Smart contract security audit report

Ocean smart contract security audit report

Audit Team： Noneage security team

Audit date： March 7, 2021

Ocean Smart Contract Security Audit Report

1. Overview

On March 6, 2021, the security team of LS Technology received the security audit request of the **Ocean project**. The team will conduct a report on the **Ocean smart contract** from March 6, 2021 to March 7, 2021. During the audit process, the security audit experts of Zero Hour Technology communicate with the relevant interface people of the Ocean project, maintain information symmetry, conduct security audits under controllable operational risks, and try to avoid project generation and operation during the test process. Cause risks.

Through communicat and feedback with Ocean project party, it is confirmed that the loopholes and risks found in the audit process have been repaired or within the acceptable range. The result of this Ocean smart contract security audit: **passed.**

Audit Report MD5：AE3070EF6A7D358860E529A2688FBD9D

2. Background

2.1 Project Description

> Project name: Ocean
>
> official website: https://oceandao.finance/
>
> Contract type: DeFi Token contract
>
> Code language: Solidity
>
> Official GitHub repository address: https://github.com/oceanswap/ocean-core/tree/audit
>
> Contract documents: Ocean.sol，OceanPool.sol，LPTokenWrapper.sol

2.2 Audit Range

Ocean officially provides contract and MD5：

Ocean.sol	24a950aaf94011d6b58a39026c68ab3d
OceanPool.sol	d41e96bab1230fbebe17b7f61ab5d0c2
LPTokenWrapper.sol	500f255429d1d3e686e1383ac97f61b4

2.3 Security Audit List

The security experts of Noneage Technology conduct security audits on the security audit list within the agreement, The scope of this smart contract security audit does not include new attack methods that may appear in the future, does not include the code after contract upgrades or tampering, and is not included in the subsequent cross-country, does not include cross-chain deployment, does not include project front-end code security and project platform server security.

This smart contract security audit list includes the following:

- Integer overflow
- Reentry attack
- Floating point numbers and numerical precision
- Default visibility
- Tx.origin authentication
- Wrong constructor
- Return value not verified
- Insecure random numbers
- Timestamp dependency
- Transaction order is dependent
- Delegatecall
- Call
- Denial of service
- Logic design flaws
- Fake recharge vulnerability
- Short address attack
- Uninitialized storage pointer
- Additional token issuance
- Frozen account bypass
- Access control
- Gas usage

3. Contract Structure Analysis

3.1 Directory Structure

```
└─contracts
  │  Ocean.sol
  ├─distribution
  │      OceanPool.sol
  └─token
        LPTokenWrapper.sol
```

3.2 OceanPool contract

Contract

OceanPool

- shutdown()
- lastTimeRewardApplicable()
- rewardPerToken()

- earned(address account)
- stake(uint256 amount)
- withdraw(uint256 amount)
- exit()
- getReward()
- transferBack(address back, uint256 amount)
- notifyRewardAmount(uint256 reward)
- earnedShutdown(address account)
- rewardPerTokenShutdown()

3.3 LPTokenWrapper contract

Contract

LPTokenWrapper

- totalSupply()
- balanceOf(address account)
- stake(uint256 amount)
- withdraw(uint256 amount)

