soken

SMART CONTRACT
SECURITY AUDIT

Zeus Farms

July, 2021

Table of Contents

Table of Contents 2
Disclaimer 3
Procedure 4
Terminology 5
Limitations 5
Token Contract Details for 07.06.2021 6
Audit Details 6
ZeusFarms Token Distribution 7
Contract Function Details 8
Vulnerabilities checking Status 11
Security Issues 12
Conclusion 13

Disclaimer

This is a comprehensive report based on our automated and manual examination of cybersecurity vulnerabilities and framework flaws. We took into consideration smart contract based algorithms, as well. Reading the full analysis report is essential to build your understanding of project's security level. It is crucial to take note, though we have done our best to perform this analysis and report, that you should not rely on the our research and cannot claim what it states or how we created it. Before making any judgments, you have to conduct your own independent research. We will discuss this in more depth in the following disclaimer - please read it fully.

DISCLAIMER: You agree to the terms of this disclaimer by reading this report or any portion thereof. Please stop reading this report and remove and delete any copies of this report that you download and/or print if you do not agree to these conditions. This report is for non-reliability information only and does not represent investment advice. No one shall be entitled to depend on the report or its contents, and Soken and its affiliates shall not be held responsible to you or anyone else, nor shall Soken provide any guarantee or representation to any person with regard to the accuracy or integrity of the report. Without any terms, warranties or other conditions other than as set forth in that exclusion and Soken excludes hereby all representations, warrants, conditions and other terms (including, without limitation, guarantees implied by the law of satisfactory quality, fitness for purposes and the use of reasonable care and skills). The report is provided as "as is" and does not contain any terms and conditions. Except as legally banned, Soken disclaims all responsibility and responsibilities and no claim against Soken is made to any amount or type of loss or damages (without limitation, direct, indirect, special, punitive, consequential or pure economic loses or losses) that may be caused by you or any other person, or any damages or damages, including without limitations (whether innocent or negligent).

Security analysis is based only on the smart contracts. No applications or operations were reviewed for security. No product code has been reviewed.

Procedure

Our analysis contains following steps:

1. Project Analysis;

2. Manual analysis of smart contracts:
- Deploying smart contracts on any of the network(Ropsten/Rinkeby) using Remix IDE
- Hashes of all transaction will be recorded
- Behaviour of functions and gas consumption is noted, as well.

3. Unit Testing:
- Smart contract functions will be unit tested on multiple parameters and under multiple conditions to ensure that all paths of functions are functioning as intended.
- In this phase intended behaviour of smart contract is verified.
- In this phase, we would also ensure that smart contract functions are not consuming unnecessary gas.
- Gas limits of functions will be verified in this stage.

4. Automated Testing:
- Mythril
- Oyente
- Manticore
- Solgraph

Terminology

We categorize the finding into 4 categories based on their vulnerability:

- Low-severity issue — less important, must be analyzed
- Medium-severity issue — important, needs to be analyzed and fixed
- High-severity issue —important, might cause vulnerabilities, must be analyzed and fixed
- Critical-severity issue —serious bug causes, must be analyzed and fixed.

Limitations

The security audit of Smart Contract cannot cover all vulnerabilities. Even if no vulnerabilities are detected in the audit, there is no guarantee that future smart contracts are safe. Smart contracts are in most cases safeguarded against specific sorts of attacks. In order to find as many flaws as possible, we carried out a comprehensive smart contract audit. Audit is a document that is not legally binding and guarantees nothing.

