HackSafe

Smart Contract Security Audit Report

Zipper

October 2022

Security Status

★ ★ ★ ★ ★

www.hacksafe.io

Audit Details

Audited project
Zipper

Deployer address
0xdA12391a57b16510aC82384640a44ECbd43243db

Client contacts
Zipper Team

Blockchain
Ethereum

Website
Not provided

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 Zipper to perform an audit of smart contracts:

- https://etherscan.io/token/0xa9d2927d3a04309e008b6af6e2e282ae2952e7fd#code

The purpose of the audit was to achieve the following:

- Ensutre 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 11.10.2022

Token Type	: ERC20
Contract name	: HumanStandardToken
Contract address	: 0xA9d2927d3a04309E008B6af6E2e282AE2952e7fD
Total supply	: 100,000,000,000
Token ticker	: ZIP
Decimals	: 18
Token holders	: 11,609
Transactions count	: 39,731
Compiler version	: v0.3.5+commit.5f97274
Contract deployer address	: 0xdA12391a57b16510aC82384640a44ECbd43243db
Owner address	: No owner

Social profiles

Coinmarketcap profile	: https://coinmarketcap.com/currencies/zip/
Coingecko profile	: https://www.coingecko.com/en/coins/zip/

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 as owner does not have control over the smart contract.

Insecure	Poor secured	Secure	Well-secured

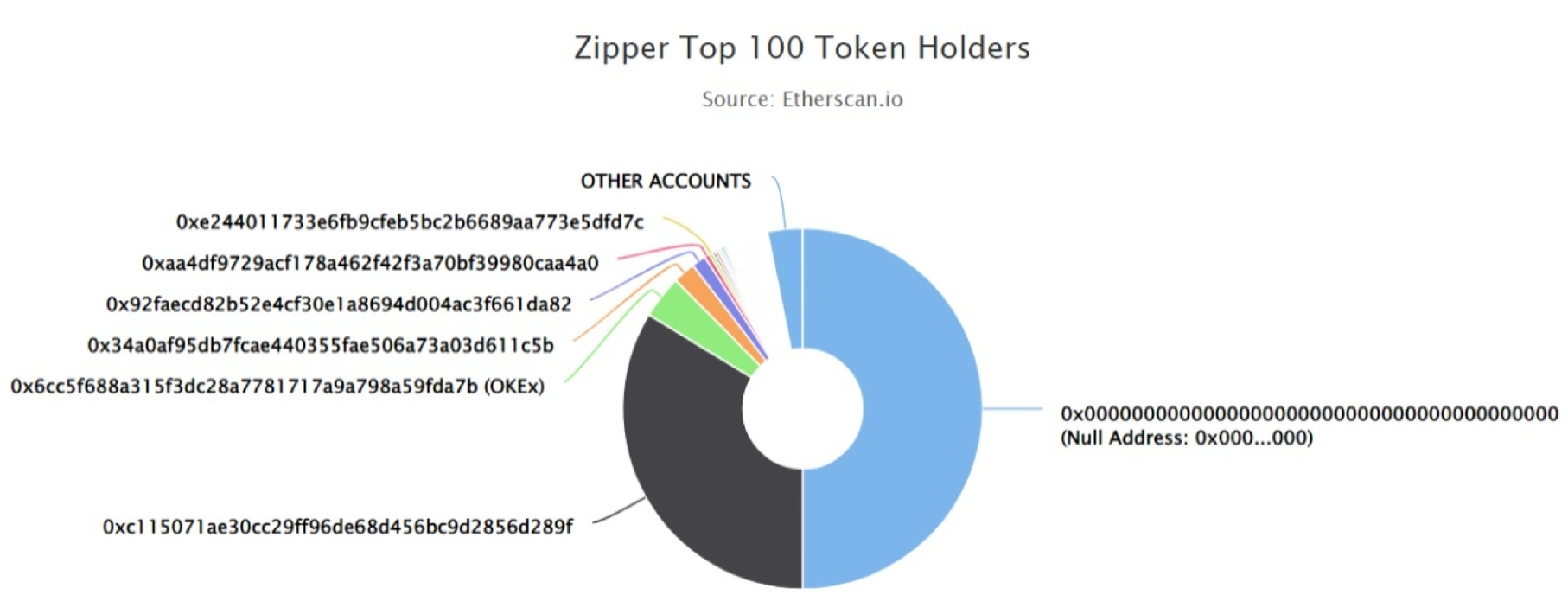
You are here

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 and some very low-level issues.

ZipperToken Distribution

The top 100 holders collectively own 96.88% (96,877,059,059.12 Tokens) of Zipper | Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 11,609



