# 智能合约安全审计报告


## Cradle 智能合约安全审计报告

审计团队：零时科技安全团队

时间： 2021-03-15

# Cradle智能合约安全审计报告

## 1.概述

零时科技安全团队于2021年03月12日，接到 **Cradle项目**的安全审计需求，团队于2021年03月15日完成了 **Cradle智能合约** 的安全审计，审计过程中零时科技安全审计专家与Cradle项目相关接口人进行沟通，保持信息对称，在操作风险可控的情况下进行安全审计工作，尽量规避在测试过程中对项目产生和运营造成风险。

经过与Cradle项目方沟通反馈确认审计过程中发现的漏洞及风险均已修复或在可承受范围内，本次Cradle智能合约安全审计结果：通过安全审计。

合约报告MD5：DB2865B5AAC6CC93E455BC7620929994

## 2.项目背景

### 2.1 项目简介

> 项目名称：Cradle
>
> 项目官网：https://cradle.finance/
>
> 合约类型：代币合约
>
> 代码语言： Solidity
>
> 合约文件： Cradle.sol

### 2.2 审计范围

Cradle官方提供合约文件及MD5：

Cradle.sol　　e39958c72fb31f2de43e89f8c8374e3a

### 2.3 安全审计项

零时科技安全专家对约定内的安全审计项目进行安全审计，本次智能合约安全审计的范围，不包含未来可能出现的新型攻击方式，不包含合约升级活篡改后的代码，不包括在后续跨链部署时的操作过程，不包含项目前端代码安全与项目平台服务器安全。

本次智能合约安全审计项目包括如下：

- 整数溢出
- 重入攻击
- 浮点数和数值精度
- 默认可见性
- Tx.origin身份验证

- 错误的构造函数
- 未验证返回值
- 不安全的随机数
- 时间戳依赖
- 交易顺序依赖
- Delegatecall调用
- Call调用
- 拒绝服务
- 逻辑设计缺陷
- 假充值漏洞
- 短地址攻击
- 未初始化的存储指针
- 代币增发
- 冻结账户绕过
- 权限控制
- Gas使用

# 3.合约架构分析

## 3.1 目录结构

```
└─Cradle
    Cradle.sol
```

## 3.2 Cradle合约

### Contract

**Context**

- _msgSender()
- _msgData()

**Ownable**

- owner()
- renounceOwnership()
- transferOwnership(address newOwner)

**Operator**

- operator()
- isOperator()
- transferOperator(address newOperator_)
- *transferOperator(address newOperator*)

**Cradle**

- setStableToken(address money, bool result)
- setFeeAccount(address _feeAccount)
- setFeeRate(uint256 _feeRate)
- projectInfo(address projectId)
- create(address stock,address money,uint256 stockAmount,uint256 moneyAmount,uint256[] memory times)

- buy(address projectId, uint256 targetStockAmount)
- userBuyInfo(address projectId, address user)
- userRelease(address projectId)
- release(address projectId) public updateProject(projectId)
- transferOut(bool isStock,address projectId,address to,uint256 amount)
- transferIn(bool isStock,address projectId,address from,uint256 amount)
- _safeTransfer(address token,address to,uint256 value)
- safeTransferFrom(address token,address from,uint256 value)
- _safeTransferFrom(address token,address from,address to,uint256 value)

### Interface

#### IERC20

- totalSupply()
- balanceOf(address account)
- transfer(address recipient, uint256 amount)
- allowance(address owner, address spender)
- approve(address spender, uint256 amount)
- transferFrom(address sender, address recipient, uint256 amount)

