Valuedefi incident: re-initialize()

Summary

2021년 5월 5일 3:22 UTC, valuedefi의 ProfitSharingRewardPool 컨트랙트의 결함으로 vBWAP/BUSD LP에서 11M USD 규모의 drain이 발생했다. 원인은 코드 한 줄을 누락한 **human error로** 밝혀졌다.

The Cause

https://bscscan.com/address/0x7a8ac384d3a9086afcc13eb58e90916f17affc89#code

```
contract ProfitSharingRewardPool {
...
bool public initialized = false;
...
modifier notInitialized() {
    require(!initialized, "ProfitSharingRewardPool: initialized");
    _;
}
...
function initialize(
    address _stakeToken,
    address _wbnb,
    address _busd,
    address _reserveFund,
    uint256 _startRewardBlock
) public notInitialized {
    require(block.number < _startRewardBlock, "late");

    stakeToken = _stakeToken;
    wbnb = _wbnb;
    busd = _busd;
    reserveFund = _reserveFund;
    startRewardBlock = _startRewardBlock;
    endRewardBlock = _startRewardBlock;

    operator = msg.sender;
    _locked = 0;

    setRewardPool(_wbnb, _startRewardBlock);
    setRewardPool(_busd, _startRewardBlock);
}
```

initialize 함수는 notInitialized modifier를 통해서 initialized 변수가 true이면 이미 초기화되었다고 판단한다. 따라서 initialize 함수에서 `initialized = true;` 구문이 존재해야 한다. 하지만 위 컨트랙트에서 그 코드가 누락되었고 아무나 initialize 함수를 호출할 수 있었다. initialize 함수에 operator를 초기화하는 코드(`operator = msg.sender;`)가 있기 때문에 initialize를 호출한 msg.sender는 이 컨트랙트를 장악하게 된다.

Attack Senario

1. re-initialize()

   initialize() 내부의 `operator = msg.sender;`을 통해서 컨트랙트 통제권 장악

   https://bscscan.com/tx/0xd3382252bc204fdc32a6b3add8c639850882b70a798399d6e00a542cdf769040

2. governanceRecoverUnsupported()

```
modifier onlyOperator() {
    require(operator == msg.sender, "ProfitSharingRewardPool: caller is not the operator");
    _;
}
...
// This function allows governance to take unsupported tokens out of the contract. This is in an effort to make someone whole, sho
// There is no guarantee governance will vote to return these. It also allows for removal of airdropped tokens.
function governanceRecoverUnsupported(
    IERC20 _token,
    uint256 amount,
    address to
) external onlyOperator {
```

```
    require(address(_token) != stakeToken, "stakeToken");
    _token.safeTransfer(to, amount);
}
```

onlyOperator를 통과하고 safeTransfer를 호출해 컨트랙트 잔고에서 to에게 amount만큼 전송

https://bscscan.com/tx/0x9ba0454c2301ad5780795ae7477e9fa7e38226be16cc282158624479e66389b6

https://bscscan.com/tx/0x77e68d369cae24cbac2e87252f192f7a26fd48c77e71e1b1b48803971d4b932d

https://bscscan.com/tx/0x9ee8d54da43218126171bcb1615032fb692624f2c9b7f82307d1988df75f296a

Fix

initialize()에 `initialized = true` 구문 추가

```
function initialize(
    address _stakeToken,
    address _wbnb,
    address _busd,
    address _reserveFund,
    uint256 _startRewardBlock
) public notInitialized {
    require(block.number < _startRewardBlock, "late");

    stakeToken = _stakeToken;
    wbnb = _wbnb;
    busd = _busd;
    reserveFund = _reserveFund;
    startRewardBlock = _startRewardBlock;
    endRewardBlock = _startRewardBlock;

    operator = msg.sender;
    _locked = 0;

    initialized = true

    setRewardPool(_wbnb, _startRewardBlock);
    setRewardPool(_busd, _startRewardBlock);
}
```

Reference

https://medium.com/valuedefi/vstake-pool-incident-post-mortem-4550407c9714

https://rekt.news/value-rekt2/