智能合约安全审计报告

1. 概要

慢雾安全团队于 2021 年 01 月 19 日，收到 Huobi ECO Chain 团队对 HugSwap 系统安全审计的申请，根据项目特点慢雾安全团队制定如下审计方案。

慢雾安全团队将采用“白盒为主，黑灰为辅”的策略，以最贴近真实攻击的方式，对项目进行安全审计。

慢雾科技 DeFi 项目测试方法：

黑盒测试	站在外部从攻击者角度进行安全测试。
灰盒测试	通过脚本工具对代码模块进行安全测试，观察内部运行状态，挖掘弱点。
白盒测试	基于项目的源代码，进行脆弱性分析和漏洞挖掘。

慢雾科技 DeFi 漏洞风险等级：

严重漏洞	严重漏洞会对项目的安全造成重大影响，强烈建议修复严重漏洞。
高危漏洞	高危漏洞会影响项目的正常运行，强烈建议修复高危漏洞。
中危漏洞	中危漏洞会影响项目的运行，建议修复中危漏洞。
低危漏洞	低危漏洞可能在特定场景中会影响项目的业务操作，建议项目方自行评估和考虑这些问题是否需要修复。
弱点	理论上存在安全隐患，但工程上极难复现。
增强建议	编码或架构存在更好的实践方法。

2. 审计方法

慢雾安全团队智能合约安全审计流程包含两个步骤:

- 使用开源或内部自动化分析的工具对合约代码中常见的安全漏洞进行扫描和测试。
- 人工审计代码的安全问题，通过人工分析合约代码，发现代码中潜在的安全问题。

如下是合约代码审计过程中我们会重点审查的漏洞列表:

（其他未知安全漏洞不包含在本次审计责任范围）

- 重入攻击
- 重放攻击
- 重排攻击
- 短地址攻击
- 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
- 不安全的接口使用
- 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

3. 项目背景

3.1 项目介绍

Hugswap 是在火币生态链搭建的一个基于资金池理念的自动化做市商（AMM）去中心化交易所。用户可在 Hugswap 上自由交易 Heco 资产，交易安全、手续费低。同时用户可通过提供流动性获取交易手续费分成。其目标是让 Heco 的资产兑换更安全、更快速、更便捷。

Hugswap 优先支持 Heco 原生资产和 H 系资产，致力于让交易更简单。

审计合约文件：

项目源代码

审计初始版本：

文件名:

hugswap-contracts-master.zip:

SHA256:

bc681584f8b4b63c153ef209d22b1c2b29936ef35aa9b60520ba5cbecc17ba8a

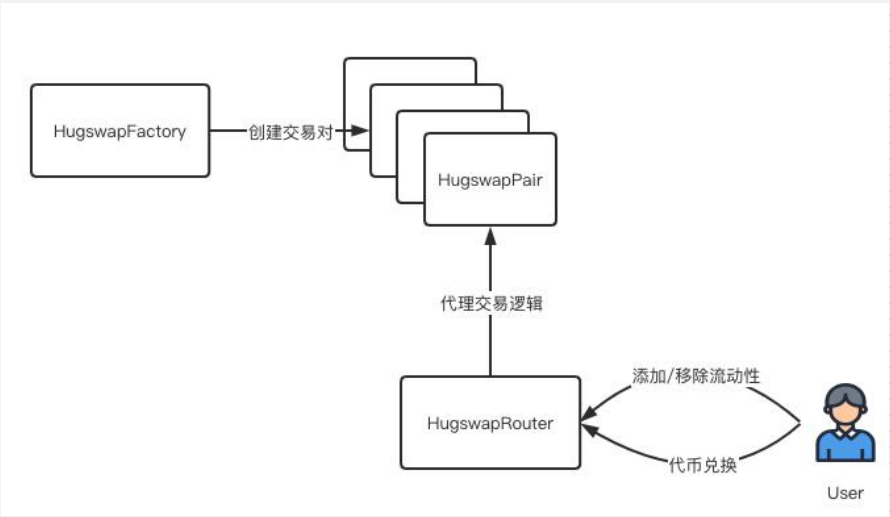
3.2 项目结构

```
contracts
├── core
│   ├── HugswapERC20.sol
│   ├── HugswapFactory.sol
│   ├── HugswapPair.sol
│   ├── HugswapRouter.sol
│   └── Multicall.sol
├── interfaces
│   ├── IERC20.sol
│   ├── IHugswapCallee.sol
```

```
|   ├── IHugswapERC20.sol
|   ├── IHugswapFactory.sol
|   ├── IHugswapPair.sol
|   ├── IHugswapRouter.sol
|   └── IWHT.sol
└── libraries
    ├── HugswapLibrary.sol
    ├── Math.sol
    ├── SafeMath.sol
    ├── TransferHelper.sol
    └── UQ112x112.sol
```

3.3 项目架构

Hugswap 系统整体分成 3 个部分，分别是 HugswapPair、HugswapFactory 及 HugswapRouter。其中 HugswapFactory 负责以 HugswapPair 合约为模版创建交易对，HugswapRouter 合约负责代理用户的交易数据到 HugswapPair 中，方便用户操作。总体架构图如下：



4. 代码概述

4.1 主要合约函数可见性分析

在审计过程中，慢雾安全团队对核心合约的可见性进行分析，结果如下:

HugswapFactory			
Function Name	**Visibility**	**Mutability**	**Modifiers**
constructor	Public	can modify state	-
allPairsLength	External	-	-
getInitCode	External	-	-
createPair	External	can modify state	-
setFeeTo	External	can modify state	-
setFeeToSetter	External	can modify state	-

