iSwap V4 审 计 报 告

Version 1.0.0

报告编号：2021120800011018

灵踪安全发布

2021年12月8日

灵踪安全

FAIRYPROOF

01. 介绍

本报告包含了灵踪安全在iSwap团队要求下对iSwap V4项目代码进行审计的结果。

审计开始时间：

2021年12月1日

审计结束时间：

2021年12月6日

审计代码文件及目录结构：

审计结束时的代码文件名及对应的SHA-256哈希值为：

```
ISwapArbitrumBridge.sol:
0x65d8c793bc1f8df6d8e728528dc4bffdcd727db5ceb1d850b70ba82206db9979

ISwapAvaxBridge.sol:
0x6f429a15ae0355e01cd0af8fef77b26db91028200c28defeb66f23ab053c875b

ISwapBaseBridge.sol:
0x0cb3028d055333c283a76415704c877562d4281bffb63c2d4660f0129919c233

ISwapBscBridge.sol:
0xd1ab95fcea393278e2adc9983bdfdbf8e6755b38245308bb689ef71e89888db6

ISwapEthBridge.sol:
0x3893a53ed7db19e2c6eeedee028c4d2ecf9ae1104911074fff8bd35469b59d03

ISwapFtmBridge.sol:
0x42bc9b01630c77649b26806827995f33a985c310b78e3d35748ffaf3d16c6461

ISwapHecoBridge.sol:
0x929f3032659481ccb2c6c119ace00fbbcbe592269059c00ad73ba0e16974c9f2

ISwapOecBridge.sol:
0x91f4a19b1dd9410547b37d4282c701686da76a46b8c437066586d1de6cf89413

ISwapPolygonBridge.sol:
0x932af9b9a35dd9aa785483c46fc3421103dee97dba2041dba7643d0a098606b0

IHRC20.sol:
0x8f909a8cc5f325334aa4c767554ce7cf2d8faa7ce6cf37dc8461e959c1d0f8f0

IRouter.sol:
0x8d3d1f32a4632fe68614cc775e5c1fde5147b08abdc2647ff4df4ac07115170f

IWETH.sol:
```

```
0xf79a254f47708877e0ff8f94fcf6a9fd6668c58e89a0e89b8537af9b3905f09f

ParamsParser.sol:
0x12db2fc15f3ea649eb490f8e613e90f81e69bb2ff68494183d92cbdcc38d365c

RevertReasonParser.sol:
0xf4cdf713b9f1ae39a549d414e6e6c86b72a148da6d01df793ca02e8d820200e1

TransferHelper.sol:
0xb9774be2bc8df05e45e941137ad5aeff1485e8d27be0023aaa2894f52fc31a12

IHub.sol:
0x54217004bc0486a44702687489456a1aae34a2914e4bd5c29afe2ea3c878f978

IHRC20.sol:
0x8f909a8cc5f325334aa4c767554ce7cf2d8faa7ce6cf37dc8461e959c1d0f8f0

IWETH.sol:
0xd8c65eebf30834fd5f2e50e3f85e0ddec8f3f59f1016862d02b1820917547b41

readme.md:
0x0e4d51e0078d0384c792fdc322ab32c70ac9a18a8f2b3d57504443309532a54e

TransferHelper.sol:
0xb9774be2bc8df05e45e941137ad5aeff1485e8d27be0023aaa2894f52fc31a12
```

审计代码的文件名及目录结构为：

```
iswap-v4/
├── ISwapArbitrumBridge.sol
├── ISwapAvaxBridge.sol
├── ISwapBaseBridge.sol
├── ISwapBscBridge.sol
├── ISwapEthBridge.sol
├── ISwapFtmBridge.sol
├── ISwapHecoBridge.sol
├── ISwapOecBridge.sol
├── ISwapPolygonBridge.sol
├── interfaces
|   ├── IHRC20.sol
|   ├── IRouter.sol
|   └── IWETH.sol
└── utils
    ├── ParamsParser.sol
    ├── RevertReasonParser.sol
    └── TransferHelper.sol

iswap-IHub/
├── IHub.sol
├── interfaces
|   ├── IHRC20.sol
|   └── IWETH.sol
├── readme.md
└── utils
    └── TransferHelper.sol
```

本次审计的目的是为了审阅iSwap项目团队基于Solidity语言编写的跨链资产交易平台应用，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iSwap团队提交的上述代码，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。

