SB SECURITY

Sparta

Security Review

Feb 14, 2025

Conducted by:
Blckhv, Lead Security Researcher
Slavcheww, Lead Security Researcher

Contents

1. About SBSecurity

SBSecurity is a duo of skilled smart contract security researchers. Based on the audits conducted and numerous vulnerabilities reported, we strive to provide the absolute best security service and client satisfaction. While it's understood that 100% security and bug-free code cannot be guaranteed by anyone, we are committed to giving our utmost to provide the best possible outcome for you and your product.

Book a Security Review with us at sbsecurity.net or reach out on Twitter @Slavcheww.

2. Disclaimer

A smart contract security review can only show the presence of vulnerabilities **but not their absence**. Audits are a time, resource, and expertise-bound effort where skilled technicians evaluate the codebase and their dependencies using various techniques to find as many flaws as possible and suggest security-related improvements. We as a company stand behind our brand and the level of service that is provided but also recommend subsequent security reviews, on-chain monitoring, and high whitehat incentivization.

3. Risk classification

	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

3.1. Impact

- **High** - leads to a significant loss of assets in the protocol or significantly harms a group of users.
- **Medium** - leads to a moderate loss of assets in the protocol or some disruption of the protocol's functionality.
- **Low** - funds are not at risk.

3.2. Likelihood

- **High** - almost **certain** to happen, easy to perform, or highly incentivized.
- **Medium** - only **conditionally possible**, but still relatively likely.
- **Low** - requires specific state or **little-to-no incentive**.

3.3. Action required for severity levels

- High - **Must** fix (before deployment if not already deployed).
- Medium - **Should** fix.
- Low - **Could** fix.

4. Executive Summary

Overview

Project	Sparta
Repository	Private
Commit Hash	a5e5b8f942bd17023f00107506da105f5d01b0c2
Resolution	3d42f456ad9ec82320bb8031a1766b4a49b8c43a
Timeline	February 11 - February 14, 2025

Scope

Sparta.sol
Airdrop.sol

Issues Found

Critical Risk	0
High Risk	0
Medium Risk	1
Low/Info Risk	4

5. Findings

5.1. Medium severity

5.1.1. `Sparta::_fixV2PoolAndAddLiquidity` can be blocked with a big enough pool donation

Severity: Medium Risk

Description: In `_fixV2PoolAndAddLiquidity`, `spartaSent` can be manipulated to calculate an arbitrary big number, bigger than the `totalSupply` of Sparta.

In order to dos the initialization, attacker can transfer, let's say `56735e8` (~$219 at the time of writing) `hex` token to the pool. `spartaSent = (56735e8 * 1804555792.50e18) / 51190e8` = `2000028772e18`

Note that, amount hex donated can be lower but then owner will have to sacrifice the token allocated for airdrop.

Furthermore, with `51300e8` ($198) hex he can make grief the supplying because `spartaSent = (51300e8 * 1804555792.50e18) / 51190e8` > `1804555792.5` (`spartaLpAmount`) and the `amount0Desired` will underflow.

```
function _fixV2PoolAndAddLiquidity(
    uint256 spartaLpAmount,
    uint256 hexLpAmount,
    uint256 deadline
) internal {
    uint256 hexBalance = IERC20(hexToken).balanceOf(uniswapV2Pair);
    uint256 spartaSent;

    if (hexBalance > 0) {
        spartaSent = (hexBalance * spartaLpAmount) / hexLpAmount;
        _transfer(address(this), uniswapV2Pair, spartaSent);

        IUniswapV2Pair(uniswapV2Pair).sync();
    }

    IUniswapV2Router02(uniswapV2Router).addLiquidity(
        address(this),
        hexToken,
        spartaLpAmount - spartaSent,
        hexLpAmount - hexBalance,
        ((spartaLpAmount - spartaSent) * (BPS - SLIPPAGE)) / BPS,
        ((hexLpAmount - hexBalance) * (BPS - SLIPPAGE)) / BPS,
        msg.sender,
        deadline
    );
}
```

Recommendation: Consider rewriting the fix logic as follows, now if the attacker has send a lot of hex, you will receive better initial price for the `Sparta` token.

<u>**Resolution:** Fixed</u>

5.2. Low/Info severity

5.2.1. Lows, Informational issues and code suggestions

Description:

1. missing access control in `Sparta::init`, despite not posing any security issue, because the `addLiquidity` will be reverting since no sparta tokens are transferred to the pair, it's better to have `onlyOwner` modifier that prevents potential malicious initializations.

2. in `Sparta::_fixV2PoolAndAddLiquidity` `addLiquidity`'s amountsDesired can be calculated prior to assigning them as arguments and then reused in amountsMin with the desired slippage.

3. in `Sparta::init` missing address(0) for the airdrop contract.

4. in `Airdrop`, if single address can claim only once `batchClaim` can be removed as it it's non functional.

Resolution: Fixed - 3,4