



# Smart Contract Security Audit Report

---

## QUINADS

September 2022

Security Status



[www.hacksafe.io](https://www.hacksafe.io)



# Audit Details



## Audited project

QUINADS



## Deployer address

0x61eBCC762976e675c47021A9330c08993Bd66A7f



## Client contacts

QUINADS Team



## Blockchain

Ethereum



## Website

<https://quinads.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 QUINADS to perform an audit of smart contracts:**

- <https://etherscan.io/token/0x86e44543164d9b97b14ef7f6f3ab7ba670cab346#code>

**The purpose of the audit was to achieve the**

- 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 16.09.2022

Token Type	: ERC20
Contract name	: QuinadsToken
Contract address	: 0x86E44543164D9b97B14ef7f6f3aB7bA670CAB346
Compiler version	: v0.4.24+commit.e67f0147
Total supply	: 20,000,000,000
Token ticker	: QUIN
Decimals	: 18
Token holders	: 407
Transactions count	: 2,599
Contract deployer address	: 0x61eBCC762976e675c47021A9330c08993Bd66A7f
Owner address	: 0x61eBCC762976e675c47021A9330c08993Bd66A7f



# Social profiles

Twitter Profile	: <a href="https://twitter.com/QuinAdsToken">https://twitter.com/QuinAdsToken</a>
Telegram profile	: <a href="https://t.me/QuinAdsTokenGlobal">https://t.me/QuinAdsTokenGlobal</a>
Coinmarketcap profile	: <a href="https://coinmarketcap.com/currencies/quinads/">https://coinmarketcap.com/currencies/quinads/</a>
Coingecko profile	: <a href="https://www.coingecko.com/en/coins/quinads/">https://www.coingecko.com/en/coins/quinads/</a>

# 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	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, 1 medium and 2 low and some very low-level issues. These issues are not critical ones.



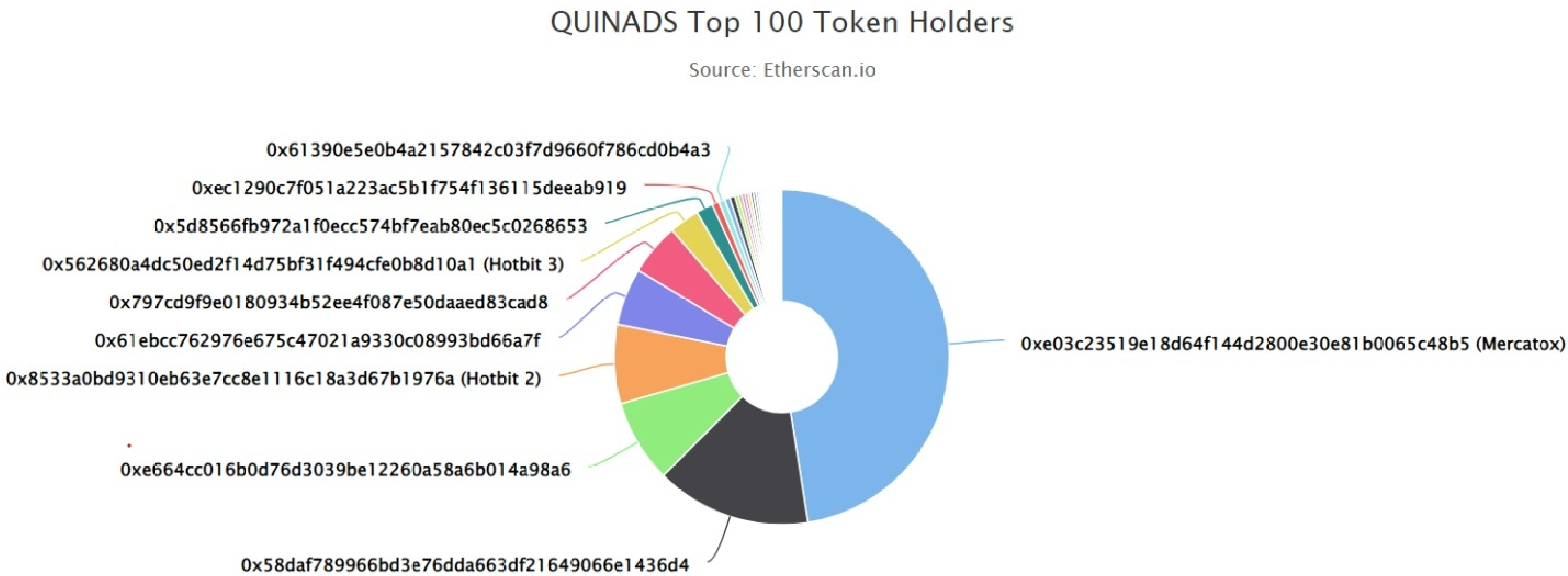
# QuinadsToken Token Distribution

 The top 100 holders collectively own 99.92% (19,983,765,477.60 Tokens) of QUINADS