Token Contract Details for 07.06.2021

Contract Name: **ZeusFarms Token**

Deployer address: **0xa3b43Ea19758bc79929dA12156866Ad26a5E136d**

Total Supply: **489,392.579595**

Token Tracker: **ZEUS**

Decimals: **18**

Token holders: **157**

Transactions count: **234**

Top 100 holders dominance: **99.83%**

Contract deployer address:
0xa3b43Ea19758bc79929dA12156866Ad26a5E136d

Audit Details

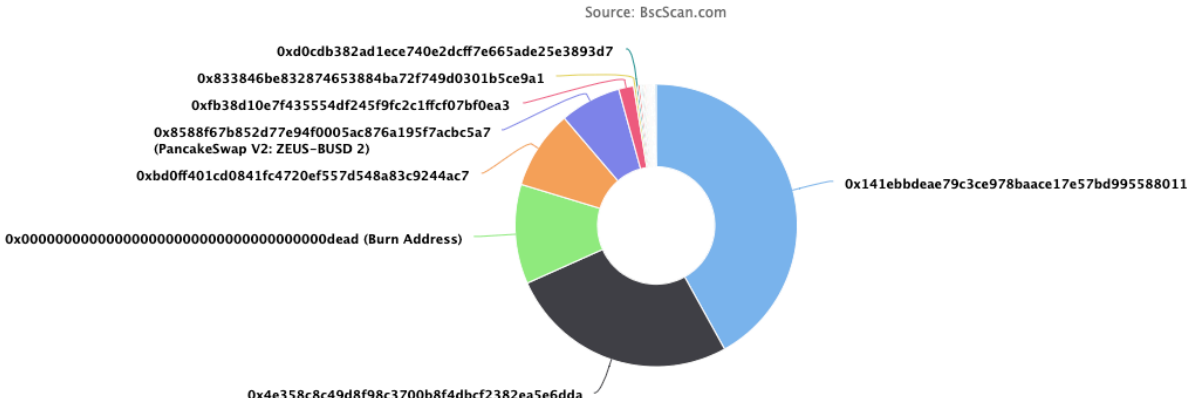
Project Name: **Zeus Farms**

Language: **Solidity**

Blockchain: **Binance Smart Chain**

Project Website: **zeusfarmsdefi.com**

ZeusFarms Token Distribution



ZeusFarms Top 10 Holders

Rank	Address	Quantity (Token)	Percentage
1	0x141ebbdeae79c3ce978baace17e57bd995588011	205,294.676627710762596495	41.9489%
2	PancakeSwap V2: ZEUS 46	129,082.72505347107264201	26.3761%
3	Burn Address	55,647.459797912725212818	11.3707%
4	0xbd0ff401cd0841fc4720ef557d548a83c9244ac7	44,214.499774027453417707	9.0346%
5	PancakeSwap V2: ZEUS-BUSD 2	34,242.35747908059028659	6.9969%
6	0xfb38d10e7f435554df245f9fc2c1ffcf07bf0ea3	8,389.495601701289686136	1.7143%
7	0x833846be832874653884ba72f749d0301b5ce9a1	1,405.83354023147758065	0.2873%
8	0xd0cdb382ad1ece740e2dcff7e665ade25e3893d7	1,154.201256767276215441	0.2358%
9	0x4c5016f43195f27dd9c7888272e4ec235a196a76	1,067.889840898251791512	0.2182%
10	0x73c351216580793d8b14f866ccb3d14de446c59d	696.453535724384582292	0.1423%

Contract Function Details

+ Context.sol
- [Int] _msgSender
- [Int] _msgData

+ [Int] IBEP20.sol
- [Ext] totalSupply
- [Ext] balanceOf
- [Ext] decimals
- [Ext] symbol
- [Ext] name
- [Ext] getOwner
- [Ext] balanceOf
- [Ext] transfer #
- [Ext] allowance
- [Ext] approve #
- [Ext] transferFrom #

+ [Lib] SafeMath
- [Int] tryAdd
- [Int] trySub
- [Int] tryMul
- [Int] tryMod
- [Int] tryDiv
- [Int] add
- [Int] sub
- [Int] sub
- [Int] mul
- [Int] div
- [Int] div
- [Int] mod
- [Int] mod

+ [Lib] Address.sol
- [Int] isContract
- [Int] sendValue #
- [Int] functionCall #
- [Int] functionCall #
- [Int] functionCallWithValue #
- [Int] functionCallWithValue #
- [Int] functionStaticCall #
- [Int] functionStaticCall #
- [Int] functionDelegateCall #
- [Int] functionDelegateCall #

- [Prv] verifyCallResult #

+ Ownable.sol (Context)
- [Pub] <Constructor> #
- [Pub] owner
- [Pub] renounceOwnership #
    - modifiers: onlyOwner
- [Pub] transferOwnership #
    - modifiers: onlyOwner

+ ZeusToken.sol

- [Pub] mint

- [Ext] delegates
- [Ext] delegate #
- [Ext] delegateBySig
- [Ext] getCurrentVotes
- [Ext] getPriorVotes

- [Int] _delegate #
- [Int] _moveDelegates #
- [Int] _writeCheckpoint #
- [Int] safe32
- [Int] getChainId

+ BEP20
- [Pub] <Constructor>
- [Ext] getOwner
- [Pub] name
- [Pub] symbol
- [Pub] decimals
- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] allowance
- [Pub] approve #
- [Pub] transfer #
- [Pub] transferFrom #
- [Pub] increaseAllowance
- [Pub] decreaseAllowance
- [Pub] mint

- [Int] _transfer #
- [Int] _mint #
- [Int] _burn #

- [Int] _approve #
- [Int] _burnFrom #

($) = payable function
= non-constant function

Vulnerabilities checking Status

Issue Description	Checking Status
Compiler Errors	Completed
Delays in Data Delivery	Completed
Re-entrancy	Completed
Transaction-Ordering Dependence	Completed
Compiler Errors	Completed
Compiler Errors	Completed
Delays in Data Delivery	Completed
Re-entrancy	Completed
Transaction-Ordering Dependence	Completed
Compiler Errors	Completed
Compiler Errors	Completed
Delays in Data Delivery	Completed
Re-entrancy	Completed
Transaction-Ordering Dependence	Completed
Compiler Errors	Completed
Compiler Errors	Completed
Delays in Data Delivery	Completed
Design Logic	Low-severity issues
Transaction-Ordering Dependence	Completed

Security Issues

1) Wrong burning issue

There is sending tokens to the dead address in overridden _transfer

Recommendation:

There should be a burn instead of sending to the dead address.

2) Owner privileges:

Owner can mint before transferring ownership to MasterChef.

Conclusion

Low-severity issues exist within smart contracts. Smart contracts are free from any critical or high-severity issues. Smart contracts contain owner privileges. Audited only the token of the project.

NOTE: Please check the disclaimer above and note, that audit makes no statements or warranties on business model, investment attractiveness or code sustainability.