HackSafe

Smart Contract Security Audit Report

CEN

October **2022**

Security Status

★ ★ ★ ★ ⯪

www.hacksafe.io

Audit Details

Audited project
CEN

Deployer address
0xC3cFA2559cB1Df2d884b7E7554fD8E97465ce65E

Client contacts
CEN Team

Blockchain
Ethereum

Website
Not provided by owner

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 CEN to perform an audit of smart contracts:

- https://etherscan.io/token/0x0bc61dded5f6710c637cf8288eb6058766ce1921#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 11.10.2022

Token Type	: ERC20
Contract name	: CENToken
Contract address	: 0x0bC61DdED5F6710c637cf8288Eb6058766ce1921
Total supply	: 904,911,274.223577000000000001
Token ticker	: CEN
Decimals	: 18
Token holders	: 1,915
Transactions count	: 8,475
Compiler version	: v0.4.24+commit.e67f0147
Contract deployer address	: 0xC3cFA2559cB1Df2d884b7E7554fD8E97465ce65E
Owner address	: 0xC3cFA2559cB1Df2d884b7E7554fD8E97465ce65E

Social profiles

Telegram profile	: https://t.me/CoinsuperEx
Coinmarketcap profile	: https://coinmarketcap.com/currencies/coinsuper-ecosystem-network/
Coingecko profile	: https://www.coingecko.com/

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 2 low and some very low-level issues. These issues are not critical ones.

