1inch Liquidity Protocol audit

30 Jan 2021, Igor Gulamov

Introduction

Igor Gulamov conducted the audit of Mooniswap V2 smart contracts.

This review was performed by an independent reviewer under fixed rate.

Scope

Smart contract source code is stored at https://github.com/1inch-exchange/mooniswap-v2. Most issues are corresponding to efbe22e5f81d8165a1e3dcd4861b506932293777 commit.

Issues

We found no critical issues. The major issue takes place only for deflationary tokens, charging fees from the recipients in transfers, these tokens are very rare in DeFi.

Bug in `ExplicitLiquidVoting.sol` is fixed at e010aeb424bd778d746ac43bd7f1f931d27ec716.

Div 0 bug in `ReferralFeeReceiver.sol` is fixed at 23c9d4fc7d2b965ae404e994c4692c64d0cd3b28.

Bug with governance-managed default parameters in `FarmingRewards.sol` is fixed at 86c339151a8693bff73c0207d8f142620ecb0aa1.

We consider the commit 4e2df9a9a6f20c429e655474a708939e42074d24 as a safe version from the informational security point of view.

Major

1. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/Mooniswap.sol#L207

https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/Mooniswap.sol#L257

Current Mooniswap does not support deflationary tokens, taking fees from the sender's balance. We propose to update `UniERC20` to ensure that the sender's balance decreases by no more than tx amount or not reduced (for self tx case).

Comment
Won't fix

Warnings

1. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/governance/rewards/Rewards.sol#L31-L40

`updateReward` is unoptimized due to the usage of multiple `SLOAD`s from the same addresses. We suggest inline internal calls and cache local variables for gas optimization.

Comment
Won't fix

2. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/inch/GovernanceMothership.sol#L61

Unoptimized code produces a lot of unnecessary SLOADs. We propose to cache the list into memory as

```
    bytes32[] memory cached = _modules._inner._values;
```

and perform all further operations with a memory cached list.

Comment
Fixed at https://github.com/1inch-exchange/mooniswap-v2/commit/d81e0e11d9e045af96aa06d0dd24c43a7628a818

3. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/Mooniswap.sol#L317-L319

Silent partially implemented tokens check could be led to unexpected behavior of further function calls. We propose making this function pure and moving tokens check out of the function.

Comment
Won't fix

4. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/libraries/LiquidVoting.sol#L83-L85

Type conversion could lead to data loss.

Comment
Fixed at https://github.com/1inch-exchange/mooniswap-v2/pull/18

Comments

1. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/governance/rewards/Rewards.sol#L55

We recommend using specialized fixed point math libraries like `WadRayMath` for fixed-point computations.

Comment
Won't fix

2. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/governance/rewards/Rewards.sol#L81

The modifier is used for function purposes here. We suggest replacing the modifier to function.

Comment
Won't fix

3. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/governance/rewards/Rewards.sol#L24-L25

Explicit initialization of variables with zero value is unnecessary.

Comment
Fixed at https://github.com/1inch-exchange/mooniswap-v2/commit/381888499ebee4db2a0ceccb8118a1b71ccc3b19

4. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/Mooniswap.sol#L98

We recommend replacing the return data type with a fixed-sized array or tuple.

Comment
Now the path consists of the tokens list only.

5. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/utils/Converter.sol#L61

https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/utils/Converter.sol#L82

We recommend replacing a type of `path` with an array of tuples or add a length evenness check.

Comment
Won't fix

6. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/utils/Converter.sol#L88

We propose replacing `i + 2 < pathLength` with `i < pathLength-2` to increase readability and scope of potential compiler optimizations.

Comment
Won't fix

7. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/utils/Converter.sol#L83

We propose to remove this check, make the length even, and consider the last token is `targetToken` always.

Comment
Won't fix

8. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/utils/Converter.sol#L97

We recommend considering the no-swap case at the beginning of the function.

Comment
Won't fix

9. https://github.com/1inch-exchange/mooniswap-v2/blob/efbe22e5f81d8165a1e3dcd4861b506932293777/contracts/governance/GovernanceFeeReceiver.sol#L25

We propose collecting on-path tokens to reduce the number of swaps.

Comment
Won't fix

Severity Terms

Comment

Comment issues are generally subjective in nature, or potentially deal with topics like "best practices" or "readability". Comment issues in general will not indicate an actual problem or bug in code.

The maintainers should use their own judgment as to whether addressing these issues improves the codebase.

Warning

Warning issues are generally objective in nature but do not represent actual bugs or security problems.

These issues should be addressed unless there is a clear reason not to.

Major

Major issues will be things like bugs or security vulnerabilities. These issues may not be directly exploitable, or may require a certain condition to arise in order to be exploited.

Left unaddressed these issues are highly likely to cause problems with the operation of the contract or lead to a situation which allows the system to be exploited in some way.

Critical

Critical issues are directly exploitable bugs or security vulnerabilities.

Left unaddressed these issues are highly likely or guaranteed to cause major problems or potentially a full failure in the operations of the contract.