HackSafe

Smart Contract Security Audit Report

loser coin

October 2022

Security Status

★★★☆☆

www.hacksafe.io

Audit Details

Audited project
loser coin

Deployer address
0x571816d7C43767C9549BbbD843b9C6e2C7560CAb

Client contacts
loser coin Team

Blockchain
Binance smart chain

Website
https://loserswap.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 loser coin to perform an audit of smart contracts:

- https://bscscan.com/address/0x843d4a358471547f51534e3e51fae91cb4dc3f28#code

The purpose of the audit was to achieve the following:

- 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 06.10.2022

Token Type	: BEP20
Contract name	: ERC20Template
Contract address	: 0x843D4a358471547f51534e3e51fae91cb4Dc3F28
Total supply	: 78,381,822,208.786638
Token ticker	: lowb
Decimals	: 18
Token holders	: 141,344
Transactions count	: 785,766
Compiler version	: v0.6.6+commit.6c089d02
Contract deployer address	: 0x571816d7C43767C9549BbbD843b9C6e2C7560CAb
Owner address	: No owner

Social profiles

Twitter Profile	: https://twitter.com/loser_coin
Telegram profile	: https://t.me/loser_coin
Coinmarketcap profile	: https://coinmarketcap.com/currencies/loser-coin/
Coingecko profile	: https://www.coingecko.com/en/coins/loser-coin/

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **“Poor Secure”**. This token contract does not contain owner control, which do make it fully decentralized as owner does not 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 1 critical, 0 high, 0 medium and 1 low and some very low-level issues.

loser coin Token Distribution

The top 100 holders collectively own 89.75% (70,350,840,031.73 Tokens) of loser coin | Token Total Supply: 78,381,822,208.79 Token | Total Token Holders: 141,344