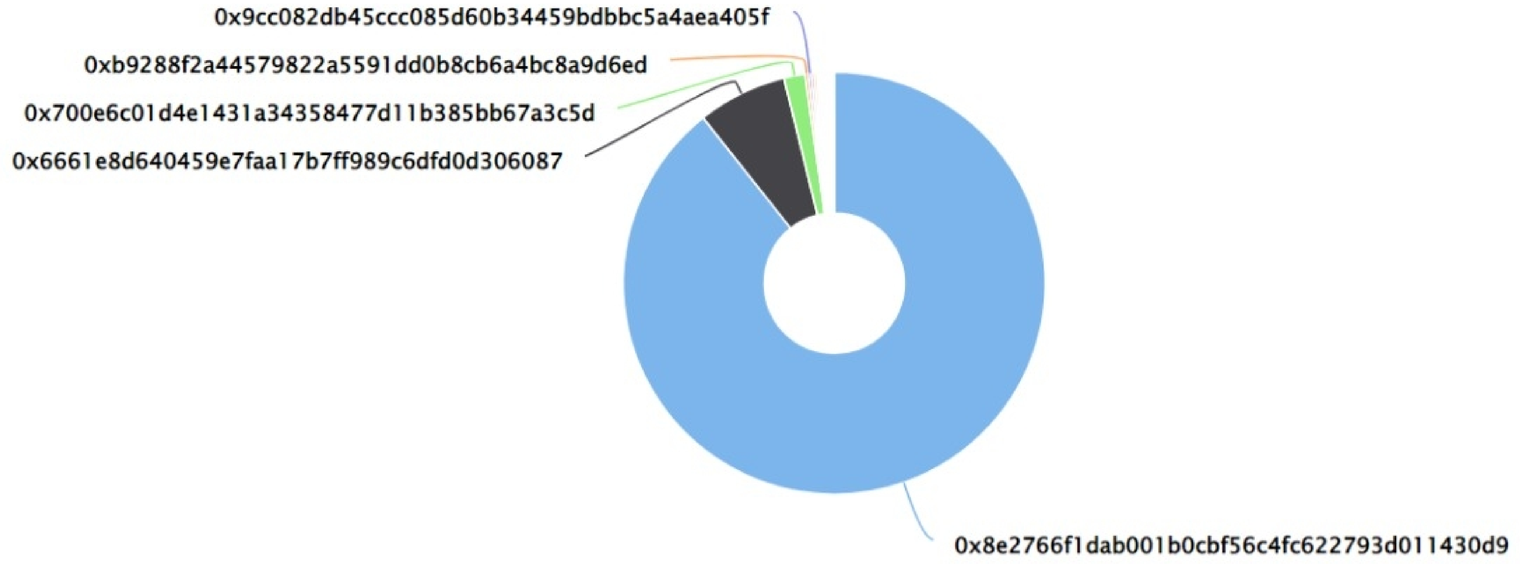
CEN Token Distribution

The top 100 holders collectively own 99.93% (904,246,346.79 Tokens) of CEN

Token Total Supply: 904,911,274.22 Token | Total Token Holders: 1,915

CEN Top 100 Token Holders

Source: Etherscan.io



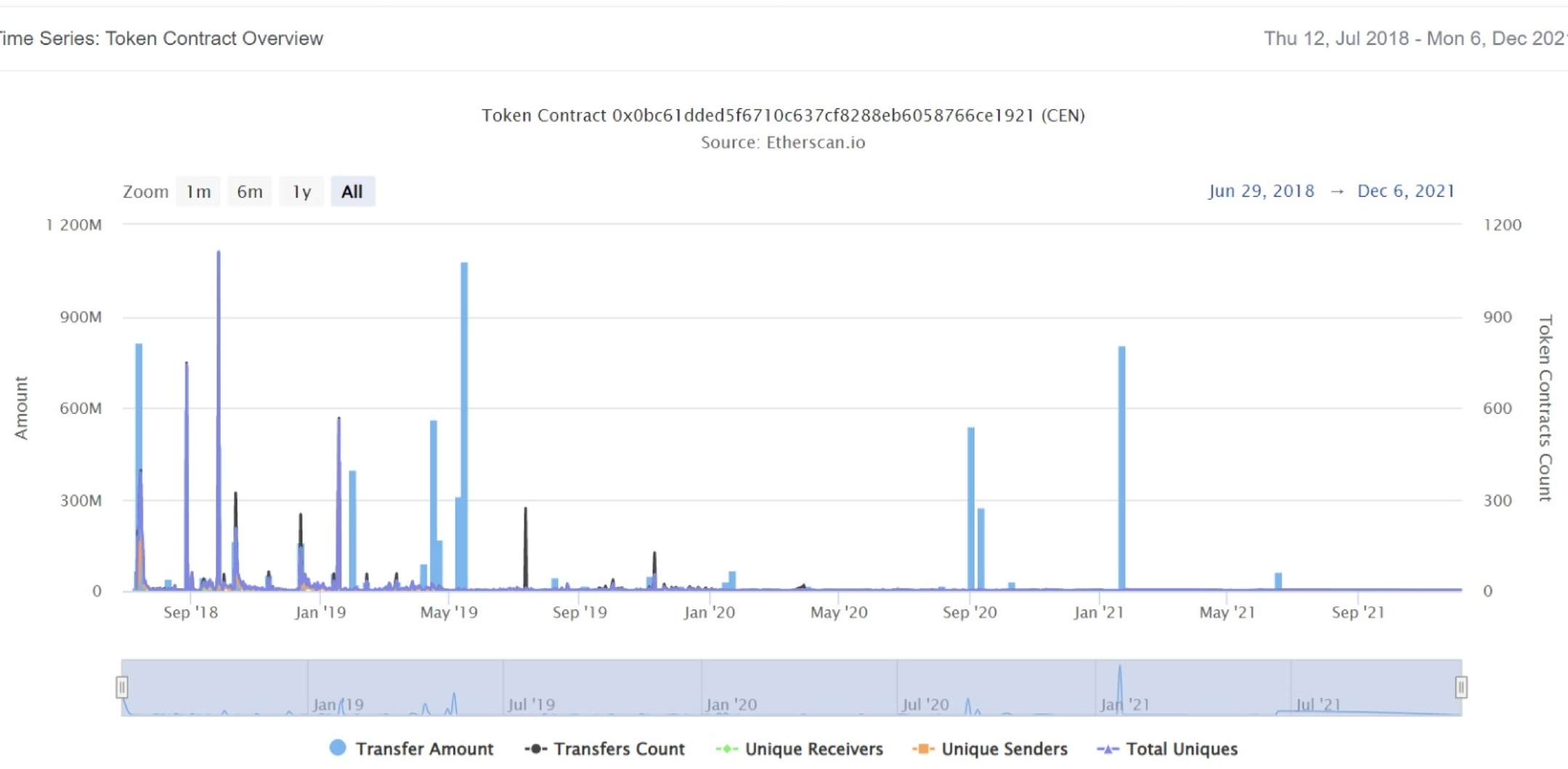
CEN Top 20 Token Holders

(A total of 904,246,346.79 tokens held by the top 100 accounts from the total supply of 904,911,274.22 token)

