智能合约安全审计报告

Hydro MultiSigWallet

SECBIT

Dec 17, 2018

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，力求尽可能全面地解决所有安全问题。

1. 综述

Hydro MultiSigWallet 是用于部署在以太坊上的多重签名钱包合约。安比（SECBIT）实验室于 2018 年 11 月 16 日至 2018 年 11 月 19 日对合约进行审计。审计过程从**实现漏洞**、**逻辑漏洞**和**风险评估**三个维度对合约进行分析。审计结果表明，MultiSigWallet 合约并未包含致命的安全漏洞，安比（SECBIT）实验室给出了如下代码修复或优化建议项（详见第4章节）。Hydro MultiSigWallet 开发团队对代码进行了优化更新。安比（SECBIT）实验室于 2018 年 12 月 11 日至 2018 年 12 月 17 日进行复审。经审计，更新后的代码解决了此前发现的所有问题，并未引入新的安全风险，达到了安全发布标准。

风险类型	描述	风险级别	最终状态
代码优化	`isConfirmed` 函数中，交易执行条件 `count == required` 的判断，在特殊情况下可能造成交易无法执行	低	已按照建议修改
实现漏洞	`getTransactionIds` 函数中，未对 `from` 和 `to` 变量进行检查，可能下溢或者超出 `_transactionIds` 数组范围造成返回结果异常	中	已删除对应函数
代码优化	`executeTransaction` 函数中，没有使用 `ownerExists` 修饰符，与合约逻辑不符	低	已按照建议修改

2. 合约信息

该部分描述了合约的基本信息和代码组成。

2.1 基本信息

以下列表展示了 MultiSigWallet 的基本信息：

名称	MultiSigWallet
行数	394, 64
文件名称	MultiSigWallet.sol, MultiSigWalletWithLock.sol

文件来源	Hydro
Git Commit（初审）	ab0d97dd01683cfadc1e02681c08e6946ef838ac
Git Commit（复审）	fabed26fdfe250bcb2a4cc0af1cb6b4bfb0098f0
合约阶段	开发阶段

2.2 合约列表

以下展示了 MultiSigWallet 项目包含的合约列表:

合约名称	行数	描述
MultiSigWallet	394	Multisignature wallet - Allows multiple parties to agree on transactions before execution.
MultiSigWalletWithLock	64	Multisignature wallet with lock function.

3. 合约分析

该部分描述了合约代码的详细分析内容，从合约类型，合约账户角色权限和合约实现功能三部分来进行说明。

3.1 合约类型

钱包类合约。

3.2 合约账户分类

MultiSigWallet 中有两种角色账户，即 owner 账户与普通用户。

普通用户

- 描述

  钱包观察者
- 功能权限
  - 在合约中存入 ETH
  - 查看钱包管理者，钱包中的交易内容与交易状态信息
- 授权方式

  任意账户

owner

- 描述

  钱包管理者
- 功能权限
  - 普通用户的所有权限
  - 提交 Transaction 提案
  - 参与 Transaction 投票，执行 Transaction
  - 转移 owner 权限
- 授权方式

  合约创建时授权，通过 Transaction 提案授权，通过 owner 权限转移授权。

3.3 功能分析

MultiSigWallet 的主要功能是实现多人管理的多重签名钱包。

阶段1：合约创建与前期准备

- 合约创建时指定合约管理者 owner 名单，钱包执行交易时所需管理者确认数。
- 合约管理者在钱包合约中存入共同管理的资金。

阶段2：提交交易提案

- 任意一个 owner 可以提交交易的提案，同时做出确认，等待其他的 owner 确认这个提案。

阶段3：提案执行

- 对于交易提案，每个 owner 都可以对其进行确认，当确认数达到合约规定值时，交易进入待执行状态，并加上时间锁。到达时间锁规定时间后，交易可以执行。

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（实现漏洞，代码优化和逻辑漏洞），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:

- 各审计小组对合约代码进行逐行分析，根据合约审计内容要求进行审计
- 各审计小组对合约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进行逐行检查、评估，汇总审计结果。审计内容总结为如下 20 大项。