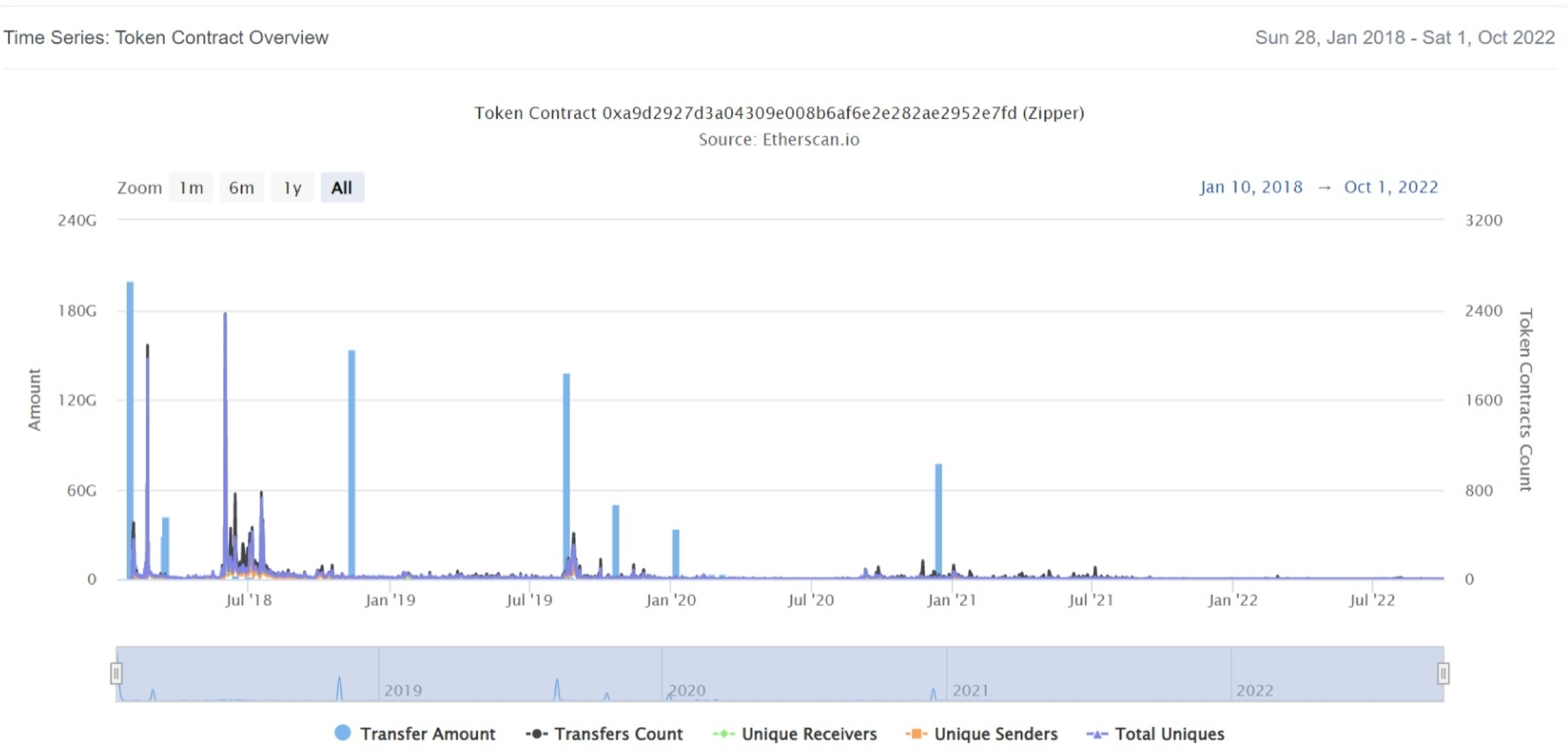
Zipper Top 20 Token Holders

(A total of 96,877,059,059.12 tokens held by the top 100 accounts from the total supply of 100,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Null Address: 0x000...000	50,000,021,111	50.0000%
2	0xc115071ae30cc29ff96de68d456bc9d2856d289f	33,700,000,188	33.7000%
3	OKEx	3,882,130,970.070251767697638696	3.8821%
4	0x34a0af95db7fcae440355fae506a73a03d611c5b	2,000,000,000	2.0000%
5	0x92faecd82b52e4cf30e1a8694d004ac3f661da82	1,276,230,294.402386858423317877	1.2762%
6	0xaa4df9729acf178a462f42f3a70bf39980caa4a0	447,917,057.114765391530545716	0.4479%
7	0xe244011733e6fb9cfeb5bc2b6689aa773e5dfd7c	320,558,324.300276677162520089	0.3206%
8	0x965ef6febe8eb31798d841ca4ad45da8bd1e17f2	257,625,489.682	0.2576%
9	0xd3a946e41736f20f4093f702b8ddf7be13224fa7	240,995,691.262	0.2410%
10	0x66264beefb1efd2a0bf6cf98db9c4ba000e5350f	223,794,428.697	0.2238%
11	0x412c23c6793a5b346ec9fd6679e853f756b90122	220,243,251.45	0.2202%
12	0x746336bbb8edbf91feda809ecf9770d8ac123a52	209,173,253.054	0.2092%
13	0x37d4e136e64c35aa25e16a8115f6c3a752797288	208,126,400.33663002	0.2081%
14	0x6704a086752b119959e8d07e1af6792594db4ea3	183,626,712.215	0.1836%
15	0x9da271822b2a7eeb5651585957fd71d442bb0e48	133,847,461.403856751863677655	0.1338%
16	0xa1328adfd383bc8c61dac5707d00cbc95f6b760d	127,853,166.052	0.1279%
17	0x6e9898efdf1d6098dd1087cf1508685e286f54b6	112,880,156.11	0.1129%
18	0x003662ae57116a248be0284fcc57661561b25e32	112,540,100	0.1125%
19	0x8ca82fcd56a2c39dafe3ead8cab4a988480371a1	100,050,924.561	0.1001%
20	0xfe699e845d95f68fb06b33af41776d18f01faad4	100,000,366.567	0.1000%

Zipper Token Distribution

Zipper Contract Overview



Contract functions details

+Token
    -totalSupply
    -balanceOf
    -transfer
    -transferFrom
    -approve
    -allowance

+StandardToken (Token)
    -transfer #
    -transferFrom #
    -balanceOf
    -approve #
    -allowance

+HumanStandardToken (StandardToken)
    -HumanStandardToken #
    -approveAndCall #

($) = 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 issues found.

✓ **Medium Severity Issues**

No medium severity issue found.

✓ **Low Severity Issues**

One low severity issue found..

1. 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.

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.