# 智能合约安全审计报告

# ACoconut BTC（acBTC）

**Phase One: AC Genesis Liquidity Mining**

SECBIT

**Oct 7, 2020**

安比（SECBIT）实验室致力于解决区块链全生态的安全问题，提供区块链全生态的安全服务。作为中国信息通信研究院区块链安全技术组成员，参与编写区块链安全白皮书和参与制定区块链安全审计规范。

安比（SECBIT）实验室智能合约审计从合约的技术实现、业务逻辑、接口规范、Gas 优化、发行风险等维度，由两组专业审计人员分别独立进行安全审计，审计过程借助于安比实验室研发的一系列形式化验证、语义分析等工具进行扫描检测，力求尽可能全面的解决所有安全问题。

# 1. 综述

ACoconut BTC（acBTC）是基于以太坊的聚合 BTC ERC20 合约。acBTC 协议支持原生 BTC 以及 BTC ERC20 代币的交换、贷款和其他产生收益的应用，并集成到一个高度安全，高效和可用的标准中。安比（SECBIT）实验室于 2020 年 9 月 24 日至 10 月 7 日对 acBTC Phase One 合约进行安全审计，审计过程从**代码漏洞**，**逻辑漏洞**和**发行风险评估**三个维度对代码进行分析。审计结果表明，acBTC Phase One 并未包含致命的安全漏洞，安比（SECBIT）实验室给出了如下几点逻辑实现存疑和代码优化建议项（详见第4章节）。

| 风险类型 | 描述 | 风险级别 | 状态 |
| --- | --- | --- | --- |
| 代码实现 | 4.3.1 ACoconut/ACoconutBTC 合约中，`setMinter()` 敏感操作未添加事件。 | 提示 | 已修复 |
| 代码实现 | 4.3.2 ACoconutBTC 合约中，`burn()`函数实现存疑。 | 低 | 已修复 |
| 代码规范 | 4.3.3 StrategyACoconutBTC 合约中，reserveRecipient 变量不可见。 | 提示 | 已修复 |
| 代码实现 | 4.3.4 StrategyACoconutBTC 合约中，`withdraw()`与`withdrawAll()`函数权限校验存疑。 | 提示 | 已讨论 |
| 代码实现 | 4.3.5 StrategyCurveRenBTC::harvest() 没有权限控制，可能存在套利机会。 | 中 | 已修复 |
| 代码优化 | 4.3.6 ACoconut 合约仅需继承 ERC20Capped 合约，_beforeTokenTransfer() 函数无需重载。 | 提示 | 已修复 |

# 2. 项目信息

该部分描述了项目的基本信息和代码组成。

## 2.1 基本信息

以下展示了 acBTC 的基本信息：

- **项目网站**
  - https://acbtc.fi
- **项目设计文档**
  - https://docs.acbtc.fi/
- **合约代码**
  - https://github.com/nutsfinance/acBTC, commit c04cb435b70f4194586ba20666abfc57fd020ce4

## 2.2 合约列表

以下展示了 acBTC 项目包含的主要合约列表：

| 合约名称 | 描述 |
| --- | --- |
| Account | 交易账户合约 |
| AccountFactory | 交易账户生成合约 |
| ACoconut | AC Token合约 |
| ACoconutBTC | acBTC Token合约 |
| ACoconutSwap | acSwap功能实现 |
| ACoconutSwapProxy | acSwap部署使用的proxy合约 |
| ACoconutVault | acVault功能实现 |
| CurveRenCrvMigrator | renCrv转移合约 |
| StrategyACoconutBTC | acBTC收益策略合约 |
| StakingApplication | 权益应用功能实现 |
| AdminUpgradeabilityProxy | 授权功能的proxy升级合约 |
| BaseUpgradeabilityProxy | proxy升级合约 |
| Initializable | 初始化控制合约 |
| Proxy | Proxy合约 |
| Controller | 资金管理合约 |
| RewardedVault | 有奖励的资金收益管理合约 |
| StrategyCurveRenBTC | renCrv收益策略合约 |
| Vault | 资金收益管理合约 |

