

# Roolend 审计报告

Version 1.0.0

报告编号: 2021061400011018

灵踪安全发布

2021年6月14日



灵踪安全  
FAIRYPROOF

# 01. 介绍

本报告包含了灵踪安全在Roolend团队要求下对Roolend项目的合约源代码进行审计的结果。

**项目通证名:**

ROO

**项目通证所在的区块链链上地址:**

暂无

**合约所在的Github仓库地址:**

<https://github.com/roolend-finance/contracts>

**合约在Github中的commit编号:**

3c4b5ce99308d7d25b2abd3e4a83b92715e7336d

**合约所在的区块链链上地址:**

暂无

**合约文件及目录结构:**

所审计合约的文件名及目录结构为：

```
contracts/
├── Comptroller
│   ├── Comptroller.sol
│   ├── ComptrollerInterface.sol
│   ├── ComptrollerStorage.sol
│   └── unitroller.sol
├── Controller
│   ├── Context.sol
│   ├── Controller.sol
│   └── Ownable.sol
├── InterestRateModel
│   ├── InterestRateModel.sol
│   └── whitePaperInterestRateModel.sol
├── Math
│   ├── CarefulMath.sol
│   ├── Exponential.sol
│   └── SafeMath.sol
├── PriceOracle
│   ├── DexPrice.sol
│   ├── IDexPair.sol
│   ├── PriceOracle.sol
│   ├── PriceOracleProxy.sol
│   └── SimplePriceOracle.sol
└── ROO
    └── OIP20
```

```
|   |   |   OIP20.sol
|   |   |   OIP20Burnable.sol
|   |   |   OIP20Detailed.sol
|   |   IOIP20.sol
|   |   ROO.sol
|   Tokens
|   |   ERC20
|   |   |   ERC20.sol
|   |   |   ERC20Burnable.sol
|   |   |   ERC20Detailed.sol
|   |   |   IERC20.sol
|   |   |   ERC20NonStandardInterface.sol
|   |   |   Maximillion.sol
|   |   |   NativeAddress.sol
|   |   |   RErc20.sol
|   |   |   RErc20Delegate.sol
|   |   |   RErc20Delegator.sol
|   |   |   RErc20Immutable.sol
|   |   |   RNativeToken.sol
|   |   |   RToken.sol
|   |   |   RTokenInterfaces.sol
|   |   |   Reservoir.sol
|   |   ErrorReporter.sol
|   |   Migrations.sol
```

**注：合约中引入的第三方合约函数（如 OIP20.sol 合约中定义的 receiveApproval 函数）不在本审计范围内。**

本次审计的目的是为了审阅Roolend项目基于Solidity语言编写的去中心化借贷应用，发现潜在的安全隐患，研究其设计、架构，并试图找到可能存在的漏洞。

我们全面阅读了Roolend团队提交的上述合约源码，并仔细审阅了上述代码中可能出现问题的方方面面，对上述合约代码给出了全面、综合的改进意见及评审结果。

## 一 免责声明

截至本报告发布之日，本报告所阐述的内容仅反映审计团队对当前智能合约安全进展及状况的理解。任何人在接触或使用与本报告相关的服务、产品、协议、平台、或任何物品时，自行承担一切可能产生的冲突、损失、利益及风险，本报告的审计团队概不负责。

本审计不涉及合约的编译器及任何超出智能合约编程语言的领域。所审计的智能合约由引用链下信息或资源所导致的风险及责任不在本审计覆盖的范围之内。

本审计无法详尽查看每一个细节，也无法穷尽每一种可能，因此本报告的审计团队鼓励本合约的开发团队及任何相关利益方对合约进行任何后续的测试及审计。

对任何第三方使用本报告中所提及或涉及的软件、源码、软件库、产品、服务、信息等一切事物所产生的冲突、损失、利益及风险，本审计团队不保证、不承诺也不承担任何责任。

本报告的内容、获取方式、使用以及任何其所涉及的服务或资源都不能作为任何形式的投资、税务、法律、监管及建议等的依据，也不产生相关的责任。

# 一 审计方式

审计Roolend项目的合约代码是为了能清晰地理解该项目的实现方式及运行原理。审计团队对合约代码进行了深入的研究、分析和测试，并收集了详尽的数据。审计团队会在本报告中会详细列举所发现的每个问题、问题所在的源码位置、问题产生的根源以及对问题的描述，并对问题给出相应的改进建议。

