FINTEH.ORG | AUDIT

AUDIT REPORT 2030101

JUSTTRON SOLIDITY CODE

HASH 6e2f4696a33cc663d9e013a9e9563c7977369e280c5494c3a2ac5b02da07df33

URL https://tronscan.org/#/contract/TTBYJiFWvSrFDXaZijK7scgVKFR1THnKXk/code

Introduction

Checks The Contract Code For Security Vulnerabilities And Bad Practices

Transaction Origin: 'tx.origin' used

Pos: 257: Check-effects-interaction: Potential violation of Checks-Effects-Interaction pattern in withdraw(): Could potentially lead to re-entrancy vulnerability.

Inline Assembly: Inline assembly used

Block Timestamp: Can be influenced by miners

Low Level Calls: Should only be used by experienced devs

Block Hash: Can be influenced by miners

Selfdestruct: Contracts using destructed contract can be broken

Gas Costs: Too high gas requirement of functions

This On Local Calls: Invocation of local functions via 'this'

Delete Dynamic Array: Use require/assert to ensure complete deletion

For Loop Over Dynamic Array: Iterations depend on dynamic array's size

Ether Transfer In Loop: Transferring Ether in a for/while/do-while loop

TRC20: 'decimals' should be 'uint8'

Proprietor of address can change code and balances

Constant/View/Pure Functions: Potentially constant/view/pure functions

Similar Variable Names: Variable names are too similar

withdraw() : Variables have very similar names "pool_bonuses" and "pool_bonus"

userInfo(address) : Variables have very similar names "pool_bonuses" and "pool_bonus"

userInfoTotals(address) : Variables have very similar names "total_deposited" and "total_deposits"

No Return: Function with 'returns' not returning

Guard Conditions: Ensure appropriate use of require/assert

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.

Pos: 138:61;143:28;

Result Not Used: The result of an operation not used

String Length: Bytes length != String length

Delete From Dynamic Array: 'delete' leaves a gap in array

Data Truncated: Division on int/uint values truncates the result

Data truncated:

Division of integer values yields an integer value again. That means e.g. 10 / 100 = 0 instead of 0.1 since the result is an integer again. This does not hold for division of (only) literal values since those yield rational constants.

Pos: 164:37; 171:27; 175:6; 177:59; 180:26; 188:11; 239:51; 260:87; 275:8; 346:148; 353:135; 353:166; 354:51; 354:82

