OIKOS

Smart contract audit

TABLE OF CONTENTS

INTRODUCTION

Our company provides comprehensive, independent smart contract auditing. We help stakeholders confirm the quality and security of their smart contracts using our standardized audit process. The scope of this audit was to analyze and document the Oikos contract.

AUDIT METHODOLOGY

1. Design Patterns

We inspect the structure of the smart contract, including both manual and automated analysis.

2. Static Analysis

The static analysis is performed using a series of automated tools, purposefully designed to test the security of the contract.
All the issues found by tools were manually checked (rejected or confirmed).

3. Manual Analysis

Contract reviewing to identify common vulnerabilities. Comparing of requirements and implementation. Reviewing of a smart contract for compliance with specified customer requirements. Checking for a gas optimization and self-documentation. Running tests of the properties of the smart contract in test net.

ISSUES DISCOVERED

Issues are listed from most critical to least critical. Severity is determined by an assessment of the risk of exploitation or otherwise unsafe behavior.

Severity Levels

- **Critical** - Funds may be allocated incorrectly, lost or otherwise result in a significant loss.
- **Medium** - Affects the ability of the contract to operate.
- **Low** - Minimal impact on operational ability.
- **Informational** - No impact on the contract.

AUDIT SUMMARY

The summary result of the audit performed is presented in the table below

Findings list:

LEVEL	AMOUNT
Critical	2
Medium	2
Low	2
Informational	2

Not suitable for deploying on mainnet!

ISSUES

(Most of the findings listed below is from the assumption that this contract is meant to be deployed on tron blockchain)

Critical

1. Ether contract addresses in contract for tron blockchain

   (Line 57,58,60,63, ArbRewarder.sol, Line 40, FeePool.sol)

   ## Description

   Ethereum contract address is present in multiple contracts. Tron blockchain cannot access ethereum blockchain and access these contracts

   ## Recommended

   Redeploy all those contracts in tron blockchain and replace the contract addresses with proper ones from tron.

2. Improper Access control

   (Line 259, FeePool.sol)

   ## Description

   Function '*closeCurrentFeePeriod()'* does not have any access specifier. Which means anyone can call the function and close the fee period, which is undesired.

   ## Recommended

   Add a modifier such as '*onlyOwner'* to restrict access to the function

Medium

1. Fallback incompatibility

   (Line 258, Depot.sol. Line 77, SynthetixAirdropper.sol)

   ## Description

   Tron blockchain does not support fallback functionality

   ## Recommended

   Either comment out or completely remove the fallback function from contract

2. Arithmetic logic error

   (Line 169, RewardDistribution.sol)

Description

While reducing length of '*distributions'* array, there is a chance where the array is empty, and applying decrement to 0 will lead to unwanted result

Recommended

Include a check for zero condition before reducing length of array

Low

1. Decimal change
   (Line 41, Synth.sol. Line 40, Synthetix.sol)
   ### Description
   Tron have fewer decimals (6) when compared to ethereum (18). It is better to use corresponding decimals
   ### Recommended
   Change decimal from 18 to 6
2. Limiting loop
   (Line 21, EscrowChecker.sol)
   ### Description
   Since we are storing *'_result'* in array of length 16, the loop should be restricted to prevent index out of bounds
   ### Recommended
   Restrict the value of '*schedules'* to never go above 8 (i*2 => 8*2 = 16)

Informational

1. (Line 231, Arbrewarder.sol) – Extra function interfaces which are not used inside the contract can be removed
2. (Line 23,33,41, DapMaintenace.sol) – These functions can be merged together to a single function with multiple parameters

CONCLUSION

Most of the findings are based on the assumption that this contract is meant to be deployed on Tron blockchain. Make the necessary changes before deploying on mainnet.