Audit of smart contracts SodaDeFi

Revision 2 dated 03.22.2021

Contents

Brief information

Project: SodaDeFi
Network: Binance
Compiler version: ^0.5.8
Optimization: enabled
The audit date: 03.22.2021

Information

The contract code was reviewed and analyzed for vulnerabilities, logical errors and developer exit scams possibilities. This work was carried out concerning the project source code and documentation provided by the customer.

Provided documentation is strongly technical, so logical analysis was made using common sense and can mismatch developer's intentions.

General conclusion

As a result of the audit, no errors were found that affect the security of users' funds on the contract. No obvious signs of an exit scam were found. No backdoors found either. This contract depends on SODA token contract. Source code of the SODA contract was provided just for information. Audit of this token is out of scope of this document.

Telescr.in guarantees the SodaDeFi contract security and performance.

Liability disclaimer

The telescr.in team within this audit framework is not responsible for the developers or third parties’ actions on the platforms associated with this project (websites, mobile applications, and so on). The audit confirms and guarantees only the smart contract correct functioning in the revision provided by the project developers (check the revision).

Confirmed by digital signature

Aggregated data

The Contract analysis was performed using the following methods:

- Static analysis
  - Checking the code for common errors leading to the most common vulnerabilities
- Dynamic analysis
  - The Contract launching and carrying out the attacks various kinds to identify vulnerabilities
- Code Review

Received data

Recommendation	**Type**	**Priority**	**Occurrence probability**	**Line of error**
Using initial price for SODA	Warning	Medium	Medium	459, 760
Possible overflow errors	Warning	Medium	Low	669
Investments use only Plan 0	Remark	Low	N/a	72, 232, 311->365
SODA staking uses only plan 1	Remark	Low	N/a	81,348->371
Linear interest rate	Remark	Low	N/a	242
Confusion in ticket definitions	Remark	Medium	N/a	574->592

A. Errors

Not found in this revision.

B. Warnings

1. Using initial price for SODA in later calculations

When calculating sodaReward Day0 price is used, although it's increasing 0.2% day by day.

2. Possible overflow errors

SafeMath is not used in this case.

C. Notice

Not found

D. Remarks

1. Investments use only Plan 0

Seems that other levels were designed but weren't implemented. This doesn't affect contract behavior.

2. SODA staking uses only plan 1

Seems that other levels were designed but weren't implemented. This doesn't affect contract behavior.

3. Linear interest rate

For simplicity contract uses linear interest rate for SODA price of +0.2% daily from initial price. Complex interest rate would use +0.2% from previous day value and would give better profitability.

4. Confusion in ticket definitions

One ticket is designed to be 1e18 decimals value. That's OK if we want to keep fractional part, but there is a confusion in it's calculation. It should be withdrawalAmount.mul(1e18).div(SODA_PER_TICKET); instead of withdrawalAmount.mul(SODA_PER_TICKET).div(1e18);. For current settings of the project (SODA_PER_TICKET = 1e18) it makes no difference, but if this value is changed, it will cause errors.

Application. Error classification

Priority	
informational	This question is not directly related to functionality but may be important to understand.
low	This question has nothing to do with security, but it can affect some behavior in unexpected ways.
Medium	The problem affects some functionality but does not result in an economically significant user funds loss.
high	This issue can result in the user funds loss.
Probability	
low	It is unlikely that the system is in a state in which an error could occur or could be caused by any party.
Medium	This problem may likely arise or be caused by some party.
high	It is highly likely that this problem could arise or could be exploited by some parties.

Application. Digital bytecode print

The audit was carried out for the code certain version on the compiler version ^0.5.8 with the optimization enabled.

To check the contract bytecode for identity to the one that was analyzed during the audit, you must:

1. Get contract bytecode (in any block explorer)
2. Get SHA1 from bytecode string
3. Compare with reference in this report

Sha1 from bytecode:

28bfa1cebf397cbc3f07254d6b5bc0f364e435eb

Sha1 from bytecode (non-metadata):

07d0315b7450b092c04eac4cb1c93db9ba633a45

Contract address:

0x5cC65CE96604E290c834f2625fA924ff7956f264

Check the digital print

Application. Signature of the audit report

```
{
  "address": "0x505ade8cea4db608250e503a5e8d4cb436044d2e",
  "msg": "As a result of the audit, no errors were found that affect the security of users' funds on the contract. No obvious signs of an exit scam were
found. No backdoors found either. This contract depends on SODA token contract. Source code of the SODA contract was provided just for information. Audit of
this token is out of scope of this document. Telescr.in guarantees the SodaDeFi contract security and performance. Sha1 from bytecode:
28bfa1cebf397cbc3f07254d6b5bc0f364e435eb Sha1 from bytecode (non-metadata): 07d0315b7450b092c04eac4cb1c93db9ba633a45 Contract address:
0x5cC65CE96604E290c834f2625fA924ff7956f264\n",
  "sig": "0xb6efe695c63f2fd7d73b48dda851c9a2b51ad96a8e3ded1ffc6d6138f00456fb00107c4a6857c38bd72a157acde959f418f3054658862ad161f35c136f0054b61b",
  "version": "3",
}
```

Check the signature