ATTIC LAB

Review Report

Produced by Attic Lab
for StableXSwap

November 27th, 2020

Critical Vulnerabilities

0

Suggestions and Recommendations

8

Summary

We've conducted a thorough review of smart contracts in StableXSwap service, namely StableXSwap Core and StableXSwap Periphery located in the following repositories:

`https://github.com/stablexlabs/stablexswap-core`
`https://github.com/stablexlabs/stablex-periphery`

Review was conducted on the master branch of each repository, commits `a013372f77734b18b8aaf841ef8a8c5540d1fd10` and `eb160b5bd7f9b378e8b6a31ac4e0144149a47ae1` respectively.

Overall, those two repositories have a lot in common with PancakeSwap contracts located here `https://github.com/pancakeswap.` So our review will assume PancakeSwap as a time-proven reference and only focus on contract differences.

Security

Since the code very closely resembles PancakeSwap, there are only several places where new potential vulnerabilities could be introduced. Each of these places is described in detail below.

Proxy Contracts

StableXSwap introduces proxy contracts for `StableXFactory` and `StableXRouter` contracts ensuring their upgradability. Both these contracts are based on well-known OpenZeppelin proxy contracts which do not have any known security issues.
The only recommendation is to try to use the same version of the OpenZeppelin contract for both proxies, which is `TransparentUpgradeableProxy.`

New Transaction

There is only one place in the code where an extra transaction is added: where an extra 1% fee is sent to the fee collection address in the `burn` method of the `StableXPair` contract. Since it uses the same address as used in the minting fee in the default PancakeSwap contract no new vulnerabilities are added here.

New Contract State

StableXRouter contract was changed to have its parameters mutable and initialized later by the initialize method. It was done through a well-known OpenZeppelin Initializable pattern, so there should be no issues.

One small recommendation would be to add an extra check for contract being initialized in its methods, so no undefined behaviour happens in this case.

Logic Review and Recommendations

We've carefully reviewed the rest of the changes and here are our comments and recommendations.

stablexswap-core
StableXPair
Fee change in _mintFee

Change 1/4th fee in Pancake swap to 4/5th fee. Btw, there is an error in the comment in Pancake swap indicating that the fee is 1/6th, in fact it is 1/4th.

Using Uniswap whitepaper as a reference we followed the math and arrived at the same formulas as currently used in StableXPair changes.

Extra fees in burn

A couple of points to consider:

1. There are two calls to get the fee address now, once in _mintFee and once in burn. Consider returning address instead of boolean from _mintFee so it can be used to send additional fee.

2. When fee in LP token is calculated, then 1% is calculated first and then subtracted from the full amount. In the case of the token fee 99% is calculated directly instead. We recommend changing the calculation there to subtracting 1% as well to make it more consistent.

```
amount0 -= amount0.div(100);
amount1 -= amount1.div(100);
```

3. Another idea would be to optimize transactions and get rid of the extra transaction completely. Currently the fee is calculated and minted (in _mintFee method), then 1% of the burnable amount is transferred to the same address and then the rest of the amount is burned. How about changing it to have only one mint (of regular fee + 1%) and then 100% burn? This way there will be one transaction less. If you're concerned about keeping the amounts in the blockchain you can specify them in contract events instead.

4. It is now stated in the comments that the fee can be adjusted later, so we recommend to store fee in the contract instead of making it hard coded and make this fee adjustable by the admin.

Fee calculation change in `swap`

Fee calculation changed from 0.2% to 0.06%. No comments or suggestions, all changes are straightforward.

`SafeMath`

Added div methods from OpenZeppelin's SafeMath library. Consider removing this file completely and simply referencing OpenZeppelin's library instead.

`stablex-periphery`
`StableXLibrary`
`getAmountOut`

Amounts adjusted to reflect fee change from 0.2% to 0.06%. No comments or suggestions, all changes are straightforward.

`getAmountIn`

Amounts adjusted to reflect fee change from 0.2% to 0.06%. No comments or suggestions, all changes are straightforward.

Tests

All existing tests are updated to work with the changes done in StableXSwap. The only thing we recommend is to add tests specifically targeted to fee amounts.

Other Comments

Solidity Version

Solidity versions are varied across the contract files, we strongly recommend updating it to use the same version across the whole project.

Code Style

While it does not affect functionality we recommend formatting the code to make it look more consistent and readable. This includes proper code indentation and using excessive empty lines. This applies mostly to the `StableXPair` contract.