HackSafe

Smart Contract Security Audit Report

SafePal

January 2023

Security Status

★ ★ ★ ★ ⯪

www.hacksafe.io

Audit Details

Audited project
SafePal

Deployer address
0xbac93b5b19fec3d8da65a81bbf79f23d33a50a2d

Client contacts
SafePal

Blockchain
Ethereum

Website
https://www.safepal.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 SafePal to perform an audit of smart contracts:

- https://bscscan.com/token/0xd41fdb03ba84762dd66a0af1a6c8540ff1ba5dfb#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 20.01.2023

Token Type	: WALLET
Contract name	: SafePalToken
Contract address	: 0xD41FDb03Ba84762dD66a0af1a6C8540FF1ba5dfb
Total supply	: 500,000,000
Token ticker	: SFP
Decimals	: 18
Token Holders	: 156,268
Transactions count	: 3,446,560
Compiler version	: v0.6.12+commit.27d51765
Contract deployer address	: 0xbac93b5b19fec3d8da65a81bbf79f23d33a50a2d
Owner address	: No owner

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **“secure”**. This token contract does not contain owner control, which do 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, 0 medium and 1 low.

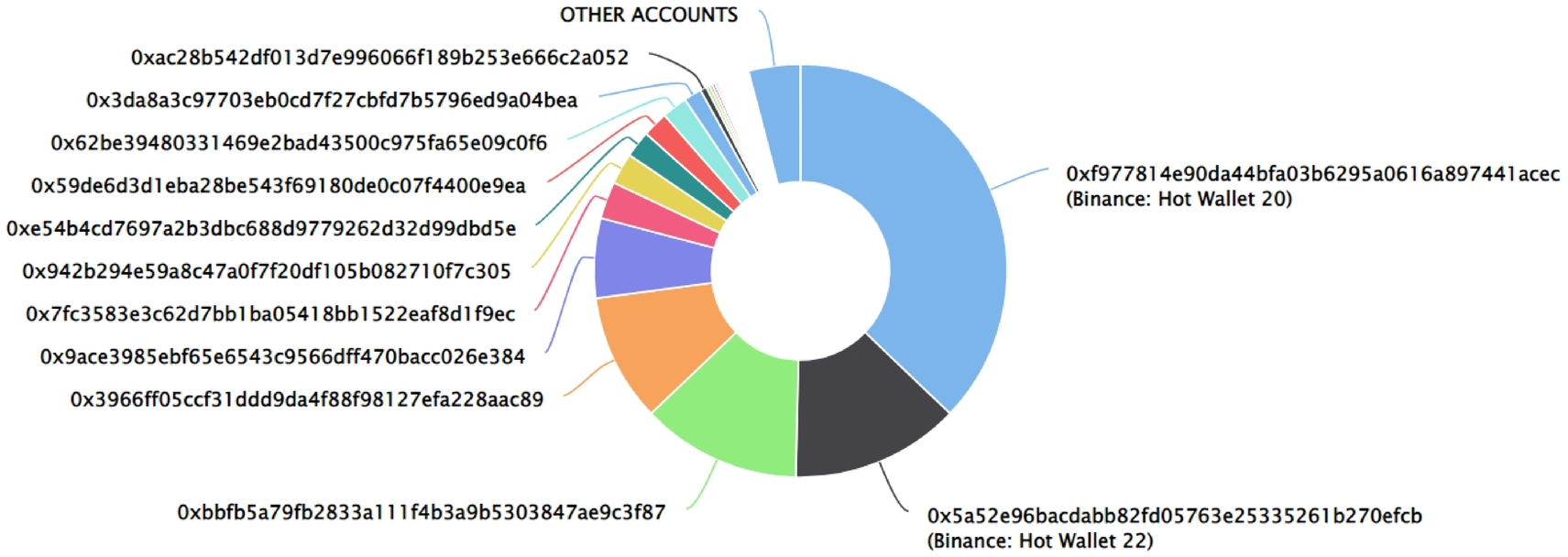
SafePal Token Distribution

The top 100 holders collectively own 95.95% (479,730,000.05 Tokens) of SafePal Token