本次审计仅针对授权方指定版本的代码、安装包及其它授权方提供的素材展开，其结论仅对相应版本的应用适用，一旦相关代码、配置、运营环境等发生变化，相应结论将不再适用。

— 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所审计的代码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。

本审计不涉及审计代码的编译器及任何超出代码编程语言的领域。如果所审计的代码为智能合约，则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项目的开发团队及任何相关利益方对所审计代码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。

— 审计方式

审计iSwap V4项目的源代码是为了能清晰地理解该项目的实现方式及运行原理。审计团队对项目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iSwap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. 人工审阅项目代码。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iSwap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项目代码。

— 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项目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。

— 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：

http://iswap.com

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iSwap团队进行了沟通和确认。

— 审计结果

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：0，低风险：1。

02. Fairyproof介绍

Fairyproof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. 项目介绍

iSwap V4是一个跨链资产交易平台。

注：目前支持ETH、BSC、HECO、OKEX、Polygon、Arbitrum、Avalanche和Fantom网络。

04. 审计代码的主要功能

所审计代码实现的主要功能有：

以稳定币作为中介，实现ETH、BSC、HECO、OKEX、Polygon、Arbitrum、Avalanche和Fantom之间的资产跨链交易。

注：

跨链交易的验证（relayer节点）采用中心化的方式实现，不在本次审计范围之内

项目方有权限存入或提取链上资产，所以跨链交易前，请先确认目标链的对应代币余额是否充足

跨链交易时，用户交易会被收取一定的手续费，白名单用户交易可能获得奖励

05. 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项目代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
- 通证发行
- 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

06. 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。

07. 本审计关注的风险重点

根据所审计代码的功能及应用场景，我们着重审查了下列功能中可能潜藏的风险。

- 数值安全

我们检查了合约中的数值处理是否有算术溢出问题。常规的加减运算容易引起整数溢出，尤其是在处理通证金额或计算奖励金额时，本合约均使用了严谨的数学模块进行了处理。

经审查此功能暂未发现明显风险。

- 手续费率设定

我们检查了用户发送通证收取的手续费是否在可控的安全范围。

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。

经审查此功能暂未发现明显风险。

- 不安全的状态更改

我们检查了合约创建时初始化的一些关键状态变量。在很多情况下，这些变量只应该初始化一次，运行后再更改可能会给整个合约运行带来意料不到的风险。

经审查此功能暂未发现明显风险。

- 资金安全与后门

我们重点检查了入金和出金函数是否存在用户资金不受控制的风险、是否可能导致用户资金受损的问题。

在原`IHub.sol`第114行，`order.from`可由用户随意指定，在`order.from`地址`allowance`足够的情况下，可能被恶意转账到他人地址。其代码片段如下：

```
TransferHelper.safeTransferFrom(order.asset, order.from, address(this),
order.amount);
```

建议将`order.from`改为`msg.sender`，以确保发起交易者只能操作自己的资金：

```
TransferHelper.safeTransferFrom(order.asset, msg.sender, address(this),
order.amount);
```

项目方已在最新提交的代码中修改此问题。

再次审查，此功能暂未发现明显风险。

- 合约迁移与升级

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。

经审查此功能暂未发现明显风险。

- 跨链交易逻辑

我们检查了跨链交易的流程及relayer的交易确认逻辑是否存在风险

经审查此功能发现风险，细节请参看“09. 问题详述”。

- 其它

经审查其它功能暂未发现明显风险。

08. 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

- 无

D. 低风险

- 交易确认逻辑导致的抢先交易

09. 问题详述

- 交易确认逻辑导致的抢先交易：低风险

问题位置及描述：

现有的Relayer交易逻辑存在一定风险。当用户在源链发起的交易时，套利者可利用交易逻辑的缺陷在目标链发起抢先交易实现无风险套利，从而造成用户非正常的滑点损失。

修改建议：

建议在relayer中以随机、乱序或其它更好的策略来执行确认逻辑，并加以交易金额的限制来排除大部分的套利风险。

项目方反馈：将在未来的升级版本中添加相关策略以排除大部分套利风险。

10. 增强建议

- 将管理员权限交给多签管理

bridge和relayer的管理员权限建议使用多签，以降低私钥泄露或被盗后的风险。

- 采取多种措施监听交易

采用多节点侦听用户交易，并时刻检查链上未完成交易，确保用户交易不漏掉，并得到及时处理