 Token Total Supply: 20,000,000,000.00 Token


|

Total Token Holders: 407



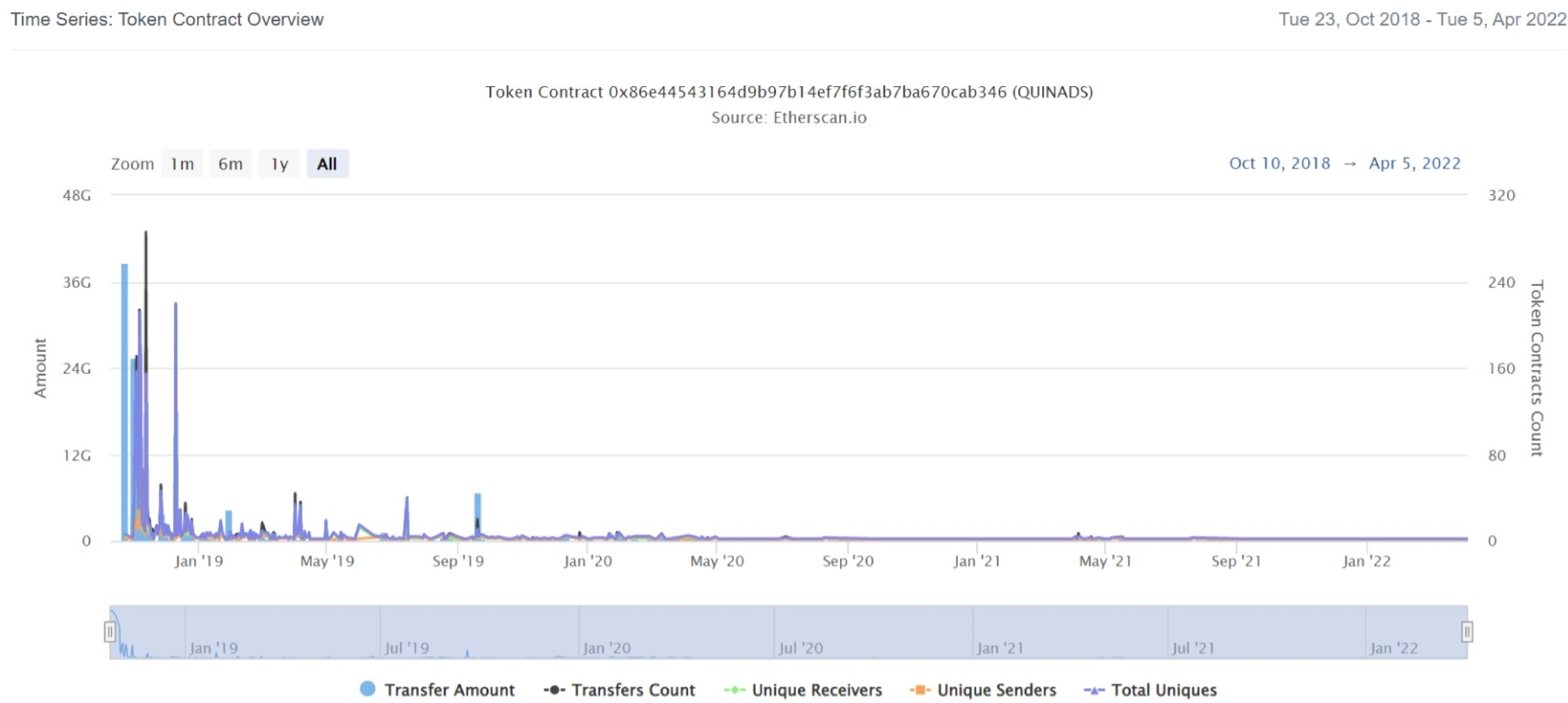
## QuinadsToken Top 20 Token Holders

(A total of 19,983,765,477.60 tokens held by the top 100 accounts from the total supply of 20,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Mercatox	9,500,492,104.111182913700158531	47.5025%
2	0x58daf789966bd3e76dda663df21649066e1436d4	3,000,000,000	15.0000%
3	0xe664cc016b0d76d3039be12260a58a6b014a98a6	1,600,000,000	8.0000%
4	Hotbit 2	1,523,493,390	7.6175%
5	0x61ebcc762976e675c47021a9330c08993bd66a7f	1,105,000,888.966786125190466222	5.5250%
6	0x797cd9f9e0180934b52ee4f087e50daaed83cad8	1,000,000,000	5.0000%
7	Hotbit 3	582,302,315.460243605250001916	2.9115%
8	0x5d8566fb972a1f0ecc574bf7eab80ec5c0268653	326,126,708.619338491000126194	1.6306%
9	0xec1290c7f051a223ac5b1f754f136115deeab919	136,849,999	0.6842%
10	0x61390e5e0b4a2157842c03f7d9660f786cd0b4a3	120,022,584.53125	0.6001%
11	0xb5a9fdd9cd0fe8393068ac1164fde1d940932ed7	100,000,000	0.5000%
12	0xc91795a59f20027848bc785678b53875934792a1	99,680,522	0.4984%
13	0x1d2400bad251eef1fa9aa984e1817dd9965d4366	70,000,000	0.3500%
14	0x7d716681be4b394f61b57512efb8a576ad78c743	68,250,000	0.3413%
15	0xa1674b743282ee6d80121a422fec511d09c05658	59,612,000	0.2981%
16	0x2513b92148f43d98a60c6673c1aee496f6ebf7f2	55,381,790.777624575200016968	0.2769%
17	0x821428bc3050edef92dfcf0519bce2f8e283f01d	54,612,000	0.2731%
18	 Globex Token	52,000,000	0.2600%
19	0x0dae0ff53cef3ce5ded45945bcf0d42c69ad55fd	52,000,000	0.2600%
20	0x30ce6f10a617fb8c66b93817adce1379853a7b13	50,964,208	0.2548%

# QuinadsToken Token Distribution

## QuinadsToken Contract Overview





# Contract functions details

## +[Lib] SafeMath

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

## +Ownable

- [Pub] <constructor>
- [Pub] renounceOwnership #
  - modifiers: onlyOwner
- [Pub] transferOwnership #
  - modifiers: onlyOwner
- [Int] \_transferOwnership #

## +Pausable (Ownable)

- [Pub] pause #
  - modifiers: onlyOwner whenNotPaused
- [Pub] unpause #
  - modifiers: onlyOwner whenPaused

## + ERC20Basic

- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] transfer

## + BasicToken (ERC20Basic)

- [Pub] totalSupply
- [Pub] transfer #
- [Pub] balanceOf

## + BurnableToken (BasicToken)

- [Pub] burn #
- [Int] \_burn

## + ERC20 (ERC20Basic)

- [Pub] allowance
- [Pub] transferFrom
- [Pub] approve

## + DetailedERC20 (ERC20)

- [Pub] <constructor> #

