



Smart Contract Security Audit Report

Crox Token

September 2022

Security Status



www.hacksafe.io



Audit Details



Audited project

Crox Token



Deployer address

0x280020e549f13E219B85A9A71D123a93D2e6172b



Client contacts

Crox Token



Blockchain

Binance smart chain



Website

<https://croxswap.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 Crox Token to perform an audit of smart contract:

- <https://bscscan.com/address/0x2c094f5a7d1146bb93850f629501eb749f6ed491#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 28.09.2022

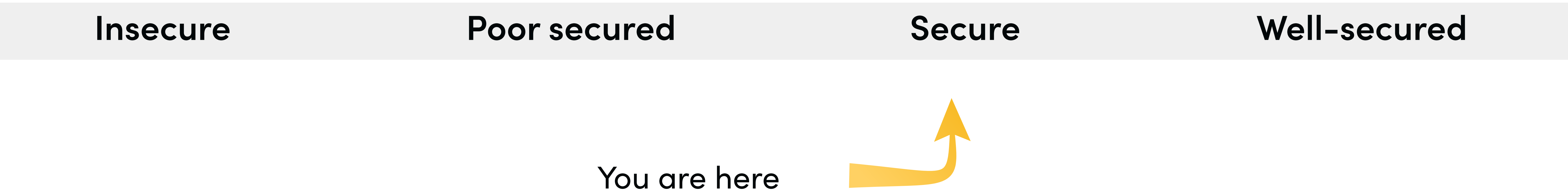
Token Type	: ERC20
Contract name	: CroxToken
Contract address	: 0x2c094F5A7D1146BB93850f629501eB749f6Ed491
Total supply	: 5,000,000
Token ticker	: CROX
Decimals	: 18
Token holders	: 8,615
Transactions count	: 206,015
Compiler version	: v0.6.12+commit.27d51765
Contract deployer address	: 0x280020e549f13E219B85A9A71D123a93D2e6172b
Owner address	: 0xeaf239a85b2c24229042a0840771a5620a36d2b3

Social profiles

Twitter Profile	: https://twitter.com/croxswap
Telegram Profile	: https://t.me/croxswap
Coingecko profile	: https://www.coingecko.com/en/coins/croxswap/
Coinmarketcap profile	: https://coinmarketcap.com/currencies/croxswap/

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.

