SOLIDIFIED

Audit Report for [TRONAVAIL]. 03,Nov, 2020.

TRON AVAIL Smart-Contract were found no vulnerabilities and no backdoors. The code was manually reviewed for all commonly known and more specific vulnerabilities.

So TRON AVAIL LIMITED Smart-Contract is safe for use in the main network.

Note: that is the only second part of the TRON AVAIL LIMITED project from their original team. Do not confuse with a lot of fakes.

www.solidified.io

The audit was conducted on commit **TK3oNcxMR9uu9WvXo8XygwDDNqyx3gDKEN**

The audit was based on the solidity compiler `0.5.0+commit.3155dd80`

https://tronscan.org/#/contract/TK3oNcxMR9uu9WvXo8XygwDDNqyx3gDKEN/

SOLIDIFIED

Audit Report for [TRONAVAIL]. 03,Nov, 2020.

Independent description of the smart-contract functionality:

The TRON AVAIL contract provides the opportunity to invest any amount in TRX (from 100 TRX) in the contract and get a 200% return on investment, if the contract balance has enough funds for payment.

Dividends are paid from deposits of users (Ponzi scheme).

It is allowed to participate in the project only from usual wallet (not smart-contract nor externally owner address).
You can create a Deposit by calling the “invest” function and attaching the required amount of TRX to the transaction (from 100 TRX inclusive).

Each subsequent Deposit is kept separately in the contract, in order to maintain the payment amount for each Deposit.

The daily percentage for user dividends starts from 1% and depends on the following factors:

Every 3,000,000 TRX on the maximum smart contract balance +0.05% until 10%. This Contract Bonus cannot decrease.

- Every 12 hours of non-withdrawal of dividends from the smart contract +0.05% until 5% (when creating new deposits, the percent keeps growing).

Maximum daily percent is 16% (1+10+5).

All dividends are calculated at the moment of request and available for withdrawal at any time.

Withdrawal is performed by calling the “withdraw” function from the address the Deposit was made.

Contract owners fee: part of the invested funds is sent to two addresses:

(marketing address) - 5%.
(the project address) - %.

The audit was conducted on commit **TK3oNcxMR9uu9WvXo8XygwDDNqyx3gDKEN**

SOLIDIFIED

Audit Report for [TRONAVAIL]. 03,Nov, 2020.

There is ten-level referral program: in the “invest” function, one can specify the address of the referrer.

As a result, the referrer (upline) will get direct transfer of share of the investor's Deposit according to the following table:

Referrer Level	1st	2nd	3rd
Percent %	5%	3%	1%

Requirements for the referrer: you can not specify your own wallet as a referrer, as well as a wallet that does not have at least one contribution in the smart contract. If wrong referrer is provided, no referrer is set.

The referrer is specified once at the time of the first deposit and is assigned to the user without the possibility of changing. From each subsequent Deposit, the referrer will get his percents.

Any user that has at least one contribution in the project can specify his own ‘refBackPercent’ - share of the referral bonus that will be returned to his direct referral (only 1 referral level).

To set refBackPercent user must call ‘setRefBackPercent’ function with percent parameter with 2 decimals (means 1% = 100, 100% = 10000).
The smart-contract has limits of total Invested value per 12 hours (since deploy of the smart contract):

Team Bonus on Partners Daily Income every ?me they make a withdrawal

---

Level	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Percent	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%

New level is activated for each direct partner, maximum 10 levels, see above.

The audit was conducted on commit **TK3oNcxMR9uu9WvXo8XygwDDNqyx3gDKEN**

SOLIDIFIED

Audit Report for [TRONAVAIL]. 03,Nov, 2020.

The contract contains statistical functions that do not require sending transactions:

- **function _refPayout(address _addr, uint256 _amount)**: To send payouts to 10 generations during withdrawal
- **function invest(address referrer):** Main function which receives the investment and allocates bonus to upline addresses
- **function withdraw():** Provide method to users for withdrawing their investment and profits from the contract. This method calculates all the types of profits available to users and then calculates the final amount available to be withdrawn from the contract by the user.
- **function getContractBalance() public view returns (uint256)**: Provides the current balance of the platform/contract.
- **function getContractBalanceRate() public view returns (uint256):** Provides the contract balance rate. This determines the bonus received by the users. Higher the platform balance, higher is the bonus available to users.
- **function getUserPercentRate(address userAddress) public view returns (uint256)**: Provide the user percent rate. This is determined by the last checkpoint when user deposited into the contract
- **function getUserDividends(address userAddress) public view returns (uint256)**: Provide the dividends available to the user on basis of multiple deposits.
- **function getUserAvailable(address userAddress) public view returns(uint256)**: Provide the balance which can be withdrawn by user. The same functionality is also provided by function getUserAvailableBalanceForWithdrawal.
- **function getUserCheckpoint(address userAddress) public view returns(uint256)**: Useds to provide the latest checkpoint of the address.
- **function getUserReferrer(address userAddress) public view returns(address)**: Provide the address who is the referrer of the current address.
- **function getUserReferralBonus(address userAddress) public view returns(uint256)**: The total amount of bonus that has been earned by the user.
- **function getUserAvailableBalanceForWithdrawal(address userAddress) public view returns(uint256)**: Provide the balance which can be withdrawn by user.
- **function isActive(address userAddress) public view returns (bool)**: Returns a boolean if the current address is active or not. A user is considered active if there is any pending amount that user can withdraw.

The audit was conducted on commit **TK3oNcxMR9uu9WvXo8XygwDDNqyx3gDKEN**

SOLIDIFIED

Audit Report for [TRONAVAIL]. 03,Nov, 2020.

The contract contains statistical functions that do not require sending transactions:

- **function getUserDepositInfo(address userAddress, uint256 index) public view returns(uint256, uint256, uint256)**: Provides the deposit details of the address. Three details are provided:
    - Amount: The amount of balance deposited
    - Withdrawn: The amount of balance withdrawn from that deposit.
    - Start: The time from when the holding started
- **function getUserAmountOfDeposits(address userAddress) public view returns(uint256)**: Returns the number of deposit transactions made by the address to the contract.
- **function getUserTotalDeposits(address userAddress) public view returns(uint256)**: Returns the total amount of balance that has been deposited by the address in the contract.
- **function getUserTotalWithdrawn(address userAddress) public view returns(uint256)**: Return the total amount of TRX that has been withdrawn by the address.
- **function isContract(address addr) internal view returns (bool)**: Returns if the address passed as the parameters is the contract address or a user address.
- **function getHoldBonus(address userAddress) public view returns(uint256)**: Get the holding bonus that is available till now.
- **function changeSpider(uint256 _roiPer) public returns(bool)**: To update the roi percentage of the contract. This can be performed only by the owner of the contract.
- **function getUserDownlineCount(address _addr) view external returns(uint256, uint256, uint256)**: Provide downline count of a user upto three levels. Downline count refers to the number of users who have used the referral of this account.
- **function getMatchBonus(address userAddress) public view returns(uint256)**: Provide current match bonus for a user.

The audit was conducted on commit **TK3oNcxMR9uu9WvXo8XygwDDNqyx3gDKEN**

This audit is not a call to participate in the project and applies only to the Smart-Contract code at the specified address.
Do not forget that you are doing all financial actions at your own risk, especially if you deal with high-risk projects.

Warning:

Beware of fake audits.

All official info available:
Website: www. solidified.io

If you have any questions or are interested in developing/auditing of Smart-Contracts, please contact us and we will consult you.

E-mail: info@solidified.io