3.4 Ocean contract

Contract

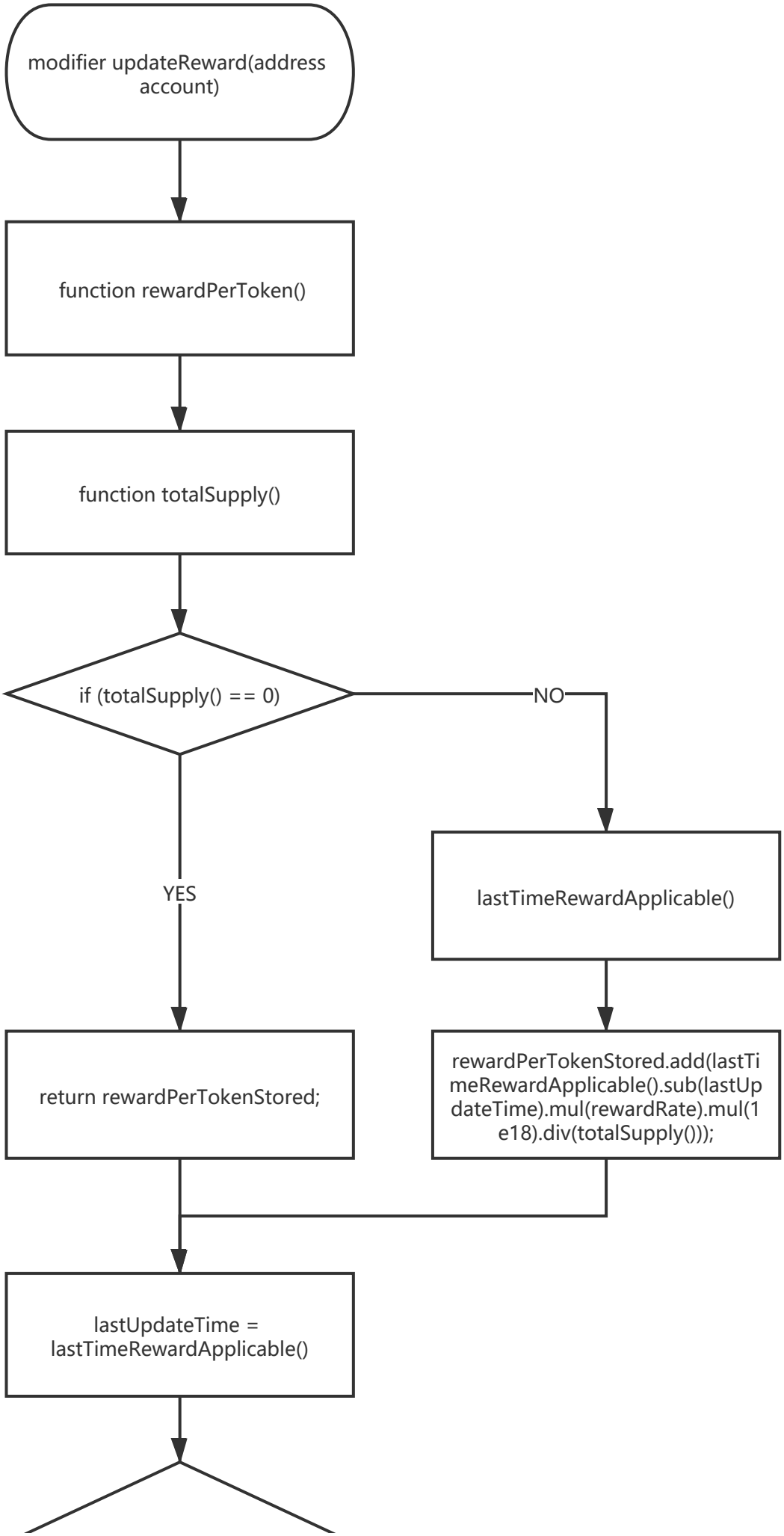
Ocean

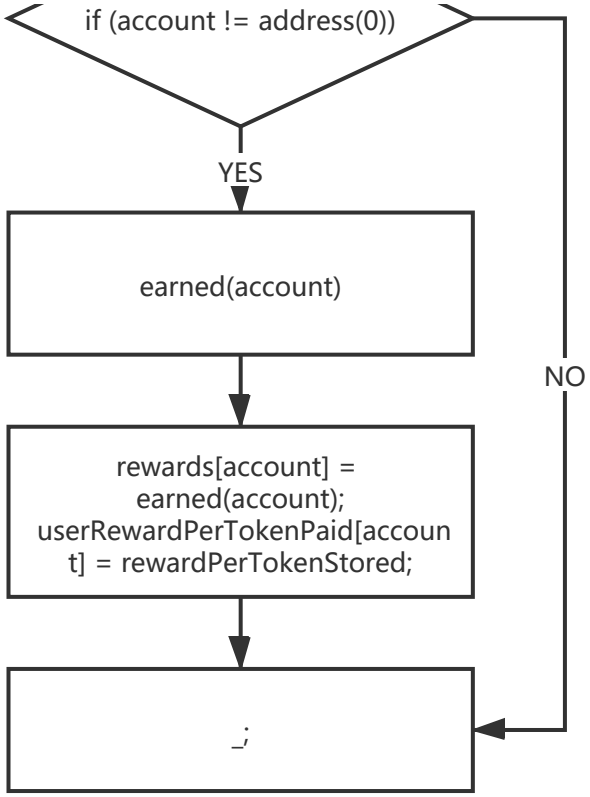
- mint(address recipient*, uint256 amount*)
- burn(uint256 amount)
- burnFrom(address account, uint256 amount)
- _beforeTokenTransfer(address from, address to, uint256 amount)

3.5 Contract Logic Flow Chart

Through the security audit of the **Ocean contract**, the security auditor listed the code flow chart of part of the contract logic in the audit process, as follows:

Part of OceanPool contract logic: updateReward()