Crox Token Token Distribution

 The top 100 holders collectively own 91.29% (4,564,325.92 Tokens) of Crox Token

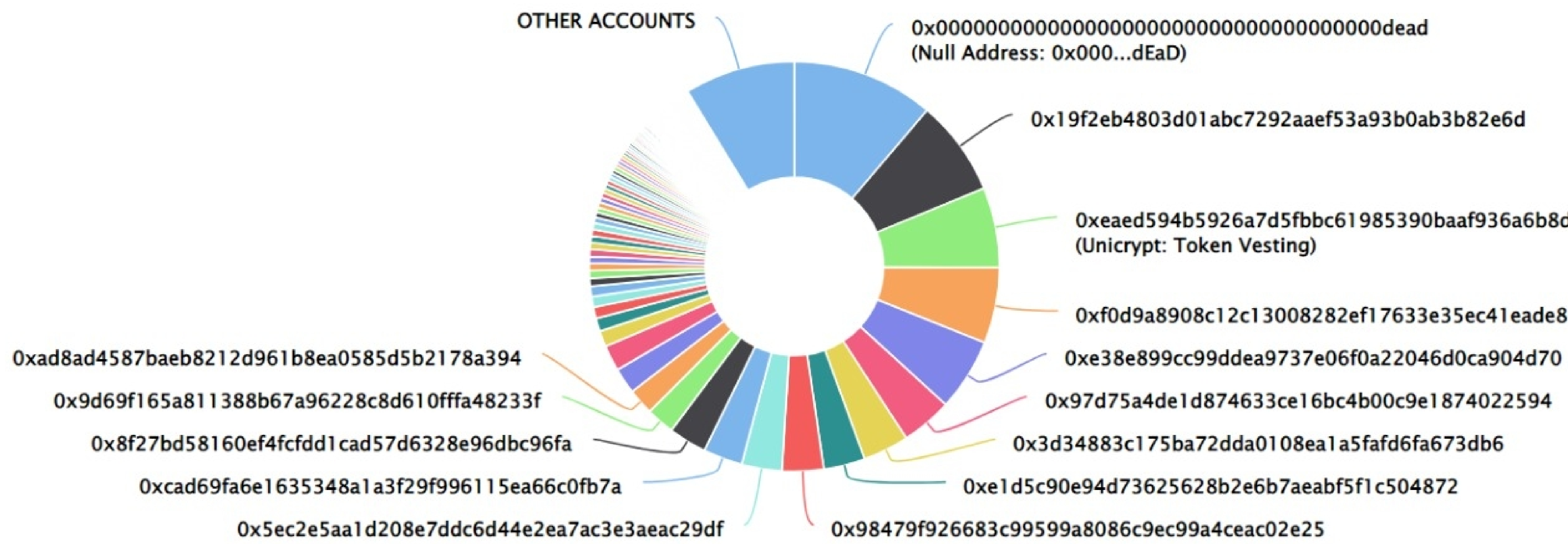
 Token Total Supply: 5,000,000.00 Token

|

Total Token Holders: 8,615








Crox Token Top 100 Token Holders

Source: BscScan.com



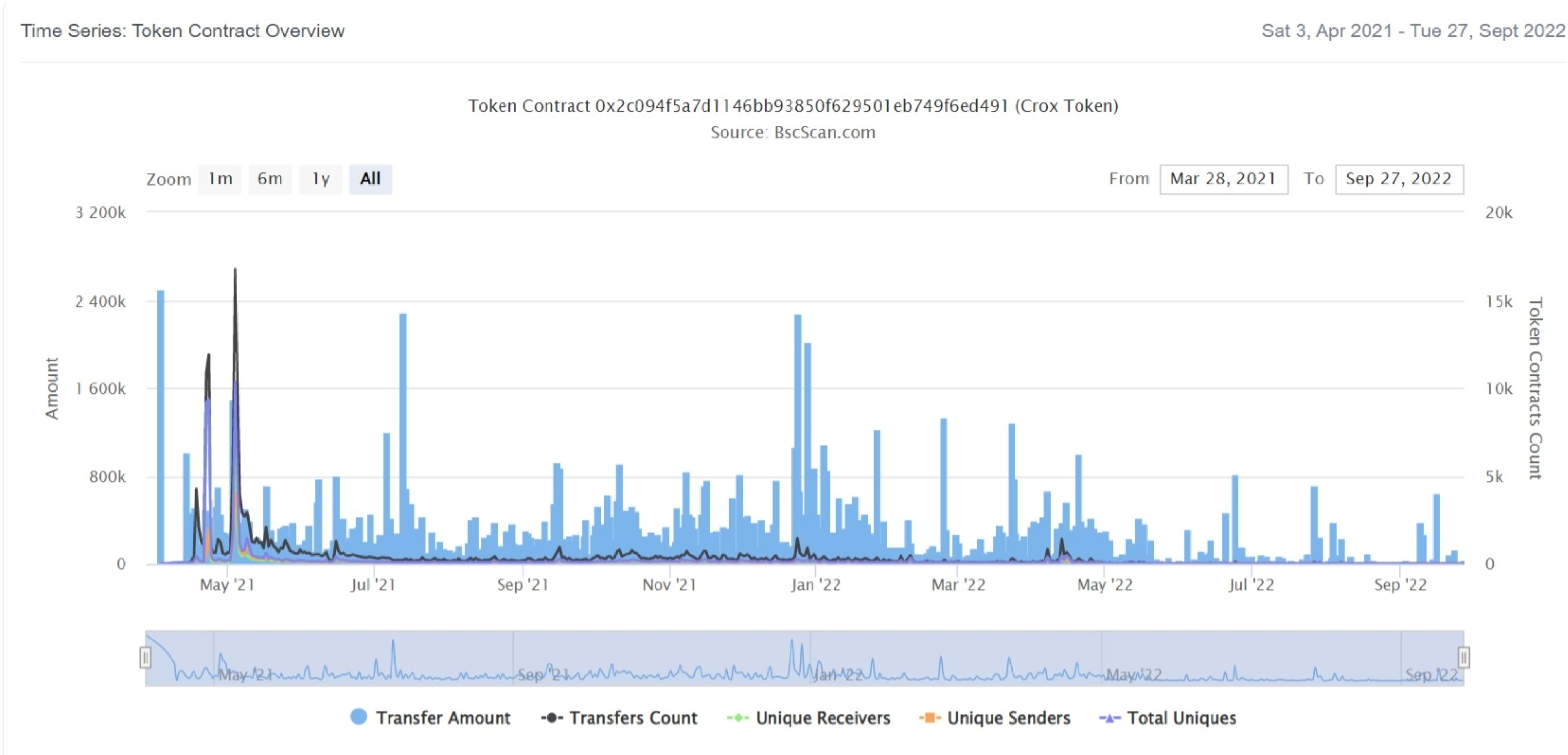
Crox Token Top 20 Token Holders

(A total of 4,564,325.92 tokens held by the top 100 accounts from the total supply of 5,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Null Address: 0x000...dEaD	561,200	11.2240%
2	0x19f2eb4803d01abc7292aaef53a93b0ab3b82e6d	377,581.402103939630815761	7.5516%
3	 Unicrypt: Token Vesting	315,597.317608681899517088	6.3119%
4	0xf0d9a8908c12c13008282ef17633e35ec41eade8	300,000.735250652064282403	6.0000%
5	 0xe38e899cc99ddea9737e06f0a22046d0ca904d70	285,502.790821184875570809	5.7101%
6	0x97d75a4de1d874633ce16bc4b00c9e1874022594	201,039.187184627543345401	4.0208%
7	 0x3d34883c175ba72dda0108ea1a5fafd6fa673db6	181,813.142222441765949437	3.6363%
8	 0xe1d5c90e94d73625628b2e6b7aeabf5f1c504872	163,031.460906277699560643	3.2606%
9	 0x98479f926683c99599a8086c9ec99a4ceac02e25	162,960.467367979875	3.2592%
10	0x5ec2e5aa1d208e7ddc6d44e2ea7ac3e3aeac29df	159,048.801227655960404695	3.1810%
11	0xcad69fa6e1635348a1a3f29f996115ea66c0fb7a	154,831.849141679883702997	3.0966%
12	0x8f27bd58160ef4fcfd1cad57d6328e96dbc96fa	150,087.99153832661450762	3.0018%
13	0x9d69f165a811388b67a96228c8d610ffa48233f	110,982.185512406701966775	2.2196%
14	0xad8ad4587baeb8212d961b8ea0585d5b2178a394	103,492.105090745818022158	2.0698%