Rank	Address	Quantity (Token)	Percentage
1	0x8e2766f1dab001b0cbf56c4fc622793d011430d9	808,343,352.191017	89.3285%
2	0x6661e8d640459e7faa17b7ff989c6dfd0d306087	61,994,454.496739701	6.8509%
3	0x700e6c01d4e1431a34358477d11b385bb67a3c5d	14,583,334	1.6116%
4	0xb9288f2a44579822a5591dd0b8cb6a4bc8a9d6ed	2,409,355	0.2663%
5	0x9cc082db45ccc085d60b34459bdbbc5a4aea405f	1,999,997	0.2210%
6	0x5d8ce5b0c26d7e2b50724cb48174be6fd6e61670	1,874,980	0.2072%
7	0x518793bc80c30faeff7a7b5bd3115a22256aaea5	1,770,125.536321	0.1956%
8	0x8aae50113d811edf6653ee28a779cdbd2a783707	1,250,000	0.1381%
9	0x40337107580fede4d8d72677901a6407b416223e	1,169,968.136816	0.1293%
10	0x40652360d6716dc55cf9aab21f3482f816cc2cbd	868,058.640814	0.0959%
11	0x281f0991ba093810d7cfc2db5ef9b0593f3ad58d	825,000	0.0912%
12	0x0845196cb7ef2cf0972af18ddc9f1147fe3086aa	769,550	0.0850%
13	0xed0594a005278919179e92d41353b4badac2630a	586,500	0.0648%
14	0x6e188897b8730937b06fc2547eace1b75bf9219f	577,500	0.0638%
15	BitUN.io 1	416,826.330453489	0.0461%
16	0x175d58a0e5df2d1c3cf84a513a64e7484137acc0	312,460	0.0345%
17	0xde636b433d3968cfdecab4c236db03f7df133580	275,000	0.0304%
18	0x739973a78bf8360274fbe5596bb5fa6eaa64bd2d	245,040	0.0271%
19	0xa9228b632c9b3c1b4fafb1bb8c3681ffae65262e	241,158.364814	0.0266%
20	0x479d182f5afee20be4551e93676636776bb4eba3	195,630	0.0216%

CEN Token Distribution

CEN Contract Overview



Contract functions details

+ SafeMath
    -[Int] safeMul
    -[Int] safeDiv
    -[Int] safeSub
    -[Int] safeAdd
    -[Int] safeMulWithPresent
    -[Int] judgement

+ CENAuth
    -[Pub] < constructor>
    -[Pub] setOwner #
      -modifiers: onlyOwner

+ CENStop (CENAuth)
    -[Pub] _status
    -[Pub] stop #
      -modifiers: onlyOwner
    -[Pub] start #
      -modifiers: onlyOwner

+ Token (SafeMath)
    -[Pub] balanceOf
    -[Pub] transfer
    -[Pub] transferFrom
    -[Pub] approve
    -[Pub] allowance
    -[Pub] burn
    -[Pvt] frozenCheck
    -[Pub] freezeAccount

+StandardToken (Token ,CENStop)
    -[Pub] transfer #
      -modifiers: stoppable
    -[Pub] transferFrom #
      -modifiers: stoppable
    -[Pub] balanceOf
    -[Pub] approve #
      -modifiers: stoppable
    -[Pub] allowance

Contract functions details

-[Pub] burn #
  -modifiers: stoppable, onlyOwner
-[Pvt] frozenCheck #
-[Pub] freezeAccount #
  -modifiers: onlyOwner

+CENToken (StandardToken)

  -[Pub] <constructor>

  -[Pub] stoppable

($) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Low issue
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 issue found.

Low Severity Issues

Two low severity issue found.

1. Unlocked Compiler Version.

- **Description**

  The contract utilizes an unlocked compiler version. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

- **Recommendation**

  It is advisable that the compiler version is alternatively locked at the lowest version possible so that the contract can be compiled. For example, for version 0.4.24 the contract should contain the following line:

  pragma solidity 0.4.24;

2. Too 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 :

- CEN Contract:
    - Owner can set owner.
    - Owner can stop and start transfer, approve, burn tokens.
    - Owner can burn.
    - Owner can freeze accounts.

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. Following are Admin functions:

- Setowner
- Stop
- Start
- Burn
- Freezeaccount

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.