**1. Critical**   
 I changed the code according to Certik recommendation.

Virgo.sol

    function emergencyWithdraw(uint256 \_pid) public {

        PoolInfo storage pool = poolInfo[\_pid];

        UserInfo storage user = userInfo[\_pid][msg.sender];

        pool.lpToken.safeTransfer(address(msg.sender), user.amount);

        emit EmergencyWithdraw(msg.sender, \_pid, user.amount);

        user.amount = 0;

        user.rewardDebt = 0;

    }

Changed Virgo.sol

    function emergencyWithdraw(uint256 \_pid) public {

        PoolInfo storage pool = poolInfo[\_pid];

        UserInfo storage user = userInfo[\_pid][msg.sender];

        if(\_pid == 0) {

            spica.burn(msg.sender,user.amount);

        }

    uint256amount =user.amount;

    user.amount = 0;

    user.rewardDebt = 0;

    pool.lpToken.safeTransfer(address(msg.sender),amount);

    emit EmergencyWithdraw(msg.sender,\_pid,amount);

    }

**2. Major**

In most cases, it is recommended to use timelock function and multisig wallet because the authority should not be taken over when executing the smart contract.  
 I already set the timelock at whole source, so I changed 48h delay as they recommend.

*Timelock.sol refer below*

Multi signature for metamask  
<https://www.reddit.com/r/Metamask/comments/p1yepl/how_to_add_multisig_for_wallet/>

텍스트이(가) 표시된 사진

자동 생성된 설명

**Timelock.sol**

텍스트이(가) 표시된 사진

자동 생성된 설명

**3. Minor**

In most cases, I changed the code according to Certik recommendation..