HackSafe

Smart Contract Security Audit Report

Tidal Token

December 2022

Security Status

★ ★ ★ ★ ★

www.hacksafe.io

Audit Details

Audited project
Tidal Token

Deployer address
0xeb715BAc0aC8207c4B430e1C942018Beb3e8326E

Client contacts
Tidal Token Team

Blockchain
Ethereum

Website
https://tidal.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 Tidal Token to perform an audit of smart contracts:

- https://etherscan.io/token/0x29cbd0510eec0327992cd6006e63f9fa8e7f33b7#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 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 13.12.2022

Token Type	: DEFI
Contract name	: TidalToken
Contract address	: 0x29CbD0510EEc0327992CD6006e63F9Fa8E7f33B7
Total supply	: 19,838,000,000.999999999838
Token ticker	: TIDAL
Decimals	: 18
Token Holders	: 6,465
Transactions count	: 65,562
Compiler version	: v0.6.12+commit.27d51765
Contract deployer address	: 0xeb715BAc0aC8207c4B430e1C942018Beb3e8326E
Owner address	: No owner

Audit Summary

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

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.

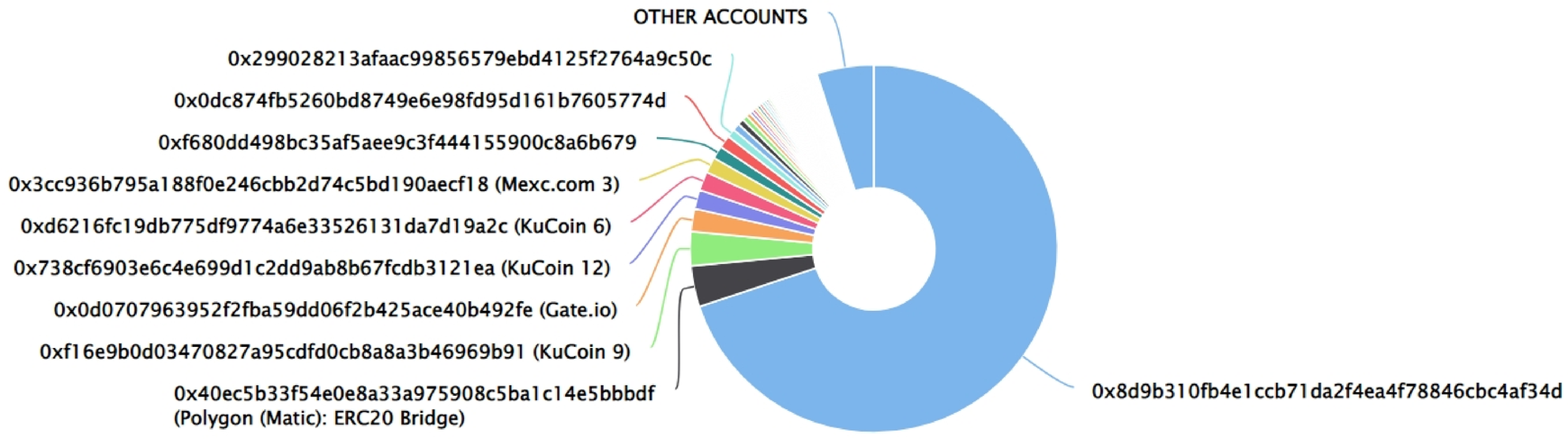
TidalToken Distribution

The top 100 holders collectively own 94.99% (18,843,439,053.30 Tokens) of Tidal Token