15	0xc7233971c7a8d8a18e16fb972d515c853edefbf6	102,137.834915459195551582	2.0428%
16	0xd13447e654f6a76049974634aa88988d2d37ec32	101,866.120013290753917344	2.0373%
17	 0x4de4e7759e6fcfecb1b5be5949efd5deb3097721	62,885.571437401752274808	1.2577%
18	0xc103b511ee43339a72c0f70da9b868d7ff129679	51,910.001947977092333669	1.0382%
19	0xa9228b632c9b3c1b4fafb1bb8c3681ffae65262e	44,513.084867929343032139	0.8903%
20	 0xf41acba80328c076a207badcbbd7bc331c0853f5	41,529.800701934506810005	0.8306%

Crox Token Token Distribution

Crox Token Contract overview



Contract functions details

+ Context

- [Int] _msgSender
- [Int] _msgData

+ Ownable (Context)

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

+ [Int] IBEP20

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

+ [Lib] SafeMath

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

+ BEP20 (Context, IBEP20, Ownable)

- [Pub] < constructor>
- [Ext] getOwner
- [Pub] name

Contract functions details

- [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] _burnFrom #

+ CroxToken (BEP20)

- [Pub] <constructor> #
- [Pub] mint #
 - modifiers: onlyOwner
- [Ext] delegates
- [Ext] delegate
- [Ext] delegateBySig
- [Ext] getCurrentVotes
- [Ext] getPriorVotes
- [Int] _delegate
- [Int] _moveDelegates
- [Int] _writeCheckpoint
- [Int] safe32
- [Int] getChainId

(\$) = 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 issues 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 :

- Crox Token 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 as smart contract ownership has not 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.