



# Smart Contract Security Audit Report

---

## SeedOn

November 2022

Security Status



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



# Audit Details



## Audited project

SeedOn



## Deployer address

0xD4868c3Db0FBA91aAEBFEDaBaF06975c729ED37a



## Client contacts

SeedOn Team



## Blockchain

Binance smart chain



## Website

<https://seedon.io/>



# 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 SeedOn to perform an audit of smart contracts:

- <https://bscscan.com/token/0x7672843C25c5ba11191dA8dA40C0881D7E77D9E0#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 18.11.2022

Token Type	: DEFI
Contract name	: SeedOn
Contract address	: 0x7672843C25c5ba11191dA8dA40C0881D7E77D9E0
Total supply	: 350,000,000
Token Ticker	: SEON
Decimals	: 18
Token Holders	: 1,667
Transactions count	: 40,933
Compiler version	: v0.8.7+commit.e28d00a7
Contract deployer address	: 0xD4868c3Db0FBA91aAEBFEDaBaF06975c729ED37a
Owner address	: No owner



# Social profiles

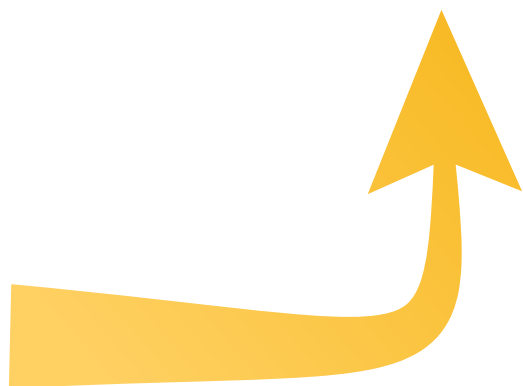
Facebook profile	: <a href="https://facebook.com/seedOn-103498612114380">https://facebook.com/seedOn-103498612114380</a>
Twitter profile	: <a href="https://twitter.com/seedonventure">https://twitter.com/seedonventure</a>
Telegram profile	: <a href="https://t.me/SeedOnVenture">https://t.me/SeedOnVenture</a>
linkedin profile	: <a href="https://www.linkedin.com/company/seedon">https://www.linkedin.com/company/seedon</a>

# Audit Summary

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

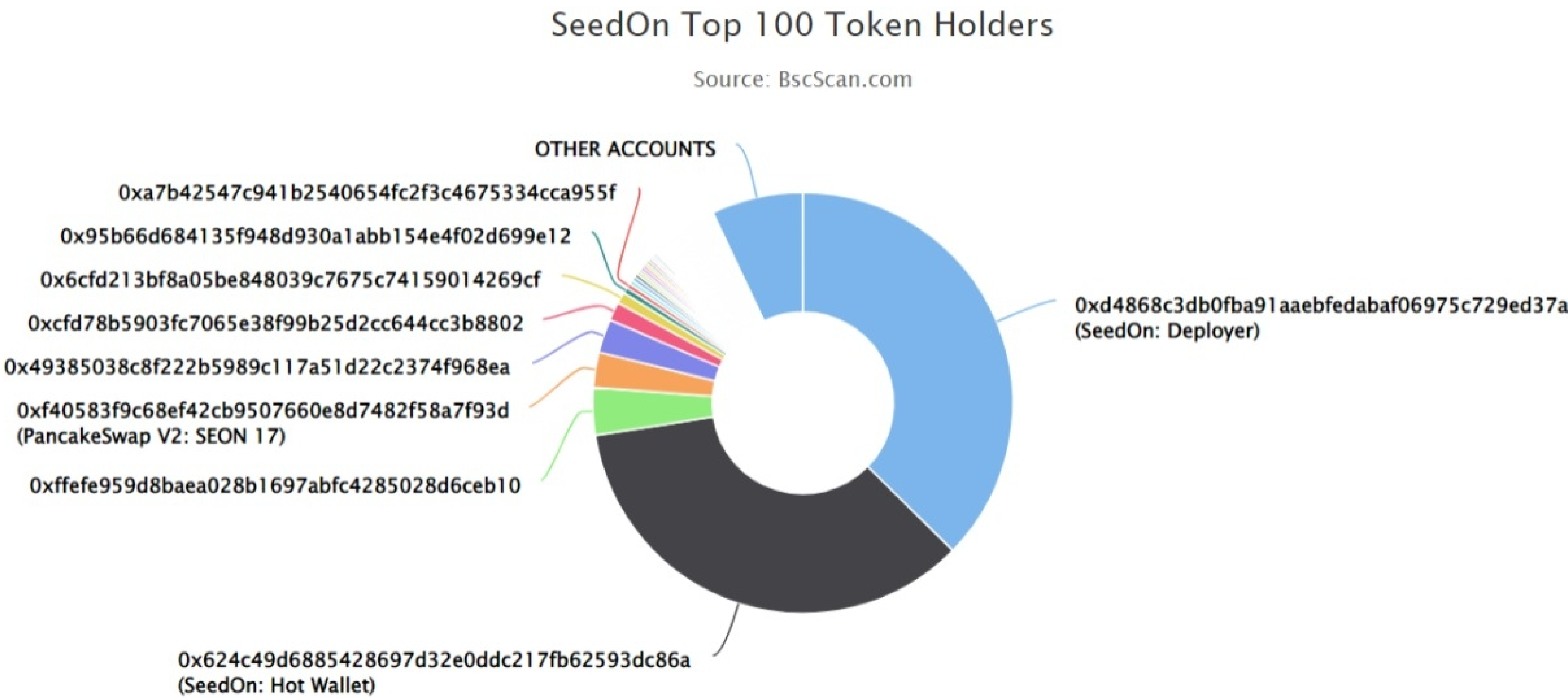
We found 0 critical, 0 high, 0 medium and 1 low.



