Telescr.in

Audit of smart contracts tronvercity

Revision 1 dated 01.15.2021

Telescr.in

Contents

Input data

constructor(address payable marketingAddr, address payable projectAddr) public

Contract constructor, sets Marketing and Project addresses on contact creation

function invest(address referrer) public payable

Main investing entry point. Splits deposit into 4 parts: Marketing - 8%, Project - 2%, Referral - up to 8%, Contract - all the rest. Links user to referrer, and adds user deposit to deposit pool.

Limitations: investor address is not contract, value is more than or equal 100 TRX, investor made less than 100 deposits.

function withdraw() public

Main withdraw entry point. Computes dividends for user at the moment of method call, sends tokens to user and enables anti-panic mode (see documentation).

function getContractBalance() public view returns (uint256)

Returns contract balance

function getContractRate() public view returns (uint256)

Returns current contract's interest rate

function getUserPercentRate(address userAddress) public view returns (uint256)

Returns current user's interest rate

function getUserAvailable(address userAddress) public view returns (uint256)

Returns available user's dividends

function getUserCheckpoint(address userAddress) public view returns(uint256)

Returns user's checkpoint timestamp (anti-panic mode)

function isActive(address userAddress) public view returns (bool)

Returns true if user have active deposits

function getUserDepositInfo(address userAddress, uint256 index) public view returns(uint256, uint256, uint256)

Returns deposit information: deposit amount, withdrawn amount, start timestamp

function getUserAmountOfDeposits(address userAddress) public view returns(uint256)

Returns number of user's deposits

function getUserTotalDeposits(address userAddress) public view returns(uint256)

Returns total amount of deposits

function getUserTotalWithdrawn(address userAddress) public view returns(uint256)

Returns total amount of withdrawn funds

function getUserBlockRemovalTime(address userAddress) public view returns(uint256)

Returns unblocking timestamp for anti-panic mode

function getUserLastDepositDate(address userAddress) public view returns(uint256)

Returns timestamp of most recent user's deposit

function getSiteStats() public view returns (uint256, uint256, uint256, uint256, uint256, uint256)

Returns statistics of contract: number of investors, invested amount, withdrawn amount, number of deposits, contract balance, contract interest rate

function getUserStats(address userAddress) public view returns (uint256, uint256, uint256, uint256, uint256)

Returns user statistic: interest rate, available dividends, timestamp of anti-panic mode end, withdrawn amount, timestamp of last action

function getUserDepStats(address userAddress) public view returns (uint256, uint256, uint256)

Returns user deposit statistics: number of investments, invested amount, last deposit time

function getUserRefStats(address userAddress) public view returns (address, uint32, uint32, uint32)

Returns referral statistics of user: referrer, referrals level 1, 2 and 3

Brief information

Project: tronvercity.com
Network: TRON
Compiler version: 0.5.10
Optimization: enabled
The audit date: 01.15.2021

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 for the project is very poor, so logical analysis was made using common sense and can contradict developer's logic.

During the audit no errors were found that can affect the security of funds. Exit scam possibility was not detected.

General conclusion

As a result of the audit, no errors were discovered that affect the safety of funds of smart contract users. No clear signs of an exit scam were found.

Telescr.in guarantees the safety and performance of the Tronvercity contract.

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.

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**	**Probability of occurrence**
Losing part of account balance probability for withdraw	Warning	Low	Low
No explicit return value	Notice	Low	Low

A. Errors

Not found.

B. Remarks

Not found.

C. Warnings

1. Losing part of account balance probability for withdraw

function withdraw(): Logical issue: when the amount to be withdrawn is greater than balance of the contract - the maximum possible amount is displayed, but there is no record that the amount is not fully withdrawn.
Recommendation: check the balance of the contract before withdraw.

D. Notice

1. No explicit return value

function getUserLastDepositDate(address userAddress) - There is no explicit return operator in case user didn't make any deposit.

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.
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 0.5.10 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:

05f859638a2f746ea41ec08ebd5337af29b8fdda

Sha1 from bytecode (non-metadata):

6a19c9f6f6017194cee8fac03e86746ad68673d2

Contract address:

TRFhuhCgMiTraZDtd2Pq7NGhY39YVYBQJD

Check the digital print

Telescr.in

Application. Signature of the audit report


```
{
  "address": "0x505ade8cea4db608250e503a5e8d4cb436044d2e",
  "msg": "As a result of the audit, no errors were discovered that affect the safety of funds of smart contract users. No clear signs of an exit scam were found. Telescr.in guarantees the safety and performance of the Tronvercity contract. Sha1 from bytecode: 05f859638a2f746ea41ec08ebd5337af29b8fdda Sha1 from bytecode (non-metadata): 6a19c9f6f6017194cee8fac03e86746ad68673d2 Contract address:  TRFhuhCgMiTraZDtd2Pq7NGhY39YVYBQJD",
  "sig": "0xeab94261446d160a1ac2db0c0538fc81eac26e3fcd456bb3640a647db9e40fa438183f8392b9293ab20e0191a9290512d929b9fb4691749f8c21e9fd2901f2291c",
  "version": "3"
}
```


Check the signature