灵踪安全审计的流程如下：

1. 背景研究。灵踪安全团队会阅读项目介绍、白皮书、合约源码等一切Roolend团队所提供的相关材料及信息，以确保灵踪安全团队理解项目合约的规模、范围及功能。
2. 自动化检测。此步骤主要用自动化工具扫描源码，找到常见的潜在漏洞。
3. 人工审阅合约源码。此步骤由工程师逐行阅读代码，找到潜在的漏洞。
4. 逻辑校对。此步骤审计工程师将对代码的理解与Roolend团队提供的材料及信息相比较，检查代码的实现是否符合项目的定义及白皮书等信息中的描述。
5. 测试用例检测。此步骤包括两部分：
  - i. 测试用例设计。审计工程师将根据前述步骤对项目背景的理解及合约代码的理解，针对项目可能的执行逻辑及方式设计测试用例。
  - ii. 测试范围分析。该步骤会详细检查所设计的测试用例是否覆盖了合约代码的所有逻辑分支，并判断测试用例执行后，合约代码的逻辑是否能得到充分的执行及检查
  - iii. 符号执行。该步骤将运行测试用例以测试合约代码所有可能的执行路径。
6. 优化审查。该流程将根据合约的应用场景、调用方式及业界最新的研究成果从可维护性、安全性及可操作性等方面审查合约代码。

# 一 报告结构

本报告列举的每个问题都被设置了一个安全级别，这些安全级别根据其对合约的影响及安全隐患的大小而定。我们对每个问题都给出了相应的改进建议。为了便于读者阅读，我们分别按主题内容和安全级别这两种方式罗列了所有的问题，并提出了全面增强安全性的建议。

# 一 引用文档

在审阅过程中，我们参考了与项目相关的文档以加深对项目逻辑、功能及应用的理解。本次报告参阅的文档资料如下：

<https://roolend.finance/#/markets>

项目白皮书

上述文档被视为本项目代码实现及功能的定义。当我们认为代码实现与文档定义有分歧时，我们及时咨询并与Roolend团队进行了沟通和确认。

# 一 审计结论

经过审计，当前发现的风险数量为：致命风险：0，高危风险：0，中度风险：1，低风险：1。

结论：当前合约代码审计发现风险。

## 02. 灵踪安全介绍

灵踪安全是一家领先的区块链技术公司，公司为行业企业提供安全审计和咨询方面的服务。灵踪安全研发了自己的一系列合约编写和安全审计标准，为众多客户提供了周到、严谨的服务。

## 03. 被审计合约项目介绍

本项目为去中心化借贷应用。

## 04. 合约主要功能

被审计合约的功能包括项目通证的发行、存贷业务、预言机的使用。

### 注意事项：

- 1 本项目 `R1PriceOracle.sol` 使用了外部预言机获取通证的价格，由外部预言机引发的风险不在本次审计范围内。
- 2 项目中的合约 `DexPrice.sol` 为项目方原本用于从 DEX 获取通证价格的方式，后被 `R1PriceOracle.sol` 所取代，`DexPrice.sol` 中获取通证价格的方式应该避免使用。
- 3 合约中的函数 `receiveApproval` 为第三方合约实现函数，此函数的风险不在本审计范围内。