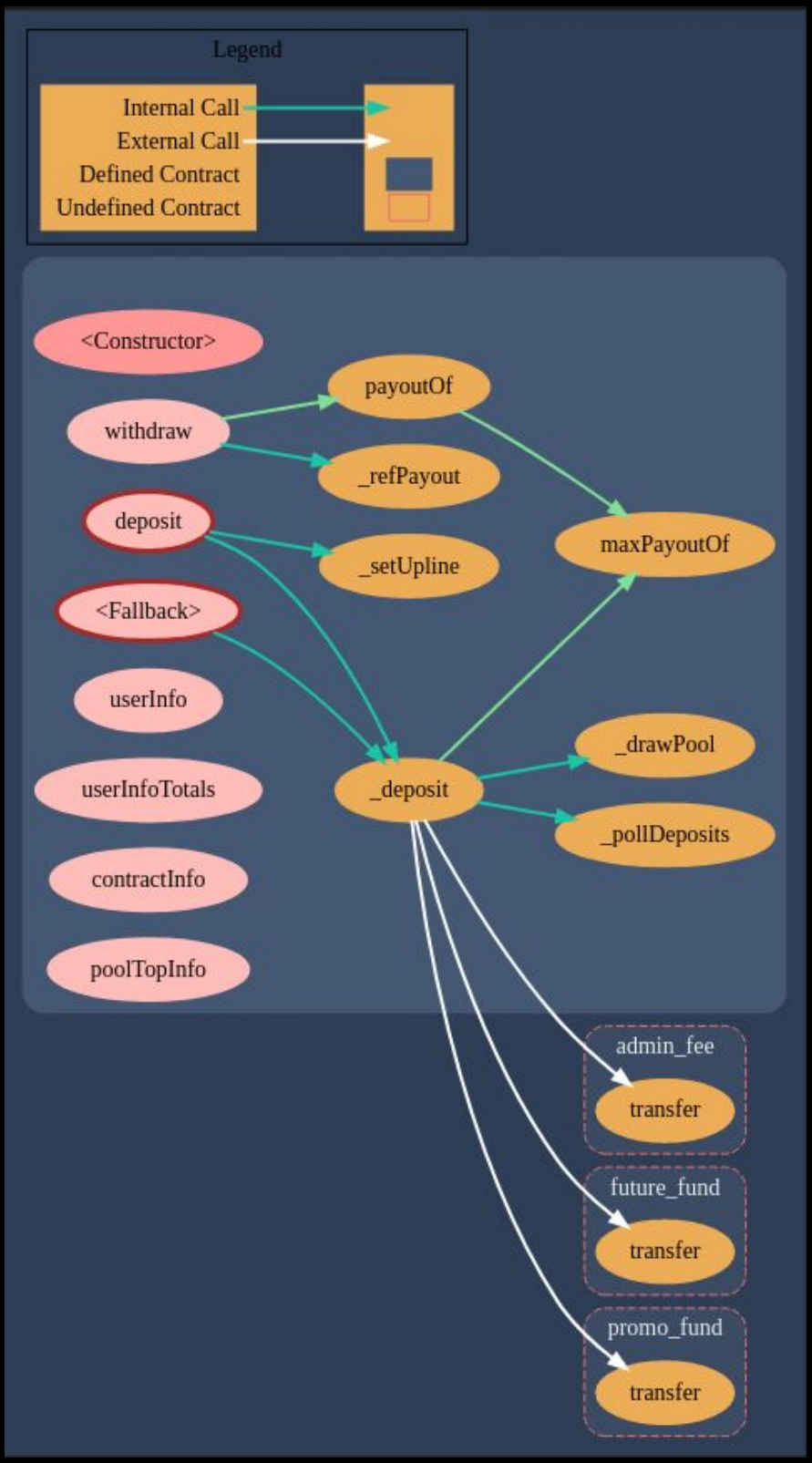
finteh.org recommended
https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/math/SafeMath.sol

Solidity unit testing

BallotTest

✓ Check winning proposal

✓ Check winning proposal with return value



1.1 Main scheme of contract JustTron

Logic functionality

Rewards distribution

The 400% is returned in 4 ways (1 passive and 3 via marketing) and when the 400% is accumulated through any of the 4 ways, a new deposit must be made equal or greater to continue receiving from the fund.

1.2% Daily return on your Deposit (maximum 400 days) 100% Passive.

Pos: 156; 170: *Source* _pollDeposits *probably is* _poolDeposits

Matching Commission on Partners Daily Income every time they make a withdrawal

1st generation 30%

2nd generation 10%

3rd generation 10%

4th generation 10%

5th generation 10%

6th generation 8%

7th generation 8%

8th generation 8%

9th generation 8%

10th generation 8%

11th generation 5%

12th generation 5%

13th generation 5%

14th generation 5%

15th generation 5%

16th generation 3%

17th generation 3%

18th generation 3%

19th generation 3%

20th generation 3%

1 new level is activated for each direct partner, maximum 20 levels, see above.

ALL Deposits set aside in pool, every 24 hour 10% of the pool is shared among top 10 sponsors in volume by

1st position 30%

2nd position 20%

3rd position 15%

4th position 10%

5th position 9%

6th position 5%

7th position 5%

8th position 3%

9th position 2%

10th position 1%

Minimum and maximum deposit limits

1st cycle, minimum deposit 50 TRX, up to 100 000 TRX.

2nd cycle, equal or greater than previous deposit, up to 300 000 TRX.

3rd cycle, equal or greater than previous deposit, up to 900 000 TRX.

4th cycle and beyond, equal or greater than previous deposit, up to 2 000 000 TRX.

sending TRX directly to the contract is not supported by logic

sending TRX to the contract from another contract is not supported by logic

All transactions will fail if the contract balance is less than the withdrawal amount, when the withdrawal function is called, because contract balance is not checked

Pos: 36 *This parameter is not used in any strings* (mapping(uint8 => address) public permanent_top;

finteh.org recommended

Collect garbage every non-usable parameters and merge withdrawal addresses for less gas

---

Backdoor for investor funds can be withdrawn by not owner

Bugs allowing to steal money from the contract

were not detected in code