注：ACoconutSwap 和 CurveRenCrvMigrator 尚在开发和测试中，不包含在 acBTC Phase One 的审计范围内。

# 3. 代码分析

该部分描述了项目代码的详细内容分析，从「角色分类」和「功能分析」这两部分来进行说明。

## 3.1 角色分类

acBTC Phase One 中主要涉及以下几种关键角色，分别是 Governance Account（治理账户）、Minter（铸币者）、Wallet Account（钱包账户）、Common Account（普通账户）。

- 治理账户 (Governance Account)
  - 描述

    合约的治理者
  - 功能权限
    - 设定 AC Token 或 acBTC 的铸币者
    - 设定合约的基础参数
  - 授权方式

    合约的创建者，或者由治理账户转让授权
- 铸币者 (Minter)
  - 描述

    对 AC Token 或 acBTC Token 进行铸币操作
  - 功能权限

    铸币
  - 授权方式

    由治理账户授权
- 钱包账户 (Wallet Account)
  - 描述

    用户与 acBTC 合约交互使用的账户
  - 功能权限
    - 分级权限管理（owner, admin, operator）
    - owner 账户授权或取消 admin 账户权限
    - admin 账户授权或取消 operator 账户权限
    - operator 账户对钱包账户中的 ETH 和 ERC20 Token 进行转账操作
    - operator 账户对钱包账户中的 ERC20 Token 进行授权操作
    - operator 账户在owner授权后对其账户 ERC20 Token 进行转账操作
    - operator 账户可发起远程调用

    - 授权方式
        - owner 为合约的创建者，或者由 owner 转让授权
        - admin 由 owner 授权
        - operator 由 admin 授权
- 普通账户（Common Account）
    - 描述

      持有 AC Token 或 acBTC Token 的账户
    - 功能权限
        - 对账户上的 AC Token 或 acBTC Token 进行转账
        - 授权其他账户转账
        - AC Token 持有者参与投票治理
    - 授权方式

      AC Token 或 acBTC Token 的持有者

### 3.2 功能分析

acBTC 是基于以太坊的聚合 ERC20 BTC Token 合约，合约实现关键功能分为以下几项：

- acBTC

  acBTC 是 聚合 BTC ERC20 Token 的合约，基于 Curve 的 StableSwap 算法集成了 BTC ERC20 Token 存储与交换应用。
- acVault

  acVault 是 renCrv 的资金库，抵押 renCrv 的用户可在 renCrv 迁移前，从合约中获得收益。renCrv 迁移后成为 acBTC 资金库，用户可在 acBTC 中获得收益。
- acSwap

  acSwap 是 ERC20 BTC Token 的去中心化交易所，它管理聚合的 ERC20 BTC Token，包括 WBTC 和 renBTC，并从中引导 acBTC 的价值。

注：acSwap 功能尚在开发和测试中，不包含在 acBTC Phase One 的审计范围内。

## 4. 审计详情

该部分描述合约审计流程和详细结果，并对发现的问题（代码漏洞，代码规范和逻辑漏洞），合约发行的风险点和附加提示项进行详细的说明。

### 4.1 审计过程

本次审计工作，严格按照安比（SECBIT）实验室审计流程规范执行，从代码漏洞，逻辑问题以及合约发行风险三个维度进行全面分析。审计流程大致分为四个步骤:

- 各审计小组对代码进行逐行分析，根据审计内容要求进行审计
- 各审计小组对漏洞和风险进行评估
- 审计小组之间交换审计结果，并对审计结果进行逐一审查和确认
- 审计小组配合审计负责人生成审计报告

