935394	0.0694%
14	0x7f8af9e942e9a8d28d3af63bb9dedf1e18ab313f	15.376745357434892843	0.0486%
15	0xa0865656aa89d9479b5738ba2856151f159e2369	15.218543432531927225	0.0481%
16	0xb8f8b3455e117e42ee7eb3dad3aff1483c908b43	12.614093152235073527	0.0398%
17	0xae3d5bb5190be74e14748bbbbe55ffad84965dec	12.225028346345444044	0.0386%
18	0xb5f383998d4e58c140c15c441c75bb79170b6b45	10.829129294279587352	0.0342%
19	0x736d97339b4c65be485639c6bd39d0d617bd44cb	10.0000000000000032	0.0316%
20	0x88e977fac9fd0e54f650afc566f2fb72f8b7029b	10	0.0316%

PupToken Distribution

PupToken Contract Overview



Contract functions details

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

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

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

Address.sol
+ [Lib] Address
    -[Int] isContract
    -[Int] sendValue
    -[Int] functionCall
    -[Int] functionCall
    -[Int] functionCallWithValue
    -[Int] functionCallWithValue
    -[Int] functionStaticCall
    -[Int] functionStaticCall
    -[Int] functionDelegateCall

Contract functions details

-[Int] functionDelegateCall
-[Pvt] _verifyCallResult

+[Int] IBEP20
-[Ext] totalSupply
-[Ext] decimals
-[Ext] symbol
-[Ext] name
-[Ext] getOwner
-[Ext] balanceOf
-[Ext] transfer
-[Ext] allowance
-[Ext] approve
-[Ext] transferFrom

+ BEP20 (Context, IBEP20, Ownable)
-[Pub] < constructor>
-[Ext] getOwner
-[Pub] name
-[Pub] decimals
-[Pub] symbol
-[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 #

+ PupToken (BEP20)
-[Pub] mint #

Contract functions details

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

($) = 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.	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 issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

One low severity issue found.

1. Too old compiler version.

- **Description**

  Contract has been deployed using too old compiler version.

- **Recommendation**

  It is advisable that the compiler version of solidity should be among the new compiler versions.

Centralization

Owner Privileges :

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

This smart contract has some functions which can be executed by the owner (Admin) only. If the admin wallet private key would be compromised, it would create trouble as smart contract ownership has not been renounced. Following are the only admin functions:

- Mint
- Renounceownership
- Transferownership

Conclusion

Smart contract contains low 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.