编号	分类	结果
1	合约各功能可以正常执行	通过
2	合约代码不存在明显的漏洞（如整数溢出等）	通过
3	能够通过编译并且没有任何警告输出	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	符合安全开发规范	通过
6	底层调用（call, delegatecall, callcode）或内联汇编的操作不存在安全隐患	通过
7	代码中不包含已过期或被废弃的用法	通过
8	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
9	不存在冗余代码	通过
10	不存在受时间和外部网络环境影响的隐患	通过
11	调用外部合约符合规范，如 Token 合约	通过
12	合约不存在明显的 Gas 损耗	通过
13	业务逻辑实现清晰明确	通过
14	代码实现逻辑与注释、项目白皮书等资料保持一致	通过
15	代码不存在设计意图中未提及的逻辑	通过
16	业务逻辑实现不存在疑义	通过
17	机制设计不存在明显的缺陷	通过
18	博弈论角度评估业务逻辑不存在公平性问题	通过
19	不存在危及相关项目方利益的明确风险	通过

4.3 问题列表

1. `isConfirmed` 函数中，交易执行条件 `count == required` 在特殊情况下可能造成交易无法执行。
    - 风险级别：**低**
    - 问题类型： 代码优化
    - 问题描述：

      合约中每当管理者执行 `confirmTransaction` 函数，立即调用 `executeTransaction` 执行交易，当管理者对交易确认数达到 `required` 时，通过 call 调用执行交易。在实际执行中，由于合约内金额不足等原因，存在 call 调用失败，可能存在交易确认数大于 `required` 的情况，此时需要管理者额外执行 `revokeConfirmation` 函数，才能正常执行该交易。
    - 影响结果：

      可能造成钱包内的交易无法正常执行。
    - 修改方案：

      建议将 `isConfirmed` 函数中交易执行条件改为 `count >= required`。
    - 问题状态：

      已按照建议修改。

2. `getTransactionIds` 函数中，未对 `from` 和 `to` 变量进行检查，可能产生下溢或者超出 `_transactionIds` 数组范围造成返回结果异常。

- 风险级别：**中**
- 问题类型：实现漏洞
- 问题描述：

  函数中未对 from 和 to 变量进行检查，存在整数下溢的问题。

```
_transactionIds = new uint256[](to - from);
for (i = from; i < to; i++) {
    _transactionIds[i - from] = transactionIdsTemp[i];
}
```

- 影响结果： 无法正常查询到结果。
- 解决方案： 建议增加校验，判断 from 和 to 是否在交易个数的范围内，以及 to 是否大于 from。

```
  require(from<to && to<=transactionIdsTemp.length, "variable out
of range.");
```

- 问题状态:

  已删除 `getTransactionIds` 函数。

3. `executeTransaction` 函数可以由任意地址进行调用，与合约逻辑不符。
   - 风险级别：**低**
   - 问题类型：代码优化
   - 问题描述:

     `executeTransaction` 函数用于执行钱包合约中的交易提案，可以由任意用户进行调用，与合约逻辑不符。
   - 影响结果

     符合交易条件的未执行交易提案可能通过任意用户执行。
   - 规避方案

     建议在 `executeTransaction` 函数中使用 `ownerExists` 修饰符，使代码逻辑更加清晰。
   - 问题状态:

     已按照建议修改。

4.4 风险提示

安比（SECBIT）实验室在对 MultiSigWallet 合约风险进行评估以后，指出合约存在如下风险项:

1. `executeTransaction` 函数中，call 调用任意函数，可能引入风险。
   - 风险级别：**中**
   - 问题类型：实现漏洞
   - 问题描述:

   在 `executeTransaction` 函数中，合约执行了 call 调用，钱包管理者需要对 call 调用的内容进行检查:

```
  function executeTransaction(uint256 transactionId)
      public
      notExecuted(transactionId)
  {
```

```
        if (isConfirmed(transactionId)) {
            Transaction storage transaction =
transactions[transactionId];
            transaction.executed = true;
            if
(transaction.destination.call.value(transaction.value)
(transaction.data))
                emit Execution(transactionId);
            else {
                emit ExecutionFailure(transactionId);
                transaction.executed = false;
            }
        }
    }
```

5. 结论

安比（SECBIT）实验室在对 Hydro MultiSigWallet 合约进行分析后，发现了部分代码缺陷和漏洞，并提出了对应的修复及优化建议。MultiSigWallet 实现了多重签名钱包的功能。在代码功能上，存在 2 处优化建议，1 处实现漏洞；存在 1 项风险点，上文均已给出具体的分析说明。同时，安比（SECBIT）实验室认为，MultiSigWallet 合约代码质量较高。

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，也不为钱包交易背书。

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
audit@secbit.io
@secbit_io