智能合约安全审计报告

BookFinance 智能合约审计报告

审计团队：零时科技安全团队

时间： 2021-02-26

BookFinance智能合约安全审计报告

1.概述

零时科技安全团队于2021年02月26日，对BookFinance智能合约进行了安全审计，审计过程中跟相关接口人进行沟通，保持了信息对称，在操作风险可控的情况下进行安全测试工作，规避了在测试过程中对产生和运营造成风险。

通过对BookFinance智能合约安全审计，此合约安全等级高。

合约报告MD5：**837BA35C9DBE9203393DF011575262AE**

2.项目背景

2.1 项目简介

> 项目名称：BookFinance
>
> 合约类型：代币合约
>
> 代码语言： Solidity
>
> 官方GitHub仓库地址：https://github.com/book-finance/book-core/tree/audit
>
> 合约文件： BBS2BBSPool，BKK1BBS2BBSPool，DOT1BBS2BBSPool，HBO1BBS2BBSPool，HT1BBS2BBSPool，MDX1BBS2BBSPool，USDT1BBS2BBSPool，Share，LPTokenWrapper。

2.2 审计范围

BookFinance官方提供合约文件及对应MD5：

BBS2BBSPool.sol	ef4e25981367c538b744ab38d2c4c0ae
BKK1BBS2BBSPool.sol	46dda0bb1f8141e7bedc303ffbe3d2de
DOT1BBS2BBSPool.sol	ee47398063d2f35a2b7ddfc68f2f162f
HBO1BBS2BBSPool.sol	7faf598ece6e34442e0b9c8acb5137dd
HT1BBS2BBSPool.sol	c351e982eeaf862f40f851ec3dc5d212
MDX1BBS2BBSPool.sol	73444f3476850387b1611f82117758c1
USDT1BBS2BBSPool.sol	2b36a19d95a645f82aff8225246c1ab5
Share.sol	89c99e5792620206da1623b894fc32dc
LPTokenWrapper.sol	500f255429d1d3e686e1383ac97f61b4

2.3 安全审计项

- 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

```
└─contract
    BBS2BBSPool.sol
    BKK1BBS2BBSPool.sol
    DOT1BBS2BBSPool.sol
    HBO1BBS2BBSPool.sol
    HT1BBS2BBSPool.sol
    LPTokenWrapper.sol
    MDX1BBS2BBSPool.sol
    Share.sol
    USDT1BBS2BBSPool.sol
```

3.2 BBS2BBSPool合约

Contract

BBS2BBSPool

- shutdown()
- lastTimeRewardApplicable()
- rewardPerToken()
- earned(address account)
- stake(uint256 amount)
- withdraw(uint256 amount)
- exit()

- getReward()
- transferBack(address back, uint256 amount)
- notifyRewardAmount(uint256 reward)

3.3 BKK1BBS2BBSPool合约

Contract

BKK1BBS2BBSPool

- shutdown()
- lastTimeRewardApplicable()
- rewardPerToken()
- earned(address account)
- stake(uint256 amount)
- withdraw(uint256 amount)
- exit()
- getReward()
- transferBack(address back, uint256 amount)
- notifyRewardAmount(uint256 reward)

3.4 DOT1BBS2BBSPool合约

Contract

DOT1BBS2BBSPool

- shutdown()
- lastTimeRewardApplicable()
- rewardPerToken()
- earned(address account)
- stake(uint256 amount)
- withdraw(uint256 amount)
- exit()
- getReward()
- transferBack(address back, uint256 amount)
- notifyRewardAmount(uint256 reward)

3.5 HBO1BBS2BBSPool合约

Contract

HBO1BBS2BBSPool

- shutdown()
- lastTimeRewardApplicable()
- rewardPerToken()
- earned(address account)
- stake(uint256 amount)
- withdraw(uint256 amount)

- exit()
- getReward()
- transferBack(address back, uint256 amount)
- notifyRewardAmount(uint256 reward)