### Library

#### Math

#### SafeMath

#### SafeERC20

#### Address

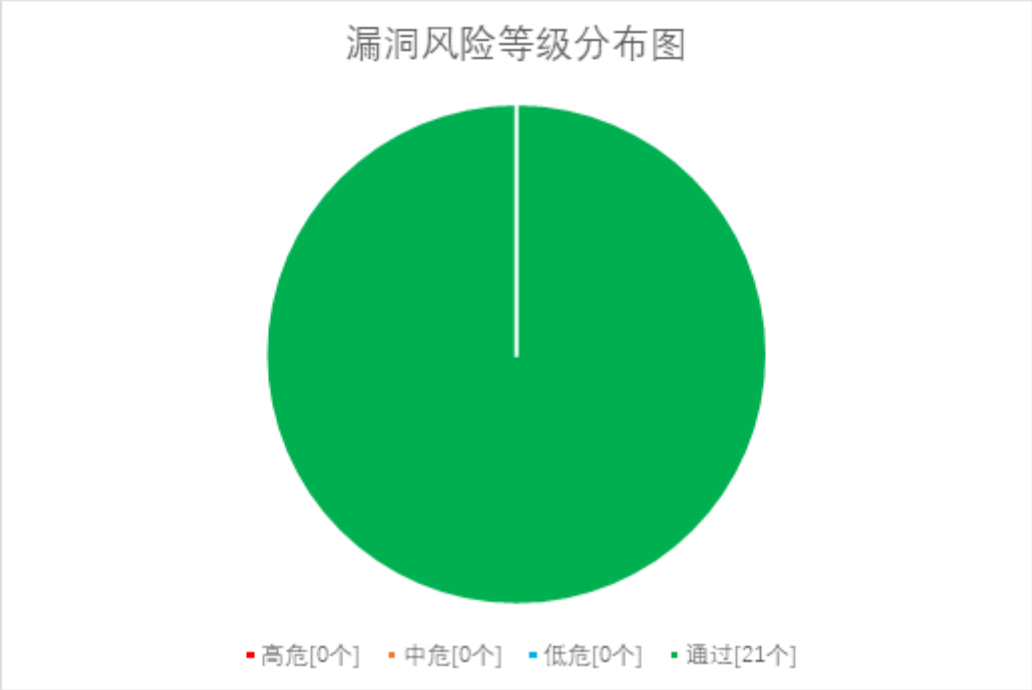
- isContract(address account)
- functionCall(address target, bytes memory data)
- functionCall(address target,bytes memory data,string memory errorMessage)
- functionCallWithValue(address target,bytes memory data,uint256 value)
- functionCallWithValue(address target,bytes memory data,uint256 value,string memory errorMessage)
- functionStaticCall(address target, bytes memory data)
- functionStaticCall(address target,bytes memory data,string memory errorMessage)
- functionDelegateCall(address target, bytes memory data)
- functionDelegateCall(address target,bytes memory data,string memory errorMessage)
- _verifyCallResult(bool success,bytes memory returndata,string memory errorMessage)

# 4.审计详情

## 4.1 漏洞分布

本次安全审计漏洞风险按危险等级分布：

| 漏洞风险等级分布 | | | |
|---|---|---|---|
| 高危 | 中危 | 低危 | 通过 |
| 0 | 0 | 0 | 21 |




本次智能合约安全审计高危漏洞0个，中危0个，低危0个，通过21个，安全等级高。

## 4.2 漏洞详情

对约定内的智能合约进行安全审计，未发现可以直接利用并产生安全问题的安全漏洞，通过安全审计。

## 4.3 其他风险

其他风险是指合约安全审计人员认为有风险的代码，在特定情况下可能会影响项目稳定性，但不能构成直接危害的安全问题。

### 4.3.1 管理员权限安全问题

- **发生原因**

智能合约中的存在管理员设置核心变量，当管理员密钥丢失或者被恶意人员控制，则会影响项目的正常运行。

- **问题点**

通过审计合约发现Cradle合约中部分设置由管理员控制，如果管理员密钥丢失或者被恶意人员控制，会影响项目稳定性。具体代码如下：

```
function setStableToken(address money, bool result) public onlyOwner {
    stableTokens[money] = result;
}

function setFeeAccount(address _feeAccount) public onlyOwner {
    feeAccount = _feeAccount;
}

function setFeeRate(uint256 _feeRate) public onlyOwner {
    feeRate = _feeRate;
}
```

- **安全建议**

需安全有效存储部署者地址私钥，避免丢失或者被恶意人员获取。

# 5.安全审计工具

| 工具名称 | 功能 |
| --- | --- |
| Oyente | 可以用来检测智能合约中常见bug |
| securify | 可以验证以太坊智能合约的常见类型 |
| MAIAN | 可以查找多个智能合约漏洞并进行分类 |
| 零时内部工具包 | 零时(鹰眼系统)自研发工具包+https://audit.noneage.com |

# 6.漏洞风险评估标准

| 漏洞等级 | 漏洞风险描述 |
| --- | --- |
| 高危 | 能直接导致代币合约或者用户数字资产损失的漏洞，比如：整数溢出漏洞、假充值漏洞、重入漏洞、代币违规增发等。 能直接造成代币合约所属权变更或者验证绕过的漏洞，比如：权限验证绕过、call代码注入、变量覆盖、未验证返回值等。 能直接导致代币正常工作的漏洞，比如：拒绝服务漏洞、不安全的随机数等。 |
| 中危 | 需要一定条件才能触发的漏洞，比如代币所有者高权限触发的漏洞，交易顺序依赖漏洞等。不能直接造成资产损失的漏洞，比如函数默认可见性错误漏洞，逻辑设计缺陷漏洞等。 |
| 低危 | 难以触发的漏洞，或者不能导致资产损失的漏洞，比如需要高于攻击收益的代价才能触发的漏洞 无法导致安全漏洞的错误编码问题。 |

零时科技
NONEAGE

咨询电话：86-17391945345 18511993344
邮　　箱：support@noneage.com
官　　网：www.noneage.com
微　　博：weibo.com/noneage