# SeedOn Distribution


💡 The top 100 holders collectively own 92.96% (325,354,892.39 Tokens) of SeedOn

💡 Token Total Supply: 350,000,000.00 Token | Total Token Holders: 1,667



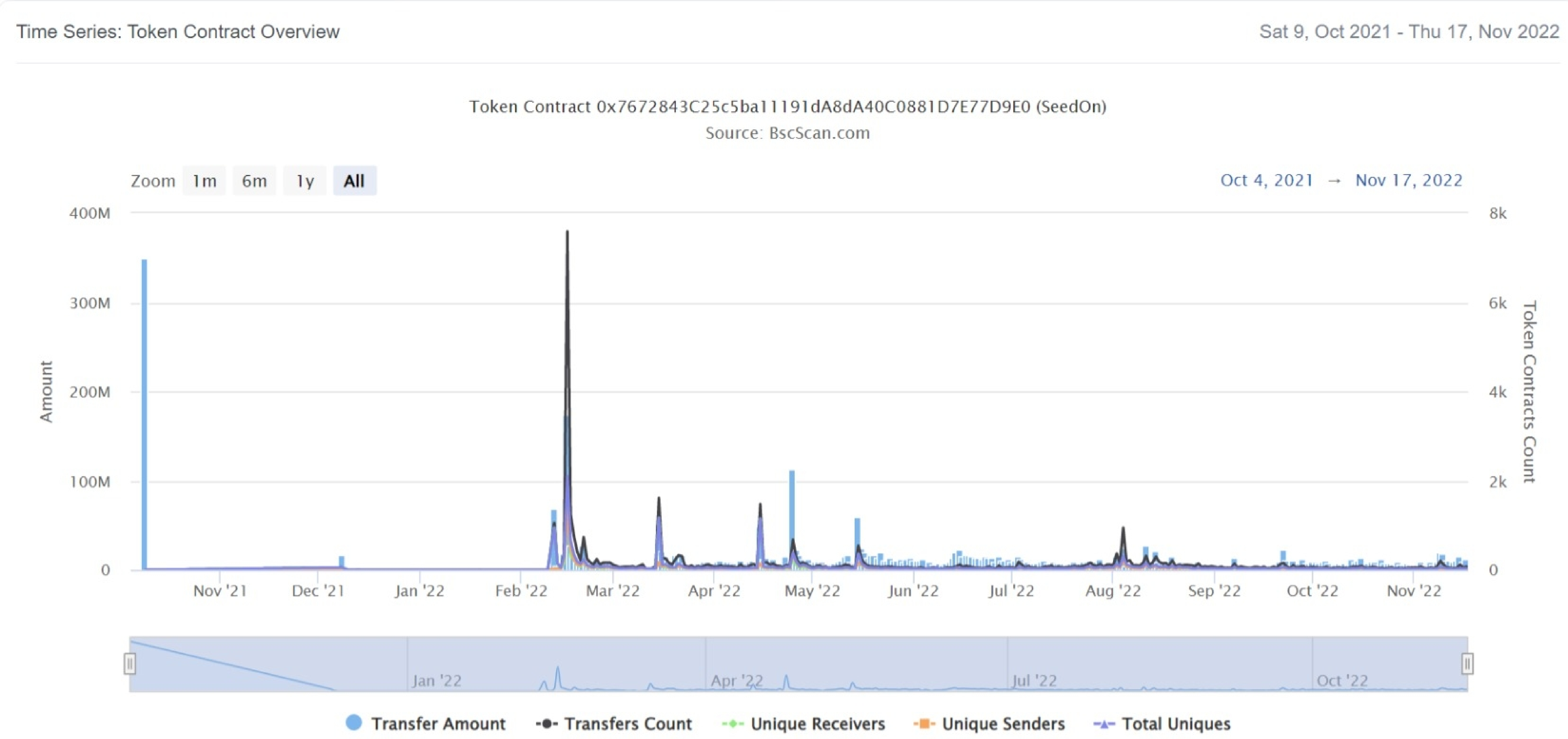
## SeedOn Top 20 Token Holders

(A total of 325,354,892.39 tokens held by the top 100 accounts from the total supply of 350,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	SeedOn: Deployer	130,768,776.676676399758565376	37.3625%
2	SeedOn: Hot Wallet	123,109,890.2461525062128153	35.1743%
3	0xffefe959d8baea028b1697abfc4285028d6ceb10	12,773,689.026233	3.6496%
4	PancakeSwap V2: SEON 17	9,460,896.530825132042216625	2.7031%
5	0x49385038c8f222b5989c117a51d22c2374f968ea	9,000,000	2.5714%
6	0xcfd78b5903fc7065e38f99b25d2cc644cc3b8802	4,966,773.83	1.4191%
7	0x6cfd213bf8a05be848039c7675c74159014269cf	3,070,801.122762040235054899	0.8774%
8	0x95b66d684135f948d930a1abb154e4f02d699e12	1,648,089.441599999827378176	0.4709%
9	0xa7b42547c941b2540654fc2f3c4675334cca955f	1,250,000	0.3571%
10	0xb617670bc253283c2ecc21f22767237d2b4b19aa	1,175,212.311253611365916713	0.3358%
11	0x9a5590a23d4a812a9cd49b03fc987b6cb32aff54	1,105,623.306899999962251264	0.3159%
12	0xe1441aa3ba8b5277510dfbc7b66e7110cd5f2fe6	876,861.04	0.2505%
13	0x5def5a1100b7f1b612962ad034d08fb855599be4	849,218.20648	0.2426%
14	0x14ccbc0458886a620dcd110175170c32a37a3805	804,742.680295710396791944	0.2299%