## 05. 本审计的主要工作

在审计过程中，灵踪安全着重配合项目方对代码的实现逻辑、存贷应用及预言机应用可能存在的风险点进行了审计。

## 06. 风险种类

当前审计采用智能工具静态分析和人工审计相结合的方法，从以下多个风险种类方面对合约源码进行了全方位的审计。

- 重入攻击
- 重放攻击
- 重排攻击
- 注入攻击
- 拒绝服务攻击
- 交易顺序依赖
- 条件竞争攻击
- 权限控制攻击
- 整数上溢/下溢攻击
- 时间戳依赖攻击
- Gas 使用, Gas 限制和循环
- 冗余的回调函数
- 函数状态变量的显式可见性
- 逻辑缺陷
- 未声明的存储指针
- 算术精度误差
- tx.origin 身份验证
- 假充值漏洞
- 变量覆盖
- 设计缺陷
- 潜在后门
- 代币发行
- 管理权限
- 代理升级
- 委托调用插槽共享
- 用户资金安全
- 迁移管理

## 07. 风险分级

本报告中的每个问题都被设置了一个安全等级，程度由高到低排列如下：

**致命** 风险及隐患需要立刻解决。

**高危** 风险及隐患将引发风险及问题，必须解决。

**中度** 风险及隐患可能导致潜在风险，最终仍然需要解决。

**低** 风险及隐患主要指各类处理不当或者会引发警告信息的细节，这类问题可以暂时搁置，但建议最终解决。

## 08. 本审计关注的风险重点

根据本合约的功能及应用场景，我们着重审查了下列功能中可能潜藏的风险。

### - 通证发行

我们检查了通证发行是否有不合规的增发接口，以保护投资者的利益和系统的稳定运行。  
经审查此功能暂未发现明显风险。

### - 权限检查

我们检查了每一个能改变合约状态的函数是否具备合适的权限，重点检查那些必须管理员权限才能操作的函数。  
经审查此功能发现风险，细节请参看“11. 问题详述”。

### - 资金安全

我们检查了用户存储在合约中的资产是否存在安全隐患。  
经审查此功能暂未发现明显风险。

### - 价格获取、计算机制

我们检查了项目价格获取和计算的机制，重点关注是否存在价格受到操控的风险。  
经审查此功能暂未发现明显风险。

### - 合约迁移/升级

我们重点检查了合约是否有不安全的升级迁移功能，避免用户资产遭受意料之外的损失。  
经审查此功能发现风险，细节请参看“11. 问题详述”。

## - 其它

经审查其它功能暂未发现明显风险。

# 09. 基于风险等级的问题列表

## A. 致命风险

- 无

## B. 高危风险

- 无

## C. 中度风险

- RErc20Delegator.sol

合约迁移/升级

## D. 低风险

- SimplePriceOracle.sol

测试合约的权限控制

# 10. 基于合约文件的问题列表

- RErc20Delegator.sol

## 合约迁移/升级：中度风险

### - SimplePriceOracle.sol

#### 测试合约的权限控制：低风险

## 11. 问题详述

### - 合约迁移/升级：中度风险

问题位置及描述：

合约文件 `RErc20Delegator.sol` 用到了合约升级，若升级的合约未经审计，则存在潜在隐患。

修改建议：

建议慎用或不用升级功能，实在需要，须对升级合约进行重新审计。

**项目方反馈：**项目方已知悉，在进行必要的升级时会对合约重新进行审计。

### - 测试合约的权限控制：低风险

问题位置及描述：

合约文件 `simplePriceOracle.sol` 中的函数 `setUnderlyingPrice` 和 `setDirectPrice` 无权限控制，可由任何地址设置价格，因此可被用于操控通证价格。不过此函数来源于Compound，仅用于测试网的测试而非在正式网络中使用。

修改建议：

建议加上权限限制，或明确标注此函数仅用于测试网的测试。

**项目方反馈：**项目方已知悉，此合约的函数源自Compound，是用于测试，而非用在正式网络中。

## 12. 增强建议

## - 仅允许流动性好的资产进行抵押

借贷类项目在机制上容易受到这样的攻击：用户可以蓄意挑选流动性不强、质量差的资产进行抵押，通过价格控制抬高该资产的价格，然后进行抵押获取流动性好的优质资产。

建议项目方严格把控可抵押的资产种类，仅允许流动性好的资产进行抵押。