



Smart Contract Security Audit Report

GrimaceCoin

December 2022

Security Status



www.hacksafe.io



Audit Details



Audited project

GrimaceCoin



Deployer address

0xa6941b3e98bf846196b0a1250acfd079b4263b5



Client contacts

GrimaceCoin Team



Blockchain

Binance smart chain



Website

<https://www.grimacecoincrypto.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 GrimaceCoin to perform an audit of smart contracts:

- <https://bscscan.com/token/0xC6759a4Fc56B3ce9734035a56B36e8637c45b77E#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 understood to 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.

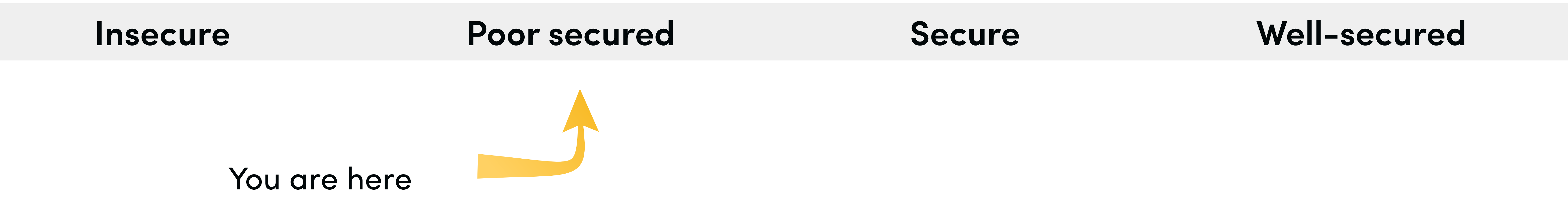
Contract Details

Token contract details for 29.12.2022

Token Type	: Charity
Contract name	: GrimaceCoin
Contract address	: 0xC6759a4Fc56B3ce9734035a56B36e8637c45b77E
Total supply	: 1,000,000
Token ticker	: Grimace
Decimals	: 18
Token Holders	: 6,896
Transactions count	: 187,175
Compiler version	: v0.7.4+commit.3f05b770
Contract deployer address	: 0xa6941b3e98bf846196b0a1250acfd079b4263b5
Owner address	: 0xa6941b3e98bf846196b0a1250acfd079b4263b5

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **“Poor Secure”**. This token contract does contain owner control, which do not make it fully decentralized



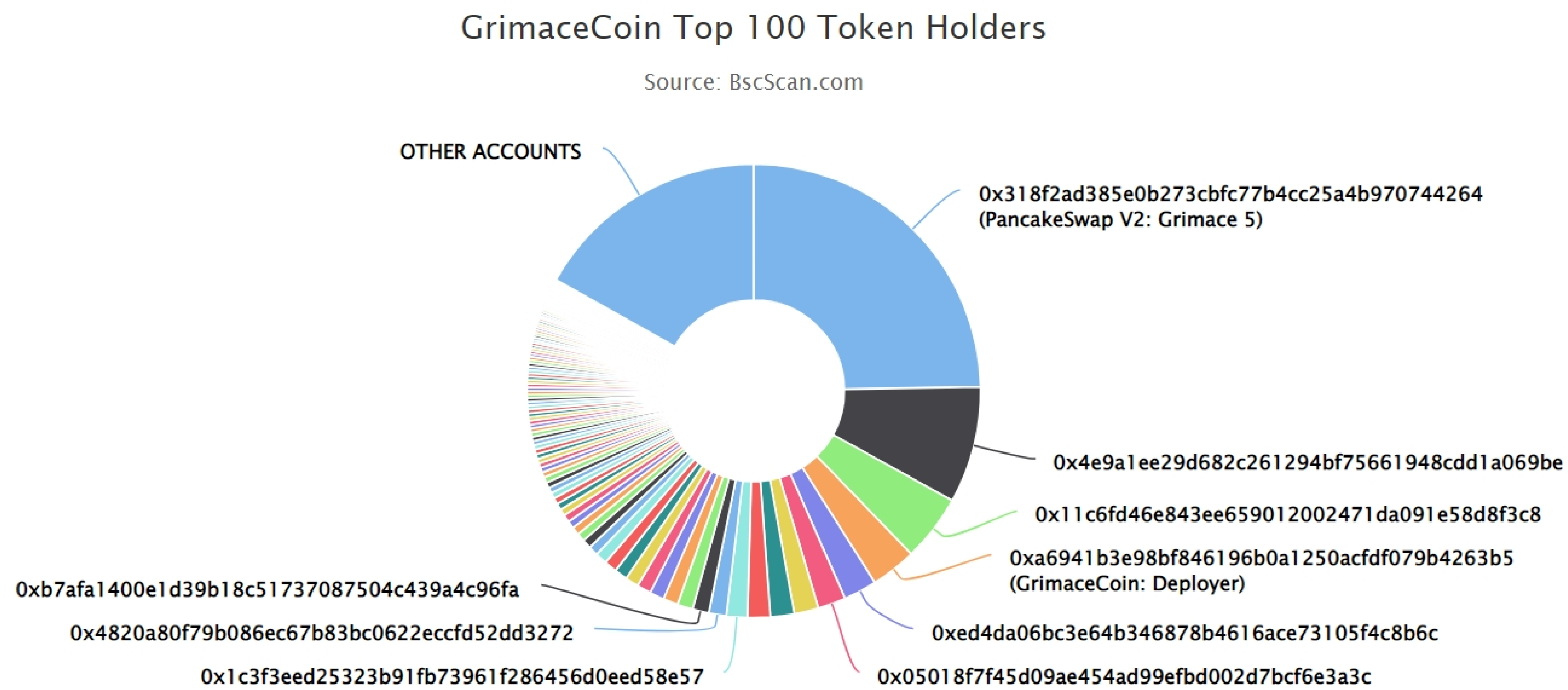
We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 0 high, 1 medium and 1 low.

