HackSafe

Smart Contract Security Audit Report

Zebi Coin

October 2022

Security Status

★★★☆☆

www.hacksafe.io

Audit Details

Audited project
Zebi Coin

Deployer address
0xd75496c0a6a36308cfda58E317aB97eA46b8F6B5

Client contacts
Zebi Coin Team

Blockchain
Ethereum

Website
https://www.zebi.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 Zebi Coin to perform an audit of smart contracts:

- https://etherscan.io/token/0x2008e3057bd734e10ad13c9eae45ff132abc1722#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 18.10.2022

Token Type	: ERC20
Contract name	: ZebiCoin
Contract address	: 0x2008e3057BD734e10AD13c9EAe45Ff132aBc1722
Total supply	: 0
Token ticker	: ZCO
Decimals	: 8
Token holders	: 7,871
Transactions count	: 65,554
Compiler version	: v0.4.19+commit.c4cbbb05
Contract deployer address	: 0xd75496c0a6a36308cfda58E317aB97eA46b8F6B5
Owner address	: 0xb7D67fE0AA7e4b932f5a98aE953702FF3a4c319C

Social profiles

Twitter profile	: https://twitter.com/ZebidataIndia
Facebook profile	: https://www.facebook.com/ZebidataIndia/
Coinmarketcap Profile	: https://coinmarketcap.com/currencies/zebi/
Coingecko profile	: https://www.coingecko.com/en/coins/zebi/
Telegram profile	: https://t.me/ZebiData

Claimed Smart Contract Features

Claimed Feature Detail

Tokenomics :

- **Name** : Zebi Coin
- **Symbol** : ZCO
- **Decimals** : 8
- **Protocol** : ERC20
- **Total supply** : 0
- **Contract address** : 0x2008e3057BD734e10AD13c9EAe45Ff132aBc1722

Our Observation

No, this is not valid as total supply is 0 while token holder's count is 7,871.

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.

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 1 critical, 0 high, 0 medium and 2 low and some very low-level issues.

