



Smart Contract Security Audit Report

ZooToken

July 2022

Security Status



www.hacksafe.io



Audit Details



Audited project

ZooToken



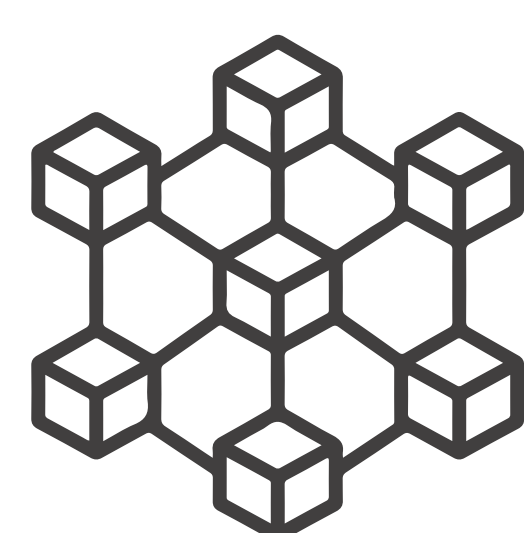
Deployer address

0xA63C637E369021F737B3bCF06A5743d6C2DdAB6c



Client contacts

ZooToken



Blockchain

Avalanche



Website

<https://zookeeper.finance/>

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 ZooToken to perform an audit of smart contract:

- <https://snowtrace.io/address/0x1B88D7aD51626044Ec62eF9803EA264DA4442F32#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 12.07.2022

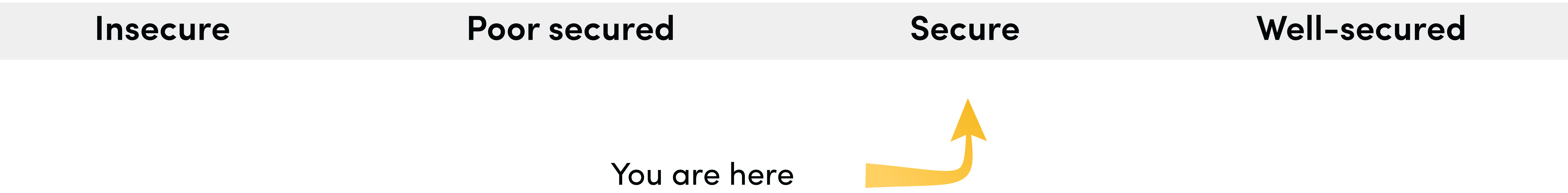
Token Type	: ERC20
Contract name	: ZooToken
Contract address	: 0x1B88D7aD51626044Ec62eF9803EA264DA4442F32
Compiler version	: v0.6.12+commit.27d51765
Total supply	: 33,075,794
Token Ticker	: ZOO
Decimals	: 18
Token Holders	: 399
Top 100 token holder's dominance	: 99.84%
Transactions count	: 28,182
Contract deployer address	: 0xA63C637E369021F737B3bCF06A5743d6C2DdAB6c
Owner address	: 0x1aC6332f1f1892B49Fb26aD1934F74F4Cd8C9dB9

Social profiles

Twitter Profile	: https://twitter.com/ZooEcosystem
Instagram Profile	: https://instagram.com/ZooEcosystem/
Reddit Profile	: https://www.reddit.com/r/ZooEcosystem/
Telegram Profile	: https://t.me/ZooEcosystem

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.



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.