GrimaceCoin Token Distribution




💡 The top 100 holders collectively own 83.15% (831,520.97 Tokens) of GrimaceCoin

💡 Token Total Supply: 1,000,000.00 Token | Total Token Holders: 6,896



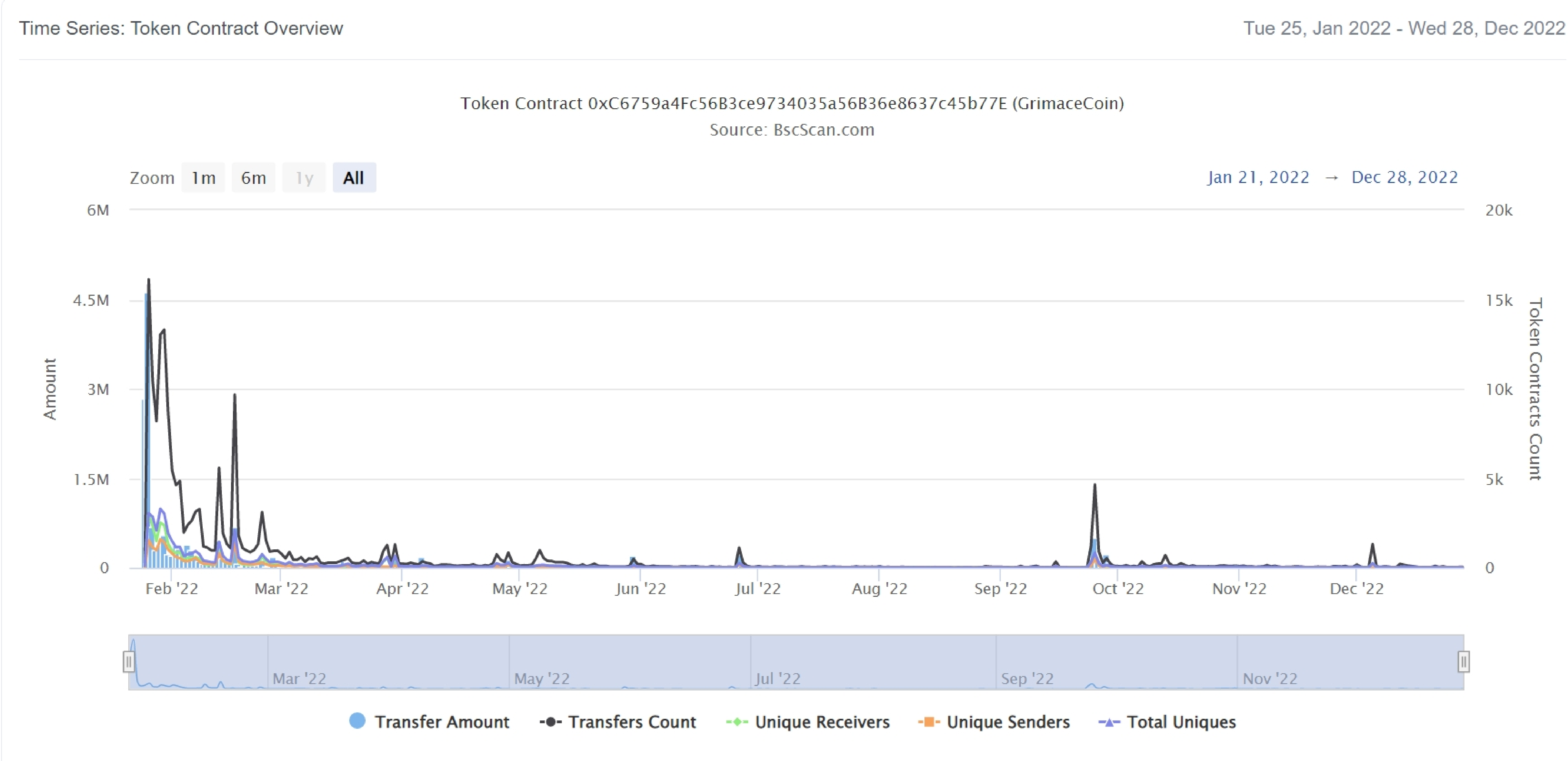
GrimaceCoin Top 20 Token Holders

(A total of 831,520.97 tokens held by the top 100 accounts from the total supply of 1,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	 PancakeSwap V2: Grimace 5	247,099.608857608958583649	24.7100%
2	 0x4e9a1ee29d682c261294bf75661948cdd1a069be	83,390.508823774885170406	8.3391%
3	 0x11c6fd46e843ee659012002471da091e58d8f3c8	47,451.80659318257933901	4.7452%
4	GrimaceCoin: Deployer	32,451.147349559935436029	3.2451%
5	0xed4da06bc3e64b346878b4616ace73105f4c8b6c	23,789.936634754793564109	2.3790%
6	0x05018f7f45d09ae454ad99efbd002d7bcf6e3a3c	19,906.548926899125037687	1.9907%
7	0xcbfe43b9c2bbb91718bee44f8127eb0d50ade483	17,270.321063197715659793	1.7270%
8	0x13b983139e0a69883088720c60f6e7767034a59d	17,069.499061653943735885	1.7069%
9	0x9a447aa3aa67557a3f2c69908ec0e9204f54dba0	16,085.720536934141321597	1.6086%
10	0x1c3f3eed25323b91fb73961f286456d0eed58e57	15,000.68040167351862952	1.5001%