15	0x7ea93d7c333190817cc40a8555dfc8ced8cc2819	801,849.54476347176796928	0.2291%
16	0x29cdd2023f67877fd22cc4c7c75ba0f074efbbbe	800,000	0.2286%
17	 0x2f2418c74e3237a2739de8b5fdd34e6cfe7a736c	768,848.409605314717882019	0.2197%
18	0x95156d4612b7036c00a6f99965622e72174c85a1	754,697.684894516371199786	0.2156%
19	0x7106a7b62f8305e4ef3dea2c76c236395108c8a8	679,999.999999999966445568	0.1943%
20	0x639349df27f5e8e096b61f3733e1ad8af5a62453	675,552.606001401969449405	0.1930%

# SeedOn Distribution

## SeedOn Contract Overview





# Contract functions details

## +Context

- [Int] \_msgSender
- [Int] \_msgData

## + [Int] IERC20

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

## + [Int] IERC20Metadata (IERC20)

- [Ext] name
- [Ext] symbol
- [Ext] decimals

## +SeedOn (Context, IERC20, IERC20Metadata)

- <constructor>
- [Pub] name
- [Pub] symbol
- [Pub] decimals
- [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] \_beforeTokenTransfer
- [Int] \_afterTokenTransfer

(\$) = 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	Passed



# 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

One 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 ^0.8.7 the contract should contain the following line:

```
pragma solidity 0.8.7;
```



# 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.