Smart Contract Audit Cryptobank

Revision 1 dated 11.30.2020

Telescr.in

Contents

Smart Contract Audit Cryptobank 1

Contents 2

Brief information 3

Information 3

General conclusion 3

Liability disclaimer 3

Aggregated data 4

Received data 4

A. Errors 5

B. Remarks 6

1. No checks 6

2. Overflow Probability 6

C. Warnings 7

Application. Error classification 8

Application. Digital bytecode print 9

Application. Signature of the audit report 1

Brief information

Project: c-bank.io
Web: TRON
Compiler version: 0.5.4
Optimization: enabled
The audit date: 11.30.2020

Information

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

During the audit, remarks were discovered that do not directly affect the funds' safety.

The detected problems full list can be found below.

General conclusion

As the audit result, 2 remarks were revealed that did not affect the users' funds security on the contract. The exit scam clear signs - not found. The first remark is related to some additional checks' absence in the contract code. The second - with the peculiarity of calculating dividends when withdrawing, which are practically unable to affect the correctness.

Telescr.in guarantees the CryptoBank 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**
No checks	Remarks	Low	Low
Overflow Probability	Remarks	Average	Low

A. Errors

Not found

B. Remarks

1. No checks

In the payable methods fallback (default method) and invest do not check that the investor's address is not the contract address. This could be a vulnerability for a reentrancy attack. Though, in the current code, there are no possibilities for its implementation.
Recommendation: add a standard check for this case: require
require(!isContract(msg.sender) && msg.sender == tx.origin);

2. Overflow Probability

The _calculateDividends(…) closed method does not use the SafeMath library for calculations. Even though the overflow probability during calculations in this method is minuscule, the general recommendation is to use SafeMath for such calculations.

C. Warnings

Not found

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 behaviour in unexpected ways.
Average	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.
Average	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 0.5.9 with the optimization enabled.

To check the contract bytecode for identity to the one that was analysed 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:
6f0edbd7c4c7cef389e607f56b60136257809705
Sha1 from bytecode (non-metadata):
5af41e13ab0d3123c95cc6b92b5af81ba9981e33
Contract address:
TCTVAmXSpc23tNWpVaoTj9jUah4dvZyNvG

Check the digital print

Telescr.in

Application. Signature of the audit report


```
{
  "address": "0x505ade8cea4db608250e503a5e8d4cb436044d2e",
  "msg": "As the audit result, 2 remarks were revealed that did not affect the users' funds security on the contract. The exit scam clear signs - not found. The first remark is related to some additional checks' absence in the contract code. The second - with the peculiarity of calculating dividends when withdrawing, which are practically unable to affect the correctness. bytecode sha1: 6f0edbd7c4c7cef389e607f56b60136257809705, bytecode sha1 (no metadata): 5af41e13ab0d3123c95cc6b92b5af81ba9981e33, contract address: TCTVAmXSpc23tNWpVaoTj9jUah4dvZyNvG",
  "sig": "0xcc3bdb9ce8ac2eb7098c9c27e93a37152516af42021b34eb9cf6b1fbaade86921bdf95ab5cda3fa92473c187595f6a3d3bb76b424ac6c2cd3483c9a31ba54abe1c",
  "version": "3"
}
```


Check the signature