HugswapERC20			
Function Name	**Visibility**	**Mutability**	**Modifiers**
constructor	Public	can modify state	-
_mint	Internal	can modify state	-
_burn	Internal	can modify state	-

_approve	Private	can modify state	-
_transfer	Private	can modify state	-
approve	External	can modify state	-
transfer	External	can modify state	-
transferFrom	External	can modify state	-
permit	External	can modify state	-

HugswapPair			
Function Name	**Visibility**	**Mutability**	**Modifiers**
getReserves	Public	-	-
_safeTransfer	Private	can modify state	-
constructor	Public	can modify state	-
initialize	External	can modify state	-
_update	Private	can modify state	-
_mintFee	Private	can modify state	-
mint	External	can modify state	lock
burn	External	can modify state	lock
swap	External	can modify state	lock
skim	External	can modify state	lock
sync	External	can modify state	lock

HugswapPair			
Function Name	**Visibility**	**Mutability**	**Modifiers**
constructor	Public	can modify state	-
receive	External	payable	-
_addLiquidity	Internal	can modify state	-
addLiquidity	External	can modify state	ensure
addLiquidityHT	External	payable	ensure
removeLiquidity	Public	can modify state	ensure
removeLiquidityHT	Public	can modify state	ensure
removeLiquidityWithPermit	External	can modify state	-
removeLiquidityHTWithPermit	External	can modify state	-
removeLiquidityHTSupportingFeeOnTransferTokens	Public	can modify state	ensure
removeLiquidityHTWithPermitSupportingFeeOnTransferTokens	External	can modify state	-
_swap	Internal	can modify state	-

swapExactTokensForTokens	External	can modify state	ensure
swapTokensForExactTokens	External	can modify state	ensure
swapExactHTForTokens	External	payable	ensure
swapTokensForExactHT	External	can modify state	ensure
swapExactTokensForHT	External	can modify state	ensure
swapHTForExactTokens	External	payable	ensure
_swapSupportingFeeOnTransferTokens	Internal	can modify state	-
swapExactTokensForTokensSupportingFeeOnTransferTokens	External	can modify state	ensure
swapExactHTForTokensSupportingFeeOnTransferTokens	External	payable	ensure
swapExactTokensForHTSupportingFeeOnTransferTokens	External	can modify state	ensure
quote	Public	-	-

getAmountOut	Public	-	-
getAmountIn	Public	-	-
getAmountsOut	Public	-	-
geAmountsIn	Public	-	-

4.2 代码审计详情

4.2.1 增强建议

4.2.1.1 代币无法移除风险

HugswapPair 合约中的 burn 函数没有处理交易对中其中代币触发转账暂停的情况。当交易对中的其中一种代币暂停转账时，用户将无法取回流动性。

代码位置： HugswapPair burn 函数

```
    // this low-level function should be called from a contract which performs important safety checks
    function burn(address to) external lock returns (uint amount0, uint amount1) {
        (uint112 _reserve0, uint112 _reserve1,) = getReserves(); // gas savings
        address _token0 = token0;                                // gas savings
        address _token1 = token1;                                // gas savings
        uint balance0 = IERC20(_token0).balanceOf(address(this));
        uint balance1 = IERC20(_token1).balanceOf(address(this));
        uint liquidity = balanceOf[address(this)];

        bool feeOn = _mintFee(_reserve0, _reserve1);
        uint _totalSupply = totalSupply; // gas savings, must be defined here since totalSupply can update in _mintFee
        amount0 = liquidity.mul(balance0) / _totalSupply; // using balances ensures pro-rata distribution
        amount1 = liquidity.mul(balance1) / _totalSupply; // using balances ensures pro-rata distribution
        require(amount0 > 0 && amount1 > 0, 'Hugswap: INSUFFICIENT_LIQUIDITY_BURNED');
        _burn(address(this), liquidity);
        _safeTransfer(_token0, to, amount0);
        _safeTransfer(_token1, to, amount1);
        balance0 = IERC20(_token0).balanceOf(address(this));
```

```
        balance1 = IERC20(_token1).balanceOf(address(this));

        _update(balance0, balance1, _reserve0, _reserve1);
        if (feeOn) kLast = uint(reserve0).mul(reserve1); // reserve0 and reserve1 are up-to-date
        emit Burn(msg.sender, amount0, amount1, to);
    }
```

修复情况：经与项目方沟通后，确认将针对上述情况，在网站上对包含 pause 的 token 交易对进行公示，避免新用户在该交易对进行交易，已注入流动性的用户需与 pauseToken 的 owner 进行协商

5. 审计结果

5.1 总结

审计结论 ：通过

审计编号 ：0X002101220001

审计时间 ：2021 年 01 月 21 日

审计团队 ：慢雾安全团队

审计总结 ：慢雾安全团队采用人工结合内部工具对代码进行分析。审计期间暂未发现高危、中危及低危安全问题。提出了 1 点增强建议。经过与项目方沟通反馈确认审计过程中发现的风险均已修复或在可承受范围内。

6. 声明

慢雾仅就本报告出具前已经发生或存在的事实出具本报告，并就此承担相应责任。对于出具以后发生或存在的事实，慢雾无法判断其智能合约安全状况，亦不对此承担责任。本报告所作的安全审计分析及其他内容，仅基于信息提供者截至本报告出具时向慢雾提供的文件和资料(简称“已提供资料”)。慢雾假

设：已提供资料不存在缺失、被篡改、删减或隐瞒的情形。如已提供资料信息缺失、被篡改、删减、隐瞒或反映的情况与实际情况不符的，慢雾对由此而导致的损失和不利影响不承担任何责任。

慢雾科技
slow mist

官方网址

www.slowmist.com

电子邮箱

team@slowmist.com

微信公众号