Token Total Supply: 500,000,000.00 Token | Total Token Holders: 156,269

SafePal Token Top 100 Token Holders

Source: BscScan.com



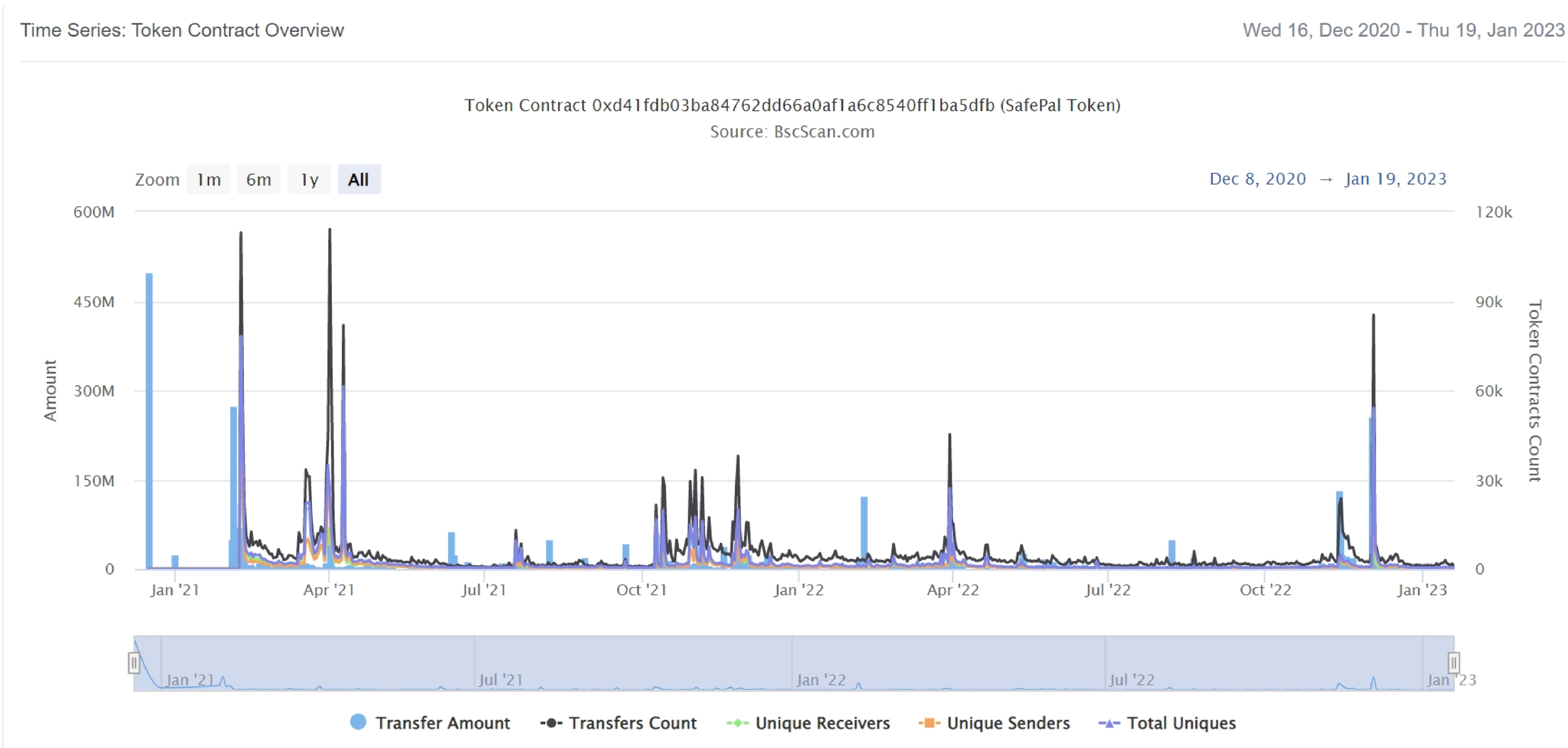
SafePal Token Top 20 Token Holders

(A total of 479,730,000.05 tokens held by the top 100 accounts from the total supply of 500,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Binance: Hot Wallet 20	185,721,298.8477226	37.1443%
2	Binance: Hot Wallet 22	66,200,000	13.2400%
3	0xbbfb5a79fb2833a111f4b3a9b5303847ae9c3f87	62,500,000	12.5000%
4	0x3966ff05ccf31ddd9da4f88f98127efa228aac89	50,000,000	10.0000%
5	0x9ace3985ebf65e6543c9566dff470bacc026e384	31,250,000	6.2500%
6	0x7fc3583e3c62d7bb1ba05418bb1522eaf8d1f9ec	14,475,129.329932663	2.8950%
7	0x942b294e59a8c47a0f7f20df105b082710f7c305	12,245,107.011771083161863588	2.4490%
8	0xe54b4cd7697a2b3dbc688d9779262d32d99dbd5e	10,416,666.6683367	2.0833%
9	0x59de6d3d1eba28be543f69180de0c07f4400e9ea	10,000,000	2.0000%
10	0x62be39480331469e2bad43500c975fa65e09c0f6	10,000,000	2.0000%
11	0x3da8a3c97703eb0cd7f27cbfd7b5796ed9a04bea	6,956,125	1.3912%
12	0xac28b542df013d7e996066f189b253e666c2a052	2,500,000	0.5000%
13	Binance: Hot Wallet 6	1,317,212.584778813278605088	0.2634%
14	FENG SHUI COIN: Deployer	1,183,067.686344955479765729	0.2366%
15	0x3058568088dd5206ec0f0670ac995979347329e2	1,135,315.24	0.2271%
16	Binance: Hot Wallet 7	906,862.40885718	0.1814%
17	0x2bc39605b157c0e47316918c297811b9018a125c	822,222.453828	0.1644%
18	0xc882b111a75c0c657fc507c04fbfcd2cc984f071	558,349.90386962291200516	0.1117%
19	0x26d67e8d001c8db46ef8ae6b5ce395019af88397	499,999.83	0.1000%
20	0xea0c911c7fe907ef7089166735f647317f109ef5	482,232.78698201	0.0964%

SafePal Token Distribution

SafePal Token Contract overview



Contract functions details

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

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

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

+ERC20 (Context, IERC20)
    - -[Pub] <constructor> #
    - -[Pub] name
    - -[Pub] symbol
    - -[Pub] decimals
    - -[Pub] totalSupply
    - -[Pub] balanceOf
    - -[Pub] transfer #
    - -[Pub] allowance
    - -[Pub] approve #
    - -[Pub] transferFrom #
    - -[Pub] increaseAllowance #
    - -[Pub] decreaseAllowance #
    - -[Int] _transfer #
    - -[Int] _mint #
    - -[Int] _burn #
    - -[Int] _approve #
    - -[Int] _setupDecimals #

Contract functions details

```
        -[Int] _beforeTokenTransfer #

+SafePalToken (ERC20)
        -[Pub] <constructor> #

($) = 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.	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 issue found.
- **Medium Severity Issues**
  No medium severity issue found.
- **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.

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.