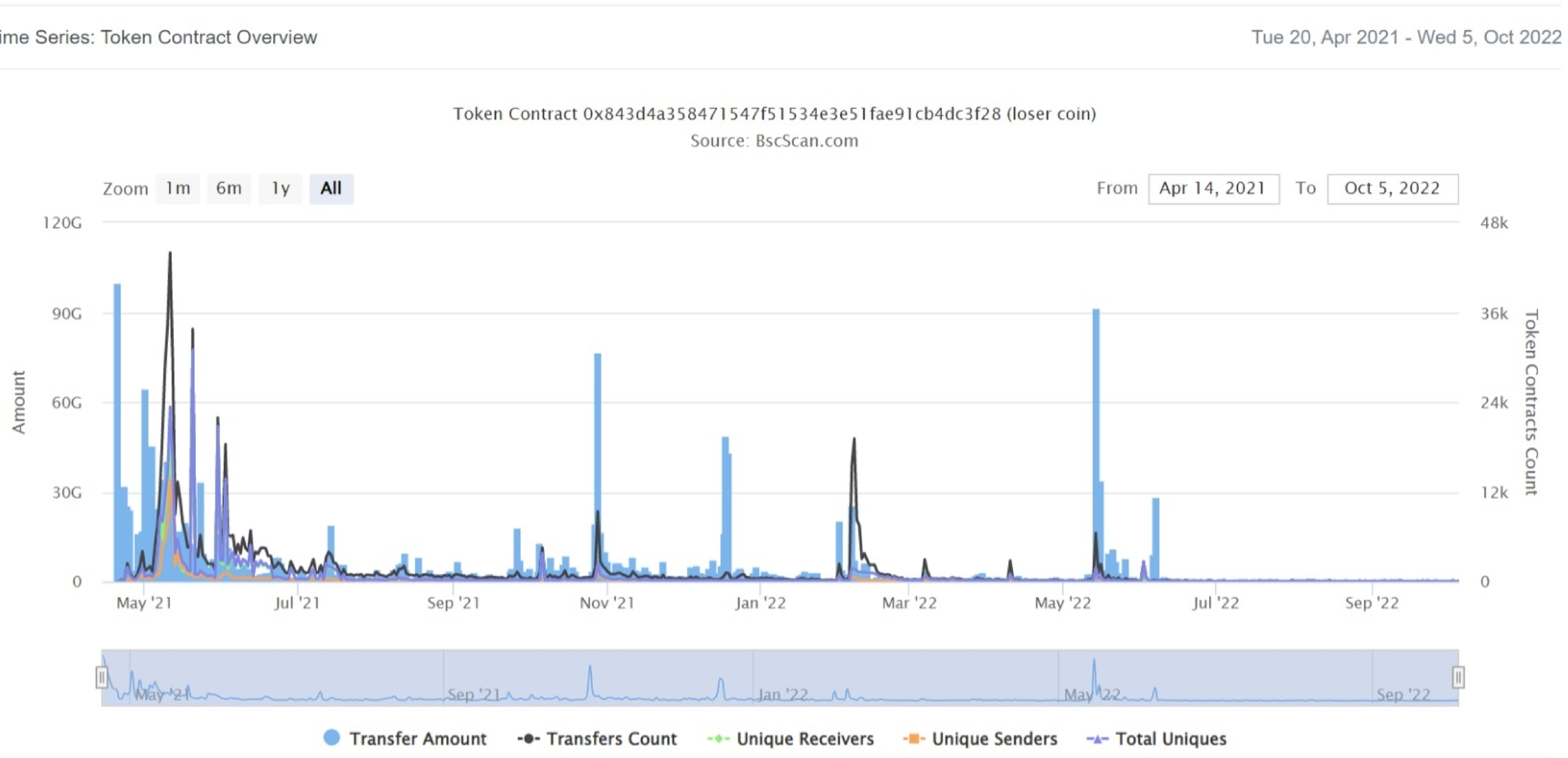
loser coin Top 20 Token Holders

(A total of 70,350,840,031.73 tokens held by the top 100 accounts from the total supply of 78,381,822,208.79 token)

Rank	Address	Quantity (Token)	Percentage
1	Null Address: 0x000...dEaD	27,266,938,803.192017306754718222	34.7873%
2	0x4b6080917a8333d5db16345642d9899e9a870d9f	25,018,023,437.947813222043981936	31.9181%
3	0x3642b52519ba81fd8a204b306d2369a0cc1bc612	2,623,213,189.009463798955710163	3.3467%
4	0x56e51f2c6a2bb85fba2b34c1aea42fe8990c4398	1,717,441,126.700419106653784057	2.1911%
5	0xfd6b98805708b346edfa4a6affeb629d4b05f540	1,472,162,448.893496892361277201	1.8782%
6	0xdc05ed497f43de362c7fb112fafabbd5ee8b48bd	1,107,067,461.134480861523176407	1.4124%
7	0x54e43e81e3662fe3bc7abb2db0fc0c67d8685f4e	1,003,039,485.873401247603645951	1.2797%
8	0xe648edf59808e494695442a5eefea282287b3227	1,002,983,690.796181526571597325	1.2796%
9	PancakeSwap V2: lowb-GAME	691,127,910.776630320004610507	0.8817%
10	0xb49036c84665dd97ac50d91894a1cbfddd52a971	472,548,810.917683924959833426	0.6029%
11	0x276ba586c1938a97e47425319d9d7f1ed9e43d11	413,588,146.477104536675099433	0.5277%
12	0xab450d37f5c8148f4125734c645f3e777a90f003	358,854,288.422738405769589936	0.4578%
13	0xb9a791a970bbe7321757c1d05289ab374b552aed	352,468,306.755197270714172663	0.4497%
14	0xff567291b421277922ee11e480d006381433b3e8	239,189,998.238759749285328044	0.3052%
15	0xfb7c0f98fb2ed350fde2d837f282767c83ef9b20	230,114,616.802960105413919782	0.2936%
16	0x308fa584f35690e8fae8b18814d67c6402417928	221,257,228.493257011373281024	0.2823%
17	0x0ad06effdd9c3b716206a1b871a0aa547ad39948	215,479,692.714540881911783719	0.2749%
18	Null Address: 0x000...001	214,963,509.117460581113506847	0.2743%
19	0x38e8865a8669e20f2cb736d9c3b4b22fef35eb93	185,000,000	0.2360%
20	0xc681d7bb73ff4238b124fd84b92c5d1721df2e70	180,430,988.376689947004793939	0.2302%

loser coin Token Distribution

loser coin Contract Overview



Contract functions details

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

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

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

+Pausable (Context)
    -[Int] <constructor>
    -[Pub] paused #
    -[Int] _pause #
      -modifiers: whenNotPaused
    -[Int] _unpause #
      -modifiers: whenPaused

+ERC20 (Context, IERC20)
    -[Pub] <constructor>
    -[Pub] name
    -[Pub] symbol
    -[Pub] decimals
    -[Pub] totalSupply
    -[Pub] balanceOf
    -[Pub] transfer #
    -[Pub] allowance

Contract functions details

-[Pub] approve #
-[Pub] transferFrom #
-[Pub] increaseAllowance #
-[Pub] decreaseAllowance #
-[Int] _transfer #
-[Int] _mint #
-[Int] _burn #
-[Int] _approve #
-[Int] _setupDecimals #
-[Int] _beforeTokenTransfer #

+ERC20Template (ERC20Pausable)
-[Pub] <constructor>
-[Pub] pause #
-modifiers: onlyPauser
-[Pub] unpause #
-modifiers: onlyPauser
-[Pub] changeUser #
-modifiers: onlyFactory
-[Pub] mint #
-modifiers: whenNotPaused, onlyOperator
-[Pub] burn #
-modifiers: whenNotPaused, onlyOperator

($) = 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.	Critical issue
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

One critical severity issue found.

1. Design logic.

- **Description**

  The contract has burn function which parameter is address and amount of token to be burn. Function describes that operator can burn anyone’s token without having allowance.

- **Recommendation**

  We advise you to check require in that function if the address passes in parameter have given allowance to operator to burn their tokens or not.

High Severity Issues

No high severity issues 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.