G₀ Group

Security Review of

Flooz Trade Multichain

March 2022

Flooz Trade Multichain / March 2022

Files in scope

Following files in:

https://github.com/flooz-link/flooz-trade-contracts/tree/654f606caa32fd0e0a98ab4f3f3b6a2766d29acb/contracts/

- FeeReceiverMultichain.sol
- FloozMultichainRouter.sol
- FeeReceiver.sol
- ReferralRegistry.sol
- libraries/PancakeLibrary.sol
- libraries/SafeMath.sol
- libraries/TransferHelper.sol

Current status

All issues except issue #6 have been fixed by the developer.

Issues

1. minOut check in executeZeroExSwap and executeOneInchSwap doesn't work when output token is ETH

type: incorrect implementation / severity: medium

When `swapData.toToken == address(0)`, checks on `line 604` and `line 523` in `FloozMultichainRouter.sol` will always fail.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/flooz-link/flooz-trade-contracts/tree/2c2c8faa22a62ae0c4842a113db21a011965aa5c/contracts

2. referrer can DoS their referee

type: security / severity: medium

When receiving ETH fee in `FloozMultichainRouter`, referrer contract can throw, which will make the whole transaction fail. Since referrer can't be changed, this allows referrers to completely block trading of their referees.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/flooz-link/flooz-trade-contracts/tree/2c2c8faa22a62ae0c4842a113db21a011965aa5c/contracts

3. minOut check is not performed in executeOneInchSwap when fee is not used

type: security / severity: minor

In `executeOneInchSwap` unlike in `executeZeroExSwap` the `minOut` check is only performed when fee is used.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/flooz-link/flooz-trade-contracts/tree/2c2c8faa22a62ae0c4842a113db21a011965aa5c/contracts

4. implicit assumption that swapData.fromToken != swapData.toToken in executeOneInchSwap & executeZeroExSwap should be enforced

type: security / severity: minor

There's an assumption that `swapData.fromToken != swapData.toToken` in `executeOneInchSwap` & `executeZeroExSwap` functions, this is not necessarily true and should be enforced to prevent unexpected behavior.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/flooz-link/flooz-trade-contracts/tree/2c2c8faa22a62ae0c4842a113db21a011965aa5c/contracts

5. executeOneInchSwap & executeZeroExSwap should use safeTransferFrom instead of transferFrom

type: security / severity: minor

`executeOneInchSwap` & `executeZeroExSwap` should use safe implementation of the `transferFrom` function.

status - fixed

Issue has been fixed and is no longer present in

https://github.com/flooz-link/flooz-trade-contracts/tree/2c2c8faa22a62ae0c4842a113db21a011965aa5c/contracts

6. FeeReceiver.executeBuyback & FeeReceiver.convertToETH are vulnerable to price manipulation attacks

type: security / severity: major

`FeeReceiver.executeBuyback` & `FeeReceiver.convertToETH` are vulnerable to price manipulation attacks, an attacker can artificially increases the price of the asset being bought by the contract by buying it in advance, then trigger the buyback function which will drive the price even higher and then immediately sell the tokens back with a profit. This can lead to privatisation of collected fees by malicious third parties.

status - aknowledged

Developer's comment: Issue aknowledged by our team and will be addresses in the future, by leveraging on-chain time weighted price oracles. For now the risk of price manipulation is rated low, as the SYA token has a implemented fee, which makes price manipulation unlikely. Additionally, the process of executeBuyback will be automated to trigger multiple buybacks a day, so that no big amounts of SYA will be accumulated."