Telescr.in

Audit of smart contracts Eco Booster

Revision 4 dated 03.18.2021

Telescr.in

Contents

Brief information

Project: Eco booster
Network: TRON
Compiler version: 0.5.10
Optimization: enabled
The audit date: 03.18.2021

Information

The contract code was reviewed and analysed 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.

General conclusion

As a result of the audit, no errors were found that affect the security of users' funds on the contract. No obvious signs of an exit scam were found. No bugs and backdoors found either.

! Important note
This contract depends on Ticket token contract. Source code of that contract wasn't provided for review. So, conclusions are made using common sense but full system behavior depends only on contract owner will and actions.

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**
Unknown implementation of ticket token	Warning	Medium	High
Mutable ticket token	Warning	Medium	Medium
Mutable static rate	Warning	Medium	Medium
Mutable dynamic rate	Warning	Medium	Medium
Impossible to become “richman” with default setting	Warning	Medium	Medium
Mutable burn rate	Notice	Low	N/a
Node never deleted while upgrading	Notice	Medium	Low
Possible error in richman insurance pays when activeInsurance called partially	Notice	Low	Low
Manager 14 has less possibilities with fund managing then others	Remark	Low	N/a
Possible optimisation function activeInsurance	Remark	Low	N/a
activateNodeBurnAmount fixed value	Remark	Low	N/a
Max invest amount value	Remark	Low	N/a

A. Errors

Not found.

B. Warnings

1. Unknown implementation of ticket token (Line 875)

This contract relies on ticket token for deployed version on the moment of audit. Implementation of this ticket is unknown and is out of scope of current audit.
The contract uses burnFromUsdt, totalVending and vendingAndBurn method of ticket contract, investors should pay attention on its implementation.

2. Mutable ticket token (Line 875, 2110-2112)

This contract uses external ticket token with unknown behavior. However, even if the token is audited, it can be replaced by owned at any time.

3. Mutable Static rate (Line 890, 2086)

Static rate can be updated by contract owner at any time.

4. Mutable Dynamic rates (Line 892, 2094)

Dynamic rates can be updated by contract owner at any time.

5. Impossible to become “richman” with default setting (Line 942)

Investor can become “richman” and pretend to insurance payings in case when one invested at least 5000U. But by default, investment amount is limited to 3000U.

C. Notice

1. Mutable burning rate (Line 1245, 2078)

Burning rate can be updated by contract owner at any time.

2. Node never deleted while upgrading (Line 1722)

Node isn't deleted from previous pool when it's upgraded to next one. It will lead to unfair insurance distribution when node that was upgraded from L1 to L4 will receive insurance from all the levels.

3. Possible error in richman insurance pays when activeInsurance called partially (Line 1973)

Not all “richman” will get their insurance in one step of activeInsurance is skipped by some reason.

D. Remarks

1. Manager 14 has less possibilities with fund managing then others (Line 2155, 2170)

Manager 14 cannot work with operational and DEX funds.

2. Possible optimisation function activeInsurance (Line 1994)

You can use mod from safemath to determine remainder instead of (sub, div, mul series)

3. activateNodeBurnAmount fixed value (Line 1403)

This function always returns 3 000 000 000 (3000 tokens)

4. Max invest amount value (Line 961)

Max invest amount is initially defined as 100 000 000, although comment sais it's 2K

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.
Medium	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.
Medium	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:

23ea883eb99e2b2f571621e31ac9ee4a95a55971

Sha1 from bytecode (non-metadata):

daccc8c7ef148b84a45d7dbb5e7ff65c82585e39

Contract address:

TAW69gAS7eogMwZnow7k3Z3hWnz7ugeCHq

Check the digital print

Application. Signature of the audit report


```
{
          "address": "0x505ade8cea4db608250e503a5e8d4cb436044d2e",
          "msg": "As a result of the audit, no errors were found that affect the security of users' funds on the contract. No obvious signs of an exit scam
        were found. No bugs and backdoors found either. ! Important note This contract depends on Ticket token contract. Source code of that contract wasn't
        provided for review. So, conclusions are made using common sense but full system behavior depends only on contract owner will and actions. Sha1 from
        bytecode: 23ea883eb99e2b2f571621e31ac9ee4a95a55971 Sha1 from bytecode (non-metadata): daccc8c7ef148b84a45d7dbb5e7ff65c82585e39 Contract address:
        TAW69gAS7eogMwZnow7k3Z3hWnz7ugeCHq",
          "sig": "0xe77b2d1a410798acbf50d14e306c5131c3ba2c219145f0775f7cc7637e84626a76edccf62910e43a37e5bc6f3c957f7b3877c2821ee21aba3a1663d1ff9aa34e1c",
          "version": "3"
}
```


Check the signature