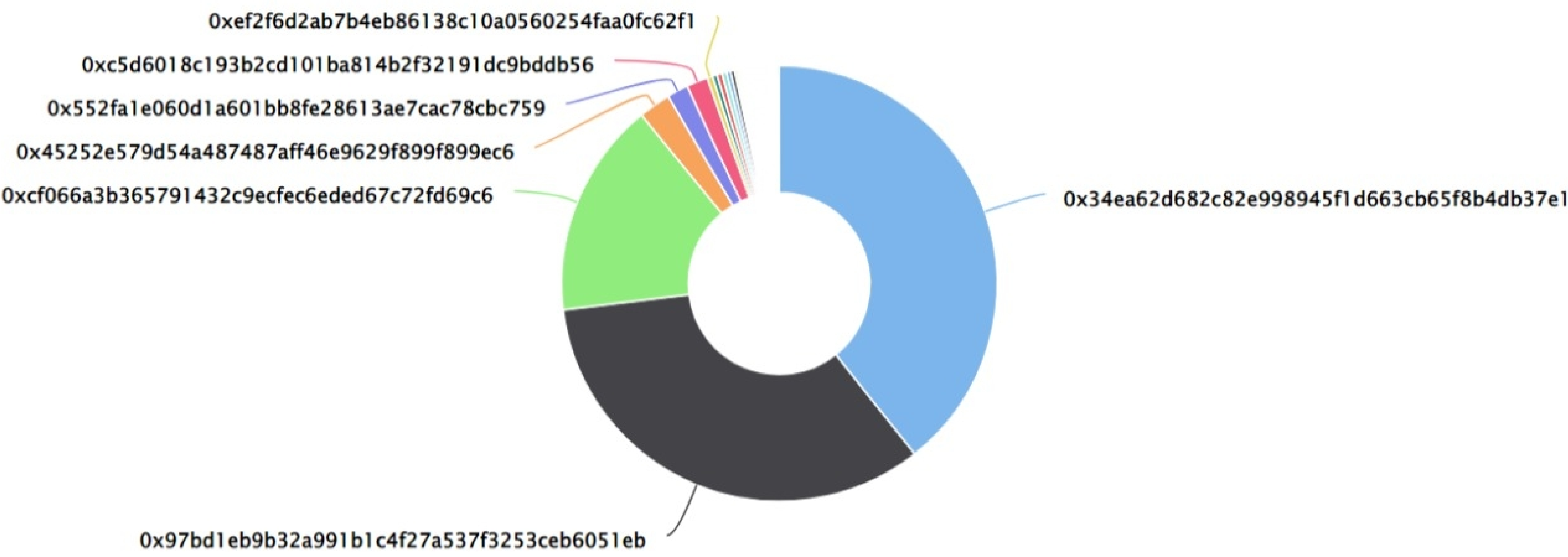
ZooToken Token Distribution

💡 The top 100 holders collectively own 99.84% (33,021,373.93 Tokens) of ZooToken

💡 Token Total Supply: 33,075,794.09 Token | Total Token Holders: 399




ZooToken Top 100 Token Holders

Source: snowtrace.io



ZooToken Token Top 20 Token Holders

(A total of 33,021,373.93 tokens held by the top 100 accounts from the total supply of 33,075,794.09 token)

Rank	Address	Quantity (Token)	Percentage
1	 0x34ea62d682c82e998945f1d663cb65f8b4db37e1	13,005,689.523002728487191761	39.3209%
2	 0x97bd1eb9b32a991b1c4f27a537f3253ceb6051eb	11,163,235.833799057527013092	33.7505%
3	 0xc066a3b365791432c9ecfec6eded67c72fd69c6	5,296,237.213345002184090815	16.0124%
4	0x45252e579d54a487487aff46e9629f899f899ec6	791,551.105613160437369097	2.3931%
5	0x552fa1e060d1a601bb8fe28613ae7cac78cbc759	522,972.483367896262421454	1.5811%
6	0xc5d6018c193b2cd101ba814b2f32191dc9bddb56	522,972.483367896262421454	1.5811%
7	0xef2f6d2ab7b4eb86138c10a0560254faa0fc62f1	132,706.188932305552781514	0.4012%
8	0xd0febb01b5ee65382c902f04d2cafe87189160f2	117,765.646121315525074027	0.3560%
9	0x7521eda00e2ce05ac4a9d8353d096ccb970d5188	117,765.59435861290245242	0.3560%
10	0x2f8283f97ce11c92c76d1beccf77a0f44d4ae7c9	117,765.592367896262421454	0.3560%
11	0x00000046dd6dd4a5bfc26d9371217119435a730e	100,000	0.3023%
12	0x980634c6b5d1967603b8e03c401611e15a2700d2	95,000	0.2872%
13	0xbb93dbfc900ce778cac47ae6671c2c328e913681	55,765.513599178633596267	0.1686%
14	0xe09efde4154343224fcf87aa1e720b7236b13cf	52,877.611114970315347223	0.1599%
15	0x971ba2f336b96ef56384befcec0122151bad950b	43,169.295586308643609829	0.1305%
16	0xfd911f754b58c0717a5b63fc79e2f0e175c03956	41,909.426025009933817922	0.1267%
17	0x393ee6c5591331dfdce93aab3731a3eb8e211f43	41,159.408184840071726285	0.1244%
18	0xe814fcc032a29182b19799e7b71b18a156b59c95	40,000	0.1209%
19	0x2200ef38d07f35e4fd21f4f305be3d158d481a7f	33,848.986836260149183226	0.1023%
20	0xc7f0bd18aae3fde423527212bc96618b0112f2d7	32,829.544689428355445077	0.0993%

Contract functions details

+ [Lib] SafeMath

- [Int] tryAdd
- [Int] trySub
- [Int] tryMul
- [Int] tryDiv
- [Int] tryMod
- [Int] add
- [Int] sub
- [Int] mul
- [Int] div
- [Int] mod
- [Int] sub
- [Int] div
- [Int] mod

+ [Int] IERC20

- [Ext] totalSupply
- [Ext] balanceOf
- [Ext] transfer
- [Ext] allowance
- [Ext] approve
- [Ext] transferFrom

+ Context

- [Int] _msgSender
- [Int] _msgData

+ Ownable (Context)

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

+ ERC20 (Context, IERC20)

- [Pub] < constructor>
- [Pub] name
- [Pub] symbol
- [Pub] decimals

Contract functions details

- [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 #
- [Int] _beforeTokenTransfer #

+ ZooToken (ERC20, Ownable)

- [Pub] mint #
- modifiers: onlyOwner
- [Pub] burn #

(\$) = 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 issue 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 $\geq 0.6.0 < 0.8.0$ the contract should contain the following line:

```
pragma solidity 0.6.12;
```

2. Too old compiler version.

• Description

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.

Centralization

Owner privileges :

- ZooToken Contract:
 - Owner can remove and transfer ownership.
 - Owner can mint new tokens.

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 but smart contract ownership has been renounced. Following are Admin functions functions:

- Transferownership
- Renounceownership
- Mint

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.