Security Audit

INTERSMART

0x6ab5E0f72c2Dc52b7F0f3D980034E80E6107F334

CRITICAL ISSUES (critical, high severity): 0

Bugs and vulnerabilities that enable theft of funds, lock access to funds without possibility to restore it, or lead to any other loss of funds to be transferred to any party; high priority unacceptable bugs for deployment at mainnet; critical warnings for owners, customers or investors.

ERRORS, BUGS AND WARNINGS (medium, low severity): 0

Bugs that can trigger a contract failure, with further recovery only possible through manual modification of the contract state or contract replacement altogether; Lack of necessary security precautions; other warnings for owners and users.

OPTIMIZATION POSSIBILITIES (very low severity): 0

Possibilities to decrease cost of transactions and data storage of Smart-Contracts.

NOTES AND RECOMMENDATIONS (very low severity): 1

Tips and tricks, all other issues and recommendations, as well as errors that do not affect the functionality of the Smart-Contract.

Conclusion:

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

So INTERSMART Smart-Contract is safe for use in the main network.

AUDIT RESULT:

Notes

1. Loop over a increasing variable (low severity):

In the 'getFreeReferrer, regUser' functions, the 'while' loop grows indefinitely as the number of invitees in structure increases before the upline for the new participant.

There is a block gas limit in the Ethereum blockchain, so the transaction size is limited. If this limit is exceeded, the transaction cannot be executed on the network at this point in time, so the user will not be able to personally invite referrals.

Take mention that the project has a limit on the withdrawal of profit - the user must personally invite at least 2 referrals, otherwise the withdrawal will not be available for him.

Independent description of the smart-contract functionality:

The INTERSMART smart contract is a referral project that distributes funds between uplines and invitees within a binary structures.

One can register in the project once by sending ETH to the smart contract address (or by calling the 'regUser' function and attaching the required amount):

The entry price is 0.1 ETH.

The owner's commission is 0.02 ETH.

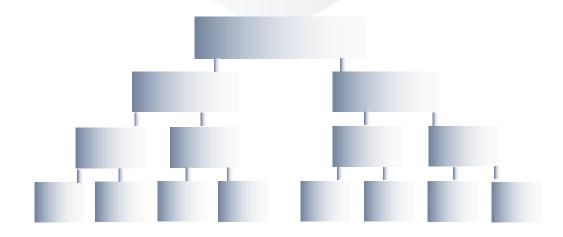
If you send a larger amount, the change will be returned automatically. In case of sending a smaller one, the transaction will be reverted and the funds will be returned to the user's wallet.

When registering, you must specify the upline user address under which you stand in the project. (when sending ETH, the address is specified in the HEX DATA of the transaction, in the case of calling the 'regUser' function - in the 'referrer' parameter).

The structure is binary (2 invitees under each user). Filled in from left to right, from top to bottom.

There are 4 "stages" in the project: when registering, the user gets to the first one, the transition to the next ones happens automatically with the increase of the structure under the user.

The overall structures of each stage in the project grow indefinitely.



Active user structure (in which payments happen):

1 stage: 63 places (including the user) - 6 levels - 32 users in the lower level - maximum 7 rounds of payments.

2 stage: 31 places (including the user) - 5 levels - 16 users in the lower level - maximum 7 rounds of payments

3 stage: 15 places (including the user) - 4 levels - 8 users in the lower level - maximum 8 rounds of payments

4 stage: 7 places (including the user) - 3 levels - 4 users in the lower level - maximum 10 rounds of payments

The user receives payments from the lowest level in active structure: half of the payments are accumulated for the autopayment of the next platform, one payment is held for reinvestment of the current stage (payments occur again in a nest round in the same stage), the remaining payments are available for withdrawal as profit.

Profit withdrawal is processed on request - you need to "send" 0 ETH to the address of the smart contract, or call the "withdraw" function. For successful withdrawal, you must personally invite at least 2 users to the project.

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

- 1. getFreeReferrer get free place in user's structure.
- 2. getContractBalance balance of smart-contract.
- 3. getAmountOfUsers amount of users in structure.
- 4. getDaysSinceStart how many days contract is working.
- 5. isRegistered if user registered or not.
- 6. getUserId ID of user on stage.
- 7. getUserById get user address by ID.
- 8. getUserReferrer upline.
- 9. getUserReferrers get list of uplines.
- 10. getUserPartners get personal invitees.
- 11. getUserAmountOfPartners amount of personal invitees.
- 12. getUserReferrals get 2 personal referrals of user.
- 13. getUserAmountOfReferrals get amount of personal referrals of user.
- 14. getUserProfit profit available to withdraw.
- 15. getUserRounds amount of rounds on stage of user.
- 16. getUserLevel highest opened stage by user.
- 17. getUserStages amount of rounds of all stages of user.
- 18. getStructure get structure on stage of user.

Juny 2021

Disclaimer:

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 on 3 resources only:

Website: www.grox.solutions

Telegram: www.t.me/groxsolutions (@groxsolutions)

YouTube: www.youtube.com/c/groxsolutions

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

Telegram: www.t.me/gafagilm (@gafagilm)

E-mail: info@grox.solutions