# Contract functions details

## + StandardToken (ERC20, BasicToken)

- [Pub] transferFrom #
- [Pub] approve #
- [Pub] allowance
- [Pub] increaseApproval #
- [Pub] decreaseApproval #

## + PausableToken (StandardToken, Pausable)

- [Pub] transfer #
  - modifiers: whenNotPaused
- [Pub] transferFrom #
  - modifiers: whenNotPaused
- [Pub] approve #
  - modifiers: whenNotPaused
- [Pub] increaseApproval #
  - modifiers: whenNotPaused
- [Pub] decreaseApproval #
  - modifiers: whenNotPaused

## +StandardBurnableToken (BurnableToken, StandardToken)

- [Pub] burnFrom #

## +QuinadsToken (DetailedERC20, StandardToken, Ownable, PausableToken, StandardBurnableToken)

- [Pub] <constructor>
- [Pub] closeICO #
  - modifiers: onlyOwner
- [Pub] addAirdropSupply #
  - modifiers: onlyOwner, icoFinished
- [Int] \_addAirdropSupply #
- [Int] doAirdrop #
- [Pub] adminClaimAirdrop #
  - modifiers: onlyOwner icoFinished canDistrAirdrop
- [Pub] adminClaimAirdropMultiple #
  - modifiers: onlyOwner icoFinished canDistrAirdrop

(\$) = 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.	Medium issue
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

One medium severity issues found.

### 1. Out of gas limit.

- **Description**

The smart contract has functions which has used for loops adminClaimAirdropMultiple. The parameters of these functions is array of addresses and unit. Large length of array

- **Recommendation**

It is advisable to either remove for loop or use smaller length of array to avoid the gas limit error.

## ✔ Low Severity Issues

Two low severity issue found.

### 1. Too old compiler version.

- **Too old compiler version.**

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.

### 2. 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.

# Security Issues

- **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;
```



# Centralization

## Owner Privileges :

- QUINADS Contract:
  - Admin can claim airdrop.
  - Admin can close ICO.
  - Admin can renounce and transfer ownership.
  - Admin can add airdrop supply.
  - Admin can pause and unpaue transfers.

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:

- Adminclaimairdrop
- Adminclaimairdropmultiple
- Transferownership
- Renounceownership
- Addairdropsupply
- Closeico
- Pause
- Unpause

# Conclusion

Smart contract contains low and medium 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.