### 4.2 审计结果

本次审计首先经过安比（SECBIT）实验室推出的分析工具 adelaide、sf-checker 和 badmsg.sender（内部版本）扫描，再利用开源安全分析工具 Mythril、Slither、SmartCheck 以及 Securify 检查，检查结果由审计小组成员详细确认。审计小组成员对合约源码和电路代码进行逐行检查、评估，汇总审计结果。审计内容总结为如下 21 大项。

| 编号 | 分类 | 结果 |
| --- | --- | --- |
| 1 | 合约各功能能够正常执行 | 通过 |
| 2 | 合约代码不存在明显的漏洞（如整数溢出） | 通过 |
| 3 | 能够通过编译器的编译并且编译器没有任何警告输出 | 通过 |
| 4 | 合约代码能够通过常见检测工具检测，并无明显漏洞 | 通过 |
| 5 | 不存在明显的 Gas 损耗 | 通过 |
| 6 | 符合 EIP20 标准规范 | 通过 |
| 7 | 底层调用（call, delegatecall, callcode）或内联汇编的操作不存在安全隐患 | 通过 |
| 8 | 代码中不包含已过期或被废弃的用法 | 通过 |
| 9 | 代码实现清晰明确，函数可见性定义明确，变量数据类型定义明确，合约版本号明确 | 通过 |
| 10 | 不存在冗余代码 | 通过 |
| 11 | 不存在受时间和外部网络环境影响的隐患 | 通过 |
| 12 | 业务逻辑实现清晰明确 | 通过 |

| | | |
|---|---|---|
| 13 | 代码实现逻辑与注释，项目白皮书等资料保持一致 | 通过 |
| 14 | 代码不存在设计意图中未提及的逻辑 | 通过 |
| 15 | 业务逻辑实现不存在疑义 | 通过 |
| 16 | 不存在危及项目方利益的明确风险 | 通过 |
| 17 | 不存在危及相关机构如交易所，钱包，DAPP 方利益的明确风险 | 通过 |
| 18 | 不存在危及普通持币用户利益的明确风险 | 通过 |
| 19 | 不存在修改他人账户余额的特权 | 通过 |
| 20 | 不存在非必要的铸币权限 | 通过 |
| 21 | 多管理角色下，管理权限划分明确，各权限优先级划分明确 | 通过 |

## 4.3 问题列表

### 4.3.1 ACoconut/ACoconutBTC 合约中，`setMinter()` 敏感操作未添加事件。

| 风险类型 | 风险级别 | 影响点 | 状态 |
|---|---|---|---|
| 代码实现 | 提示 | 操作可审计性 | 已修复 |

**问题描述**

ACoconut/ACoconutBTC 合约中， `setMinter()` 敏感操作未添加事件，普通用户无法方便查看合约的 minter 情况。

```
// ACoconut.sol/ACoconutBTC.sol
function setMinter(address _user, bool _allowed) public {
    require(msg.sender == governance, "not governance");
    minters[_user] = _allowed;
}
```

**修改建议**

建议在 ACoconut/ACoconutBTC 合约中为 `setMinter()` 敏感操作添加事件，方便社区审计 minter 权限。

**状态**

已在 6bf5964 中按照修改建议修复。

### 4.3.2 ACoconutBTC 合约中，`burn()`函数实现存疑。

| 风险类型 | 风险级别 | 影响点 | 状态 |
| --- | --- | --- | --- |
| 代码实现 | 低 | 权限管理 | 已修复 |

**问题描述**

ACoconutBTC 合约的`burn()`函数中， minter 账户可以执行任意的 burn 操作，此处未说明原因。

```
// ACoconutBTC.sol
function burn(address _user, uint256 _amount) public {
  require(minters[msg.sender], "not minter");
  _burn(_user, _amount);
}
```

**修改建议**

建议添加说明，是否为业务需要。

**状态**

已在 6bf5964 中按照修改建议添加说明。

### 4.3.3 StrategyACoconutBTC 合约中， reserveRecipient 变量不可见。

| 风险类型 | 风险级别 | 影响点 | 状态 |
| --- | --- | --- | --- |
| 代码规范 | 提示 | 变量可见性 | 已修复 |

**问题描述**

StrategyACoconutBTC 合约中，`reserveRecipient` 是 private 变量，有 setter 函数但是没有 getter 函数。因为 `reserveRecipient` 涉及转账操作，没有 getter 函数易造成误解。