modifier updateReward(address account)
function rewardPerToken()
function totalSupply()
if (totalSupply() == 0)
NO
YES
lastTimeRewardApplicable()
return rewardPerTokenStored;
rewardPerTokenStored.add(lastTimeRewardApplicable().sub(lastUpdateTime).mul(rewardRate).mul(1e18).div(totalSupply()));
lastUpdateTime = lastTimeRewardApplicable()

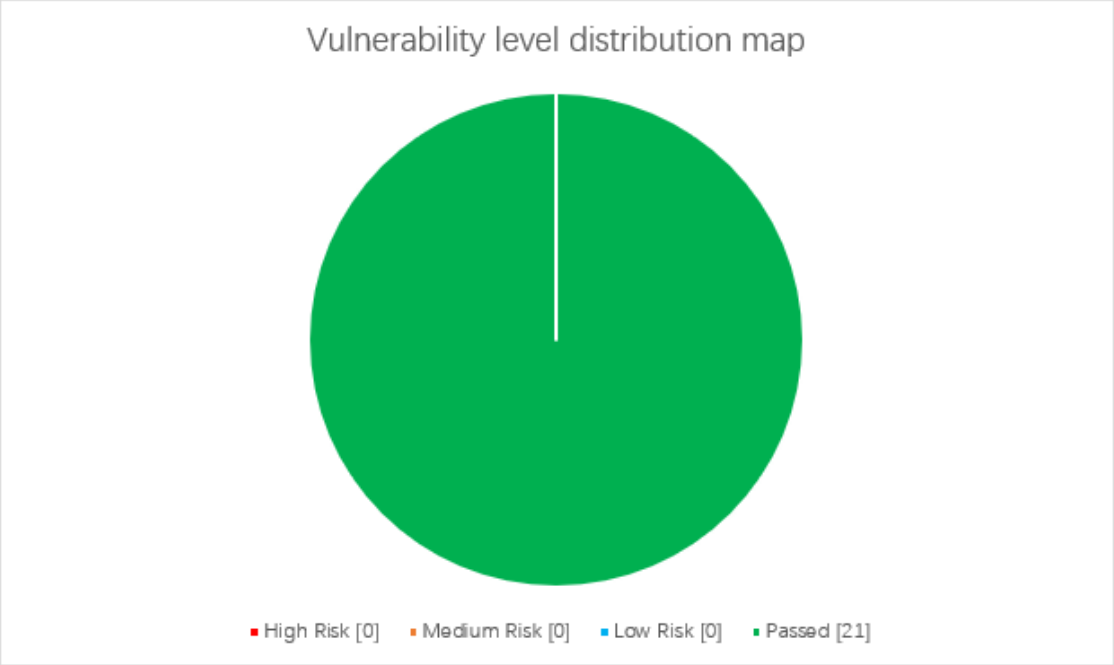


4. Audit Details

4.1 Vulnerabilities Distribution

Vulnerabilities in this security audit are distributed by risk level, as follows:

