**Spartan Protocol Bug**

The spartan protocol is a pool that allows users to swap between two tokens – BASE and TOKEN. The swap rates are maintained by AMM and there are some stability optimization for the benefit of the liquidity providers.

Spartan protocol has been hacked for 30M$ of liquidity providers funds. The bug isn’t at one specific function – but rather the use of two functions, addLiquidity and removeLiquidity, led to a bug that allows the attacker to drain the pool funds.

This summary explains the two functions logic, and then how the attacker exploited it.

**addLiquidity:**

1. Transfer tokens to spartan account.
2. Then the difference between the current spartan account tokens amount to the tokens amount recorded before the transfer is the transferred amount, mint LP token for this amount.

If 1 and 2 happens in different transactions, then the transferred tokens can be use by another account that also added liquidity and its transaction took place between 1 and 2. The correctness holds only if 1+2 are done at one transaction.

**removeLiquidity:**

1. Calculate LP token worth by calculating the portion of the those LP token from the total LP tokens and multiply by **current pool balance**.
2. Burn the tokens and pay this amount.

**The bug:**

We explain the bug by showing a simplified version of the actual attack. Suppose at the starting point of the transaction the pool balance is , and there are liquidity provider tokens minted.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attacker operation – all happens at one transaction | Total LP token supply | Attacker tokens | Actual pool balance | Recorded pool balance | Attacker cash |
|  |  |  |  |  |  |
| addLiquidity of and get tokens |  |  |  |  |  |
| transfer to pool (first step of addLiquidity without the second step) |  |  |  |  |  |
| removeLiquidity of the tokens |  |  |  |  |  |
| step 2 of the addLiquidity – the difference is counted to the user. |  |  |  |  |  |
| removeLiquidity of tokens |  |  |  |  |  |

The attacker ends with total cash amount of where at the beginning he had of cash. All of those steps must happen at the same transaction.

**How to fix** – in remove liquidity sync the pool actual and recorded balances… maybe require that are the same. Or punish who tries this things and make the recorded be the actual…

**This summary is written assuming the attacker is a man, although I don’t know if it’s true.**