3.6 HT1BBS2BBSPool合约

Contract

HT1BBS2BBSPool

- shutdown()
- lastTimeRewardApplicable()
- rewardPerToken()
- earned(address account)
- stake(uint256 amount)
- withdraw(uint256 amount)
- exit()
- getReward()
- transferBack(address back, uint256 amount)
- notifyRewardAmount(uint256 reward)

3.7 MDX1BBS2BBSPool合约

Contract

MDX1BBS2BBSPool

- shutdown()
- lastTimeRewardApplicable()
- rewardPerToken()
- earned(address account)
- stake(uint256 amount)
- withdraw(uint256 amount)
- exit()
- getReward()
- transferBack(address back, uint256 amount)
- notifyRewardAmount(uint256 reward)

3.8 USDT1BBS2BBSPool合约

Contract

USDT1BBS2BBSPool

- shutdown()
- lastTimeRewardApplicable()
- rewardPerToken()
- earned(address account)
- stake(uint256 amount)

- withdraw(uint256 amount)
- exit()
- getReward()
- transferBack(address back, uint256 amount)
- notifyRewardAmount(uint256 reward)

3.9 LPTokenWrapper合约

Contract

LPTokenWrapper

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

3.10 Share合约

Contract

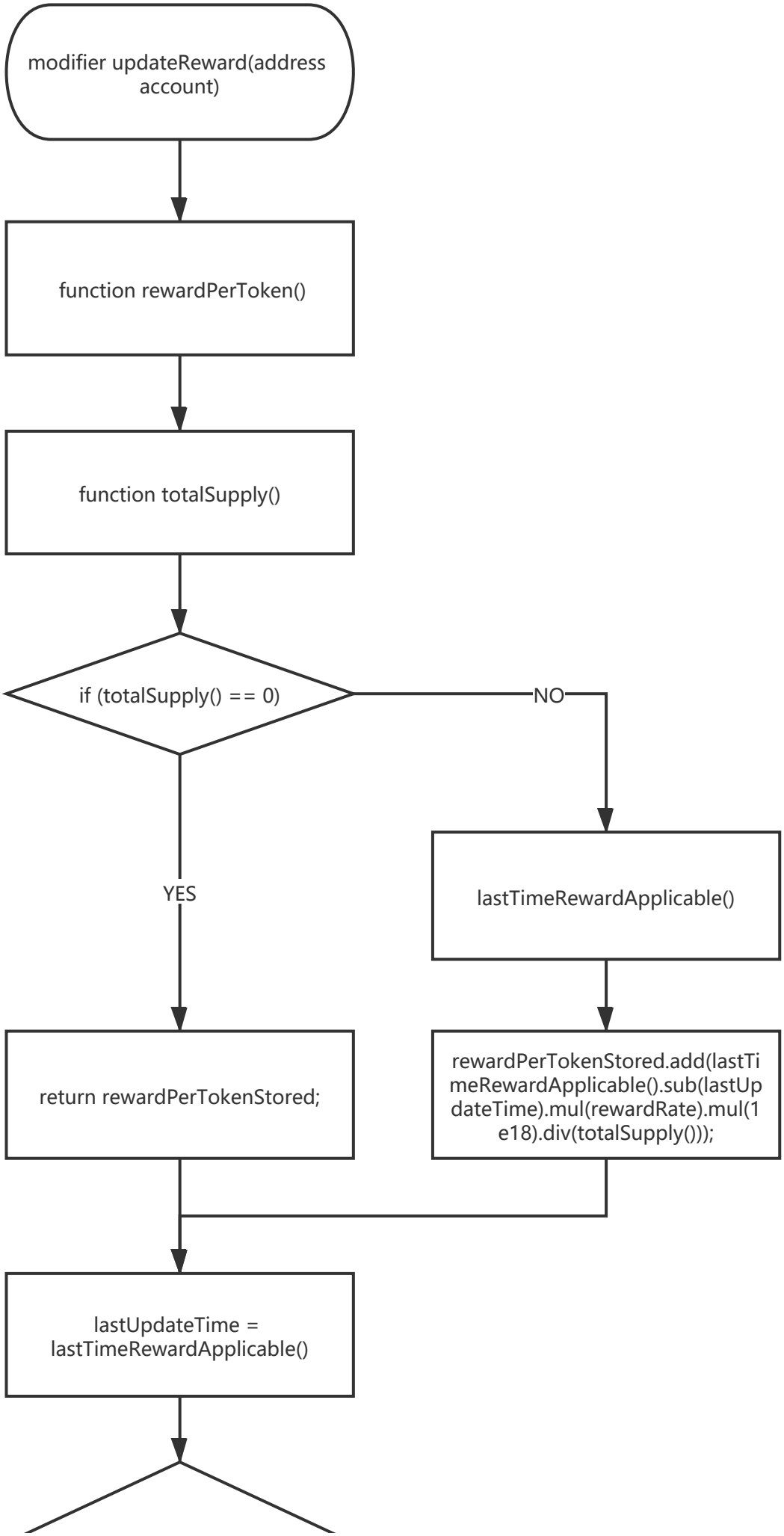
Share

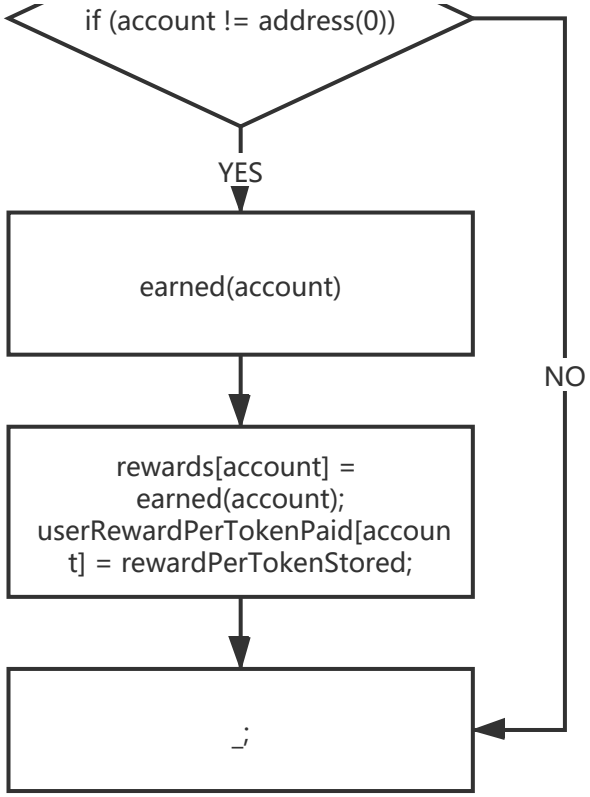
- mint(address recipient*, uint256 amount*)
- burn(uint256 amount)
- burnFrom(address account, uint256 amount)
- _beforeTokenTransfer(address from, address to, uint256 amount)

3.5 合约部分逻辑流程图

通过对BookFinance合约的安全审计，安全审计人员列出了审计过程中部分合约逻辑的代码流程图，如下：

BBS2BBSPool合约部分逻辑：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.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

通过安全审计，此项目无安全漏洞。

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 管理员权限较大

发生原因

如果智能合约中的管理员权限较大，当管理者账户私钥不慎丢失或者被恶意人员所操控，可能会影响项目稳定性。

问题点

通过审计合约发现，合约中部分的设置，更新，逻辑操作功能由管理者来操作，存在安全隐患。如果管理者账户私钥不慎丢失或者被恶意人员所操控，将会影响项目稳定性。

安全建议

管理员账户私钥应多份并妥善保存，目前已和团队进行沟通并处理风险，将风险降到最低。

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	零时科技内部审计工具包+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。

资讯电话：86-17391945345

邮　　箱：support@noneage.com

官　　网：www.noneage.com

微　　博：weibo.com/noneage