TechRate

Blockchain solutions and consulting

ABC Project

Smart Contract Security Audit

January, 2021

TechRate

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn’t say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

TechRate was commissioned by ABC Project to perform an audit of smart contracts:

- *farming.sol*
- *token.sol*
- *staking.sol*

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Issues Checking Status

№.	Issue description.	Checking status
1	Compiler warnings.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	Issues indetified
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed

18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module.	Passed
21	Fallback function security.	Passed

Security Issues

High Severity Issues

1. Owner can withdraw any amount of tokens from contract (staking.sol contract)

Issue description:
Owner (which is set in the constructor) can withdraw any amount of tokens to any address from the contract.

Recommendation:
Owner should not have any ability to withdraw user tokens. Please, reconsider your business model to guarantee safety for the users.

2. Users will not be able to get any extra funds (farming.sol contract)

Issue description:
As soon as there is subtracting dividends withdraw amount from variable Investamount (iInvest.sol#L263) there will occur the situation when users get only their funds back and will not earn any extra funds, and some of it will be lost.

Because variable Investamount equal to total invested trx, that means after the users will get back their trx, they will not be able to get any extra dividends from project.

Recommendation:
Please recheck the logic of dividends paying.

Medium Severity Issues

1. Start and end time of daily invest round are the same (farming.sol contract)

Issue description:
Start and end of invest daily round in function Deposite are the same value and equal to now. But these variables should be different.(iInvest.sol#L150)

Because of that there happens the situation that nobody will be able to get any dividends using function withdraw for new invest rounds started using function Deposite.

Recommendation:
Please recheck the logic of this function.

2. There is no possibility to get invested tokens back (farming.sol contract)

Issue description:
Once invested, the user will not be able to get his invested funds back, because there is no such function for that in contract iInvest.sol.

Recommendation:
Please add some possibility for the user to withdraw his deposits.

3. Users dividends loss possible (iInvest.sol contract)

Issue description:
If smart contract will have less funds than user withdraw dividends, users will lose this difference (their dividends amount - contract balance) and will get only available contract balance.

Recommendation:
Please store somewhere this difference (their dividends amount - contract balance) so user will be able to withdraw it in future, when the contract will have enough funds.

4. Wrong referrer (staking.sol contract)

Issue description:
Function findEthReceiver will return the wrong receiver for the matrix equal to 4.

Recommendation:
Please use x3AutoMatrix instead of x3Matrix at line 550.

Low Severity Issues and Recommendations

1. Wrong naming

Issue description:
Variables, functions and struct names do not follow Solidity Style Guides. Please follow the instructions described by this link.

- Variable investdata name should use mixedCase.
- Struct iInvest name should use CapWord style.
- Variable lastsettledDailyIinvest name should use mixedCase.
- Variable activeroundId name should use mixedCase.
- Struct iInvestData name should use CapWord style.
- Variable dividendamount name should use mixedCase.
- Variable starttime name should use mixedCase.
- Variable MaxLimit name should use mixedCase.
- Variable Investamount name should use mixedCase.
- Variable iInvestdailyroundid name should use mixedCase.
- Variable Main_Contract name should use mixedCase.
- Function IinvestExternal name should use mixedCase
- Function Deposite name should use mixedCase
- Function WithdrawFundsiInvest name should use mixedCase
- Variable thirdlevelreferrals name should use mixedCase.
- Variable fourthlevelreferrals name should use mixedCase.
- Variable oldcontractaddress name should use mixedCase.
- Variable investdata name should use mixedCase.
- Variable lastsettledDailyIinvest name should use mixedCase.
- Variable activeroundId name should use mixedCase.
- Variable noofpayment name should use mixedCase.
- Variable lastsettledDailyGlobal name should use mixedCase.
- Variable noofpayment name should use mixedCase.
- Variable lastsettledDailyGlobal name should use mixedCase.
- Variable dividendamount name should use mixedCase.
- Variable investedtilldate name should use mixedCase.
- Variable dividendamount name should use mixedCase.
- Variable starttime name should use mixedCase.
- Variable todaysinvestmentX3 name should use mixedCase.
- Variable todaysinvestmentX4 name should use mixedCase.
- Variable todaysinvestment name should use mixedCase.
- Variable dailydividendtime name should use mixedCase.
- Variable Max_Limit_Global_Withdrawal name should use mixedCase.
- Variable GlobalDailyDataListX3 name should use mixedCase.
- Variable GlobalDailyDataListX4 name should use mixedCase.
- Variable dailydividendroundid name should use mixedCase.
- Variable iInvestdailyroundid name should use mixedCase.
- Variable Iinvest name should use mixedCase.

2. Extra variables

Issue description:
Smart contracts contain not used variables, which could be removed.

- Variable todaysinvestment has no usage and could be removed.
- Variable dailydividendtime has no usage and could be removed.
- Variable todaysinvestmentroi has no usage and could be removed.
- Variable currentStartingLevel has no usage and could be removed.
- Variable idToAddress has only one usage at constructor and has no sense, so it could be removed.

3. Zero address checking required

Issue description:
There is no zero address checking in the following functions. Please add zero address checking for protecting contract owners and investors from any losses. It could be done using following line:
`require(someAddress != address(0));`

- ❏ Variable ownerAddress in the constructor could be zero address.
- ❏ Variable _roundStarter in the constructor could be zero address.
- ❏ Variable _iinvest in the constructor could be zero address.
- ❏ Variable userAddress in function registration could be zero address.
- ❏ Variable referrer in function registration could be zero address.
- ❏ Variable ownerAddress in the constructor could be zero address.
- ❏ Variable oldaddress in the constructor could be zero address

4. Empty fallback function

Issue description:
Fallback function in the iInvest.sol contract is empty and any user can send his trx to this contract and lose his money.

Recommendation:
We recommend to revert any transfers to this contract except the main contract transfers.

5. No need in matrix variable in buyNewLevel function

Issue description:
As XGold contract uses only one matrix, there is no need in sending matrix variable to the function buyNewLevel , because it does not has any usage here.

Recommendation:
We recommend to remove matrix variable usages in this contract, as soon as this contract has only one matrix

6. Double call of same function in one execution

Issue description:
Function findFreeX12Referrer has been called two times in function buyLevel1 for calling the other function and for sending its result to the event.

Recommendation:
Please call this function one time and store its value in a local variable.

7. Same functions with same body

Issue description:
Functions usersX12AutoMatrix and usersX12Matrix have the same body and one of them could be removed.

Recommendation:
Please remove one of these functions.

8. Owner dividends

Issue description:
If some referrer address is zero address then its percent will be sent to the owner.

Recommendation:
Please remove that possibility, so the owner will not get tokens instead of zero address referrers.

9. Functions for sending Dividends and finding dividends receiver has word ETH in their name

Issue description:
Functions findEthReceiver, sendETHDividends and sendETHDividendsAuto have word ETH in their names, but the contract will send TRX.

Recommendation:
Please change the names of these functions.

10. Daily round wrong round id setting

Issue description:
In the function setDailyRound there is a wrong round id setting at lines 1035 and 1039.

Recommendation:
Setting round id should be increased by one.

Conclusion

Smart contracts contain some issues which should be fixed before deploying.