**修改建议**

建议在合约中添加 getter 函数。

**状态**

已在 6bf5964 中按照修改建议修复。

### 4.3.4 StrategyACoconutBTC 合约中，`withdraw()` 与 `withdrawAll()` 函数权限校验存疑。

| 风险类型 | 风险级别 | 影响点 | 状态 |
| --- | --- | --- | --- |
| 代码实现 | 提示 | 权限管理 | 已讨论 |

**问题描述**

StrategyACoconutBTC 合约中，`withdraw()` 与 `withdrawAll()` 函数权限与 StrategyCurveRenBTC 合约中的同名函数不一致。

**修改建议**

建议明确合约中 `withdraw()` 与 `withdrawAll()` 函数的权限设计。

**状态**

已与开发者讨论，设计如此，合约维持现状。

### 4.3.5 StrategyCurveRenBTC::harvest() 没有权限控制，可能存在套利机会。

| 风险类型 | 风险级别 | 影响点 | 状态 |
| --- | --- | --- | --- |
| 代码实现 | 中 | 权限管理 | 已修复 |

**问题描述**

StrategyCurveRenBTC 合约中的 harvest() 函数没有权限控制，可能被操纵接口到其他 DEX 做交易，从而对他人形成套利机会，对收益造成影响。

**修改建议**

建议对 `harvest()` 函数添加权限控制。

**状态**

已在 c959dc2 按照修改建议修复。

### 4.3.6 ACoconut 合约仅需继承 ERC20Capped 合约，_beforeTokenTransfer() 函数无需重载。

| 风险类型 | 风险级别 | 影响点 | 状态 |
| --- | --- | --- | --- |
| 代码优化 | 提示 | 代码冗余 | 已修复 |

### 问题描述

ACoconut 合约代码实现中继承了 ERC20, ERC20Capped 两个合约，然而两个合约本身已有继承关系。因此 ACoconut 合约实现时仅需继承 ERC20Capped 合约。

```
// ACoconut.sol
contract ACoconut is ERC20, ERC20Capped {

  ...

  function _beforeTokenTransfer(address from, address to, uint256
amount) internal override(ERC20, ERC20Capped) {
      ERC20Capped._beforeTokenTransfer(from, to, amount);
  }

  ...
}
```

### 修改建议

建议在合约实现中仅继承 ERC20Capped 合约。

### 状态

已在 a0dec81 中按照修改建议修复。

# 5. 结论

安比（SECBIT）实验室在对 ACoconut BTC（acBTC）合约进行分析后，发现部分可优化项，并提出了对应的修复及优化建议，上文均已给出具体的分析说明。对于本报告中提出的问题，acBTC 开发者均已在最新版代码中进行了修复。安比（SECBIT）实验室认为 acBTC 项目代码质量较高、文档详细、测试用例完整。acBTC Phase One 完整实现了 AC Token 的初始分发，尤其是 ACoconutVault 能够同时进行 CRV 和 AC Token 挖矿，并为未来迁移至 acBTC 提供了预留接口。此外，ACoconutSwap 和 CurveRenCrvMigrator 正在进行最后的测试和验证，在上线前即将作为 acBTC Phase Two 的重点审计对象。

# 免责声明

SECBIT 智能合约安全审计从合约代码质量、合约逻辑设计和合约发行风险等方面对合约的正确性、安全性、可执行性进行审计，但不做任何和代码的适用性、商业模式和管理制度的适用性及其他与合约适用性相关的承诺。本报告为技术信息文件，不作为投资指导，也不为代币交易背书。

# 附录

## 漏洞风险级别介绍

| 风险级别 | 风险描述 |
| --- | --- |
| 高 | 可以严重损害合约完整性的缺陷，能够允许攻击者盗取以太币及Token，或者把以太币锁死在合约里等缺陷。 |
| 中 | 在一定限制条件下能够损害合约安全的缺陷，造成某些参与方利益损失的缺陷。 |
| 低 | 并未对合约安全造成实质损害的缺陷。 |
| 提示 | 不会带来直接的风险，但与合约安全实践或合约合理性建议有关的信息。 |

**安比（SECBIT）实验室致力于参与共建共识、可信、有序的区块链经济体。**


https://secbit.io

info@secbit.io

@secbit_io