11	0x4820a80f79b086ec67b83bc062eccfd52dd3272	12,383.139359906805315892	1.2383%
12	0xb7afa1400e1d39b18c51737087504c439a4c96fa	11,780.576973175584486949	1.1781%
13	0x38c835271ed6b3020e443fb21f2834aa095c514a	11,000	1.1000%
14	0x6afd3cc4856874cb4ef45cf6ddf490a67491401c	10,544.222316405849956593	1.0544%
15	0xe069b60434eaf2c27df131ca1f42bc7a272b4db6	10,397.728723435863335029	1.0398%
16	0x23267877b03e18910e0b65ad51af0fb49f38b936	10,001.691464892136350698	1.0002%
17	0x833c45ac49d2275f75d4ddd8c64735d4eafaef09	9,603.391345845523335997	0.9603%
18	0xe067f5b4984b14730e0524d377750fbc31775974	9,142.427734466931304383	0.9142%
19	0x232ee132e8b5dceace8e2ba8650f80d1b143dd1e	9,121.564165334588323236	0.9122%
20	0xa7d5f7b26ccb698ad68376b4e62c44e0b7965260	8,475.709947888740119745	0.8476%

GrimaceCoin Token Distribution

GrimaceCoin Contract Overview



Contract functions details

+[Lib] SafeMath

- [Int] add
- [Int] sub
- [Int] sub
- [Int] mul
- [Int] div
- [Int] div

+[Int] IBEP20

- [Ext] totalSupply
- [Ext] decimals
- [Ext] symbol
- [Ext] name
- [Ext] getOwner
- [Ext] balanceOf
- [Ext] transfer #
- [Ext] allowance
- [Ext] approve #
- [Ext] transferFrom #