Zebi Coin Token Distribution

Token Total Supply: 0.00 Token | Total Token Holders: 7,871

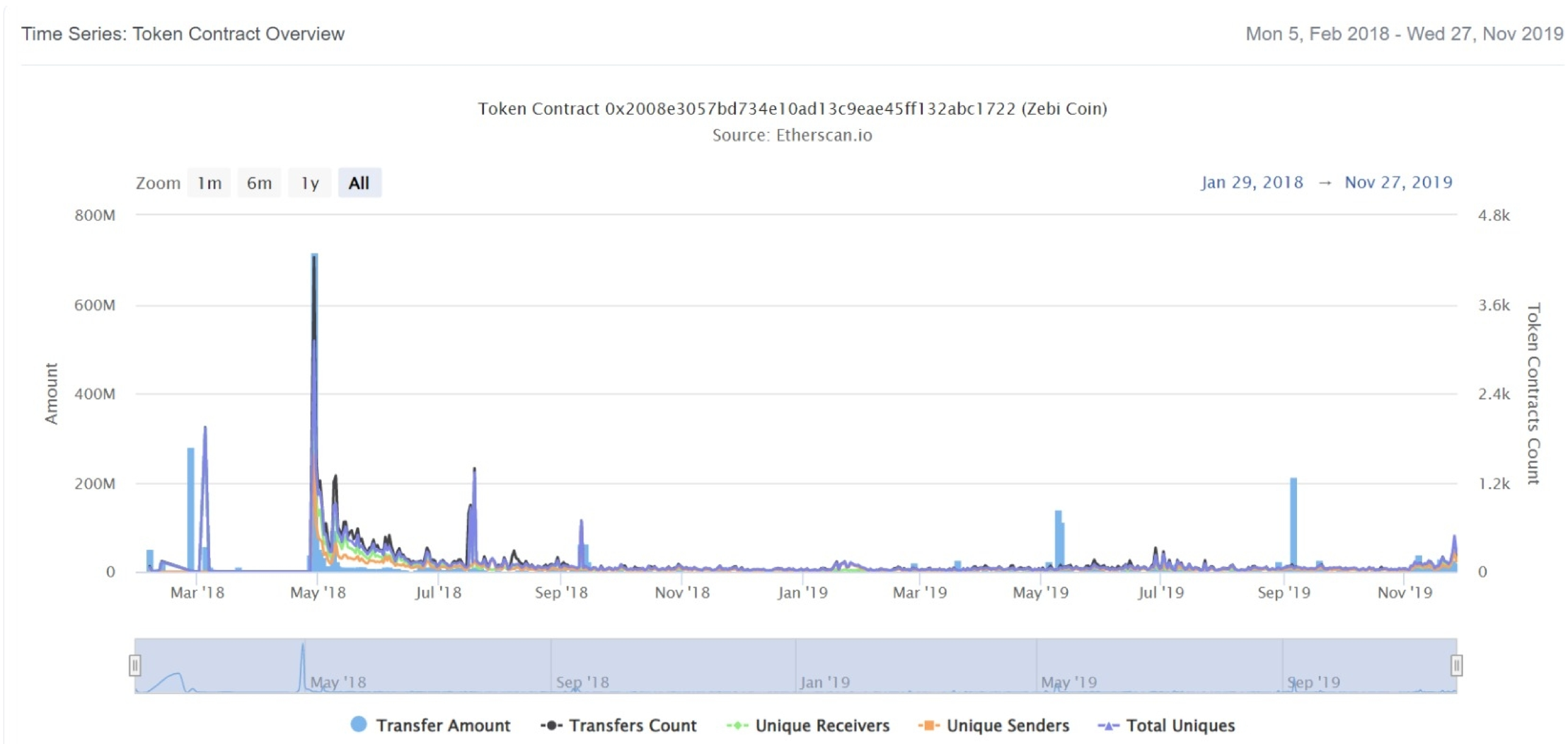
Zebi Coin Top 20 Token Holders

(A total of 27,388,404.66 tokens held by the top 100 accounts from the total supply of 0.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x55a6c92e9fd1c8ee798b29bc07e16e120e6cf926	440,432.511	-
2	0x923f9fd09443c5daa2e3786567ee072b5f825550	429,000	-
3	0x6dc9f9ad7359ec1d0471f81a45807e281a33b442	428,785.5	-
4	0x1af3af9bdce06d82305e3ff7c97e2c1ac8b8203e	428,582.07	-
5	0x0cfa4b6fcba5224a12e9131570ac54dbd75c5985	415,261.57677023	-
6	0xe5ad0f3155a1a587d2da7418767ef357253448db	406,304.65842366	-
7	0x457ebf78c4ba6a0dc770c0b0a54138feee76ec73	400,000	-
8	0xd24c079953befa1662ec5d5e8a5f2c74f7179ee0	400,000	-
9	0xb5bb0b366c2a51f5bba4ca5b103e4dbd6716339c	391,339.1013	-
10	0x08a6ab4534c5bfef29d98291756e74950e850e88	386,486	-
11	0x0297795a442ccc457908d55651d58090fa5cb763	386,479	-
12	0xc7d22c0e145cc62cd61a6ba0d35bfc1d34fcbc10	382,620.9	-
13	0xb1ac1ad3e555f891be56aa5bda9680b8bedbd4b4	380,688	-
14	0xe6a1023f438aaffd3bbb325c9ea0dea37dc92f84	369,369.46419773	-
15	0x31499cf30a4ab17c4ea6b0a35576e7feec3bb433	369,025.75124378	-
16	0x11c2e8cc3aeb67d08d6505bf72649c3d1efc2ef2	363,447.6584	-
17	0x21819f2aa2d869c69673af7ecc5239517ad07e74	362,803.5931	-
18	0xe60223de88fa820581ff4d2973290e168f43ebc7	347,232.48181531	-
19	0x1d33c40588687188d2216af9731c5ea3936c97f3	345,945.60475262	-
20	0xeba35d6ecd285c37af899ad2bf8a9b3d4de191b5	335,000.0004114	-

Zebi Coin Token Distribution

Zebi Coin Contract Overview



Contract functions details

+ Ownable
  - -[Pub] Ownable
  - -[Pub] transferOwnership #
    - -modifiers: onlyOwner

+ERC20Basic
  - -[Pub] totalSupply
  - -[Pub] balanceOf
  - -[Pub] transfer

+ERC20 (ERC20Basic)
  - -[Pub] allowance
  - -[Pub] transferFrom
  - -[Pub] approve

+[Lib] SafeMath
  - -[Int] mul
  - -[Int] div
  - -[Int] sub
  - -[Int] add

+BasicToken (ERC20Basic)
  - -[Pub] totalSupply
  - -[Pub] transfer #
  - -[Pub] balanceOf

+StandardToken (ERC20, BasicToken)
  - -[Pub] transferFrom #
  - -[Pub] approve #
  - -[Pub] allowance
  - -[Pub] increaseApproval #
  - -[Pub] decreaseApproval #

+MintableToken (StandardToken, Ownable)
  - -[Pub] mint #
    - -modifiers: onlyOwner, canMint
  - -[Pub] finishMinting #
    - -modifiers: onlyOwner, canMint
  - -[Pub] resumeMinting #

Contract functions details

-modifiers: onlyOwner
-[Ext] burn #
-modifiers: onlyOwner
-[Ext] startTransfer #
-modifiers: onlyOwner
-[Ext] endTransfer #
-modifiers: onlyOwner
-[Pub] transfer #
-[Pub] transferFrom #

+ZebiCoin (MintableToken)

($) = 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.	Critical issue
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

One critical severity issue found.

1. Scoping and Declarations.

- **Description**

  Owner can burn tokens of any address.

- **Recommendation**

  We advise you to check require if address has given allowance to owner to burn their tokens or not.

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. 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.

2. 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.

Security Issues

- **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 0.4.18 the contract should contain the following line:

  pragma solidity 0.4.19;

Centralization

Owner Privileges :

- Zebi Coin Contract:
    - Owner can and transfer ownership.
    - Owner can mint tokens.
    - Owner can finish minting.
    - Owner can resume minting.
    - Owner can burn tokens.
    - Owner can start and end transfer.
    - Owner can burn 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:

- Transferownership
- Finishminting
- Resumeminting
- Burn
- Starttransfer
- Endtransfer
- Mint

Conclusion

Smart contract contains low and critical 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.