Token Total Supply: 19,838,000,001.00 Token | Total Token Holders: 6,465

Tidal Token Top 100 Token Holders

Source: Etherscan.io



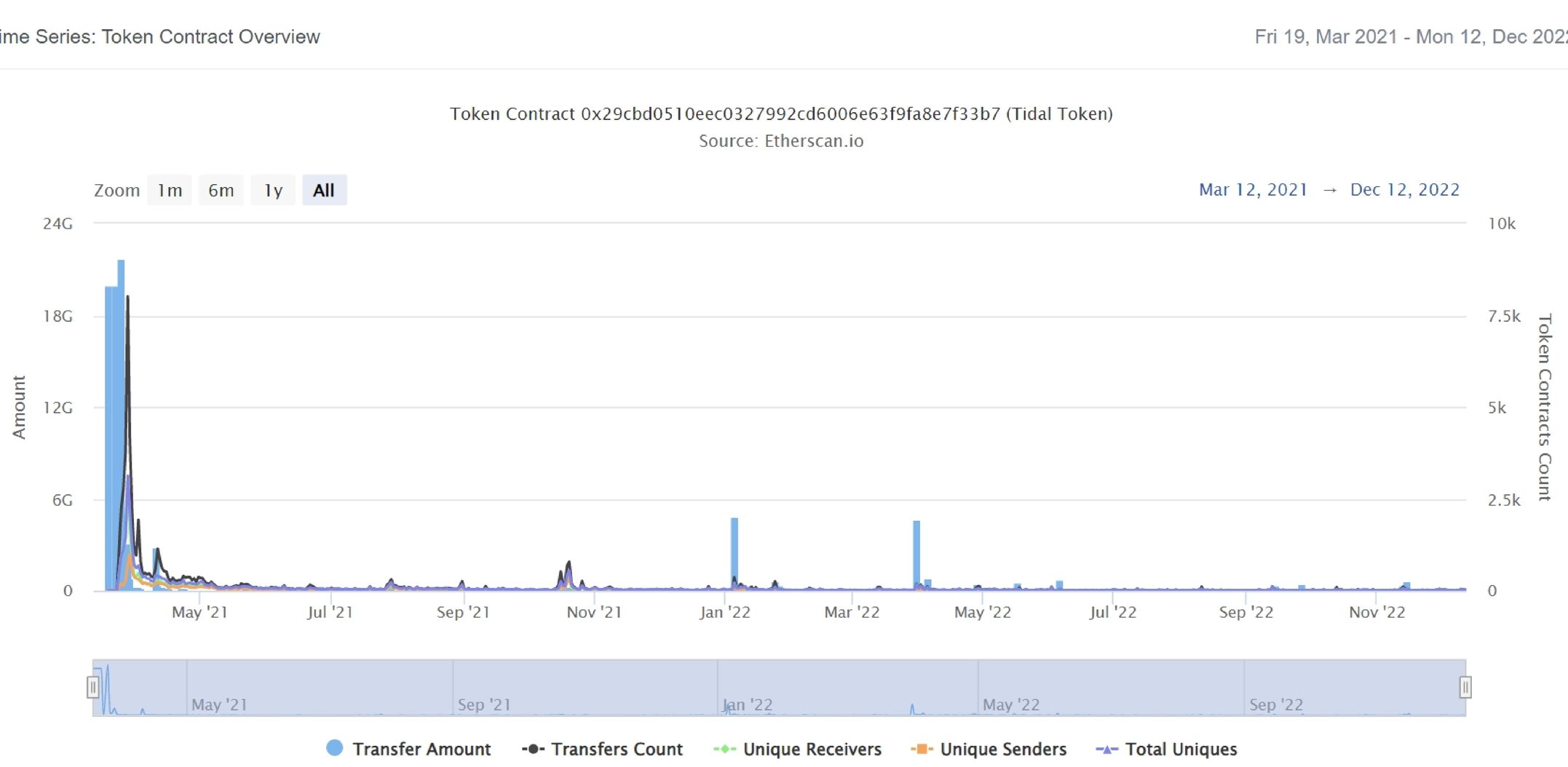
TidalToken Top 20 Token Holders

(A total of 18,843,439,053.30 tokens held by the top 100 accounts from the total supply of 19,838,000,001.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x8d9b310fb4e1ccb71da2f4ea4f78846cbc4af34d	13,868,779,591	69.9102%
2	Polygon (Matic): ERC20 Bridge	712,746,908.223767307803257756	3.5928%
3	KuCoin 9	593,958,978.762550882278703895	2.9940%
4	Gate.io	399,697,791.62336518024121008	2.0148%
5	KuCoin 12	331,179,654.586419178657366466	1.6694%
6	KuCoin 6	326,790,940	1.6473%
7	Mexc.com 3	276,627,461.324810340920887756	1.3944%
8	0xf680dd498bc35af5aee9c3f444155900c8a6b679	222,222,221.56	1.1202%
9	0x0dc874fb5260bd8749e6e98fd95d161b7605774d	218,650,022.22	1.1022%
10	0x299028213afaac99856579ebd4125f2764a9c50c	153,142,856	0.7720%
11	Gate.io 3	123,869,518.416701456839835269	0.6244%
12	Uniswap V2: TIDAL-USDC	104,942,988.803645160822676539	0.5290%
13	0x47439ee571a136fc3d38a394ebfedc966f8d491d	91,153,380	0.4595%
14	0x7582d0b7a150f6227b447a0f282bff612f0f4b3e	78,111,768.973	0.3937%
15	0x2853d6fdc1a4a445c447bd7dd17a8c656b7037b8	56,280,000	0.2837%
16	0x378faa0f46cadbb3afc4932614a268dffd28237f	52,111,111.11	0.2627%
17	0x541efe7a6c3ecbf678e2e74284eca8277dcae9ea	52,111,111.11	0.2627%
18	0xc91f2181cf37ae0a4b8da099cb35ae528364bd3e	50,580,285.781052790990390733	0.2550%
19	0x6413b9c855ffe916e7a49a8a9d457097952a73bb	49,629,629.4068	0.2502%
20	0xae0f7381156ff9cb9b273acbffcb16758b1ccc67	49,629,629.4066	0.2502%

TidalToken Distribution

TidalToken Contract Overview



Contract functions details

+[Int] IERC20
    -[Ext] totalSupply
    -[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

+[Lib] Address
    -[Int] isContract
    -[Int] sendValue
    -[Int] functionCall
    -[Int] functionCall
    -[Int] functionCallWithValue
    -[Int] functionCallWithValue
    -[Pvt] _functionCallWithValue

+ERC20 (Context, IERC20)
    -[Pub] <constructor>
    -[Pub] name
    -[Pub] symbol
    -[Pub] decimals
    -[Pub] totalSupply
    -[Pub] balanceOf
    -[Pub] transfer #
    -[Pub] allowance
    -[Pub] approve #
    -[Pub] transferFrom #
    -[Pub] increaseAllowance #
    -[Pub] decreaseAllowance #

Contract functions details

-[Int] _transfer #
-[Int] _mint #
-[Int] _burn #
-[Int] _approve #
-[Int] _setupDecimals #
-[Int] _beforeTokenTransfer

+TidalToken (ERC20)
-[Pub] <constructor>
-[Ext] burn #

($) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Passed
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**

  One low severity issue found.

1. 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

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.