+[Int] IDEXFactory

- [Ext] createPair #

+[Int] IDEXRouter

- [Ext] factory
- [Ext] factory
- [Ext] WETH
- [Ext] addLiquidity #
- [Ext] addLiquidityETH (\$)
- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
- [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens (\$)
- [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+[Int] IDividendDistributor

- [Ext] setDistributionCriteria #
- [Ext] setShare #
- [Ext] deposit (\$)
- [Ext] process #
- [Ext] claimDividend #

Contract functions details

+DividendDistributor (IDividendDistributor)

- [Pub] <Constructor> #
- [Ext] setDistributionCriteria #
 - modifiers: onlytoken
- [Ext] setShare #
 - modifiers: onlytoken
- [Ext] deposit (\$)
 - modifiers: onlytoken
- [Ext] process #
 - modifiers: onlytoken
- [Int] shouldDistribute
- [Int] distributeDividend #
- [Pub] getUnpaidEarnings
- [Int] getCumulativeDividends
- [Int] addShareholder #
- [Int] removeShareholder #
- [Ext] claimDividend #

+Auth

- [Pub] <Constructor> #
- [Pub] authorize #
 - modifiers: onlyowner
- [Pub] unauthorize #
 - modifiers: onlyowner
- [Pub] isOwner
- [Pub] isAuthorized
- [Pub] transferOwnership #
 - modifiers: onlyowner

+GrimaceCoin (IBEP20, Auth)

- [Pub] <Constructor> #
 - modifiers: auth
- [Ext] <Fallback >(\$)
- [Ext] name
- [Ext] symbol
- [Ext] decimals
- [Ext] totalSupply
- [Ext] getOwner
- [Ext] getCirculatingSupply

Contract functions details

- [Pub] balanceOf
- [Ext] allowance
- [Pub] approve #
- [Ext] approveMax #
- [Pub] claim #
- [Int] launched
- [Int] launch #
- [Ext] changeTxLimit #
 - modifiers: authorized
- [Ext] changeWalletLimit #
 - modifiers: authorized
- [Ext] changeRestrictWhales #
 - modifiers: authorized
- [Ext] changelsFeeExempt #
 - modifiers: authorized
- [Ext] changelsDividendExempt #
 - modifiers: authorized
- [Ext] changeFees #
 - modifiers: authorized
- [Ext] changeFeeReceivers #
 - modifiers: authorized
- [Ext] changeSwapBackSettings #
 - modifiers: authorized
- [Ext] changeDistributionCriteria #
 - modifiers: authorized
- [Ext] changeDistributorSettings #
 - modifiers: authorized
- [Ext] transfer #
- [Ext] transferFrom #
- [Ext] _transferFrom #
- [Ext] _basicTransfer #
- [Int] takeFee #
- [Pub] tradingStatus #
 - modifiers: onlyowner
- [Int] swapBack #
 - modifiers: locktheswap

(\$) = 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.	Medium issue
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

One medium severity issue found.

1. Time stamp dependency

- **Issue:**

This smart contract contain following functions swapBack, deposit, shouldDistribute, distributeDividend which uses **block.timestamp** means functions or contract can be manipulated by miners if they have some incentive to do so as miners can adjust the timestamp.

- **Recommendation**

It is advisable that Block timestamps should not be used for entropy or generating random numbers – i.e. they should not be the deciding factor (either directly or through some derivation) for changing an important state (if assumed to be random). This can be unnecessary if contracts aren't particularly concerned with miner manipulations of the block timestamp, but it is something to be aware of when developing contracts.

✓ 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 :

- GrimaceCoin Contract:
 - Owner Can Change Trading Status.
 - Authorized addresses can change the maximum transaction amount.
 - Authorized addresses can change maximum wallet amount restriction.
 - Authorized addresses can change restrictwhales value.
 - Authorized addresses can exclude from fees.
 - Authorized addresses can exclude from max transaction limit.
 - Authorized addresses can exclude from dividends.
 - Authorized addresses can change fees.
 - Authorized addresses can change fee receivers.
 - Authorized addresses can change distribution criteria and distribution GAS

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.

- authorize
- unauthorize
- transferOwnership
- tradingStatus
- changeTxLimit
- changeWalletLimit
- changeRestrictWhales
- changeFeeExempt
- changeTxlimitExempt
- changeDividEndexempt
- changeFees
- changeFeeReceivers
- changeSwapBacksettings
- changeDistributionCriteria
- changeDistributorSettings

Conclusion

Smart contract contains medium and 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.