Vulnerability level distribution			
High risk	Medium risk	Low risk	Passed
0	0	0	21



This smart contract security audit has 0 high-risk vulnerabilities, 0 medium-risk vulnerabilities, 0 low-risk vulnerabilities, and 21 passed, with a high security level.

4.2 Vulnerabilities Details

A security audit was conducted on the smart contract within the agreement, and no security vulnerabilities that could be directly exploited and generated security problems were found, and the security audit was passed.

4.3 Other Risks

Other risks refer to the code that smart contract security auditors consider to be risky. Under certain circumstances, it may affect the stability of the project, but it cannot constitute a security issue that directly harms.

4.3.1 Excessive admin rights

- **Issue causes**

If the administrator's authority in the smart contract is large, when the private key of the administrator's account is accidentally lost or manipulated by malicious people, it may affect the stability of the project.

- **Question detail**

Through auditing the contract, it was discovered that part of the contract's settings, updates, and logical operation functions are operated by the manager, which poses a safety hazard. If the administrator's account private key is accidentally lost or manipulated by malicious people, the stability of the project will be affected.

- **Safety advice**

It is necessary to store the private key of the deployer's address safely and effectively to avoid loss or acquisition by malicious persons and minimize the risk.

5. Security Audit Tool

Tool name	Tool Features
Oyente	Can be used to detect common bugs in smart contracts
securify	Common types of smart contracts that can be verified
MAIAN	Multiple smart contract vulnerabilities can be found and classified
Noneage Internal Toolkit	Noneage Technology Internal Security Audit Toolkit + https://audit.noneage.com

6. Vulnerability assessment criteria

Vulnerability level	Vulnerability description
High risk	Vulnerabilities that can directly lead to the loss of contracts or users' digital assets, such as integer overflow vulnerabilities, false recharge vulnerabilities, re-entry vulnerabilities, illegal token issuance, etc. Vulnerabilities that can directly cause the ownership change of the token contract or verification bypass, such as: permission verification bypass, call code injection, variable coverage, unverified return value, etc. Vulnerabilities that can directly cause the token to work normally, such as denial of service vulnerabilities, insecure random numbers, etc.
Medium risk	Vulnerabilities that require certain conditions to trigger, such as vulnerabilities triggered by the token owner's high authority, and transaction sequence dependent vulnerabilities. Vulnerabilities that cannot directly cause asset loss, such as function default visibility errors, logic design flaws, etc.
Low risk	Vulnerabilities that are difficult to trigger, or vulnerabilities that cannot lead to asset loss, such as vulnerabilities that need to be triggered at a cost higher than the benefit of the attack, cannot lead to incorrect coding of security vulnerabilities.

Disclaimer:

Noneage Technology only issues a report and assumes corresponding responsibilities for the facts that occurred or existed before the issuance of this report, Since the facts that occurred after the issuance of the report cannot determine the security status of the smart contract, it is not responsible for this.
Noneage Technology conducts security audits on the security audit items in the project agreement, and is not responsible for the project background and other circumstances, The subsequent on-chain deployment and operation methods of the project party are beyond the scope of this audit.
This report only conducts a security audit based on the information provided by the information provider to Noneage at the time the report is issued, If the information of this project is concealed or the situation reflected is inconsistent with the actual situation, Noneage Technology shall not be liable for any losses and adverse effects caused thereby.

零时科技
NONEAGE

Telephone：86-17391945345 18511993344
Email ：support@noneage.com
Site ：www.noneage.com
Weibo ：weibo.com/noneage