RAPID LABS

SMART CONTRACT SECURITY AUDIT

Final report **Plan: Simple**

Seed.photo

February 2024

RL@RAPIDLABS.FINANCE

RAPIDLABS.FINANCE

TABLE OF CONTENTS

INTRODUCTION

The report has been prepared for Seed.photo.

Name	Seed.photo
Audit date	2024-02-14 - 2024-02-14
Language	Solidity
Platform	Binance Smart Chain

ANALYZED CONTRACTS

Name	Address
SEED	0x6730f7A6BbB7b9C8e60843948f7FEB4B6a17B7F7

AUDIT PROCESS

Our audit structure consists of two stages:

Auto-analysis

- Our automated tools allow us to scan smart contract code and find potential issues
- We hand pick and verify all the issues found by the tools

Expert audit

- Manual analysis of potential issues and vulnerabilities
- Contract code is reviewed thoroughly

KNOWN ISSUES CHECKED

Title	Result
Unencrypted Private Data On-Chain	✓ passed
Code With No Effects	✓ passed
Message call with hardcoded gas amount	✓ passed
Typographical Error	✓ passed
DoS With Block Gas Limit	✓ passed
Presence of unused variables	✓ passed
Incorrect Inheritance Order	✓ passed
Requirement Violation	✓ passed
Weak Sources of Randomness from Chain Attributes	✓ passed
Shadowing State Variables	✓ passed
Incorrect Constructor Name	✓ passed
Block values as a proxy for time	✓ passed
Authorization through tx.origin	✓ passed

DoS with Failed Call	✓ passed
Delegatecall to Untrusted Callee	✓ passed
Use of Deprecated Solidity Functions	✓ passed
Assert Violation	✓ passed
State Variable Default Visibility	✓ passed
Reentrancy	✓ passed
Unprotected SELFDESTRUCT Instruction	✓ passed
Unprotected Ether Withdrawal	✓ passed
Unchecked Call Return Value	✓ passed
Floating Pragma	✓ passed
Outdated Compiler Version	✓ passed
Integer Overflow and Underflow	✓ passed
Function Default Visibility	✓ passed

ISSUE CLASSIFICATION

High risk	Issues leading to assets theft, locking or any other loss of assets or leading to contract malfunctioning.
Medium risk	Issues that can trigger a contract failure of malfunctioning.
Low risk	Issues that do now affect contract functionality. For example, unoptimised gas usage, outdated or unused code, code style violations, etc.

ISSUES

High risk issues

No issues were found

Medium risk issues

1. The owner can feeze tokens (SEED)

The owner of the contract can freeze the tokens.

```
function freezeAccount(address addr)
    public
    onlyOwner
{
  require(!frozenAccounts[addr], "Account is already Frozen!");
  require(addr != owner(), "Owner can not be frozen");
  frozenAccounts[addr] = true;
  emit AccountFrozen(addr);
}  // freezeAccount
```

Low risk issues

No issues were found

CONCLUSION

Seed.photo SEED contract was audited. 1 medium risk issue was found.

DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without RapidLabs prior written consent.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts RapidLabs to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

STATIC ANALYSIS

```
INFO:Detectors:
Different versions of Solidity are used:
- Version used: ['^0.8.0', '^0.8.4']
- ^0.8.0 (contracts/contract.sol#6)
- ^0.8.0 (contracts/contract.sol#33)
- ^0.8.0 (contracts/contract.sol#118)
- ^0.8.0 (contracts/contract.sol#203)
- ^0.8.0 (contracts/contract.sol#233)
- ^0.8.4 (contracts/contract.sol#622)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Context._msgData() (contracts/contract.sol#23-25) is never used and should be removed
ERC20._burn(address,uint256) (contracts/contract.sol#513-529) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.0 (contracts/contract.sol#6) allows old versions
Pragma version^0.8.0 (contracts/contract.sol#33) allows old versions
Pragma version^0.8.0 (contracts/contract.sol#118) allows old versions
Pragma version^0.8.0 (contracts/contract.sol#203) allows old versions
Pragma version^0.8.0 (contracts/contract.sol#233) allows old versions
Pragma version^0.8.4 (contracts/contract.sol#622) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
SEED.constructor() (contracts/contract.sol#634-636) uses literals with too many digits:
- _mint(msg.sender,1826000000 * 10 ** decimals()) (contracts/contract.sol#635)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Slither:. analyzed (6 contracts with 85 detectors), 10 result(s) found
```

RAPID LABS

YOUR PROJECT SECURED

RL@RAPIDLABS.FINANCE

RAPIDLABS.FINANCE