Final Security Audit of Narfex MasterChef contract by profiterole.group

Conclusion

This audit was made by Profiterole Group https://profiterole.group/
Auditor: Vladimir Smelov.
Date: 2023-04-15

Based on the evaluation, the Client's smart contracts are considered secure.

Our team conducted a code functionality analysis, manual examination, test cases and automated tests using static analyzers. All issues identified during the automated analysis were manually inspected, and significant vulnerabilities are highlighted in the Audit Summary section.

The Client has acknowledged and fixed or accepted all the issues.

Scope

Pre Audit Scope

repo: https://github.com/narfex/masterchef
commit: `916751ad7f8679c54ee5de2b5c87d72958bd7e7e`

- contracts/MasterChef.sol (md5 `efe5d889e727a7e55f4898d740ee61fa`)

Final Audit Scope

repo: https://github.com/narfex/masterchef
commit: `b6544d0f7501208b4c841e09f031b392bddda5f0`

- contracts/MasterChef.sol (md5 `f49b20a87dec35690d9ceb6ca7ee3286`)
- contracts/utils/EmergencyState.sol (md5 `0613e96e5beed801115b9f8039659aa3`)

Final Deployed Contract

MasterChef on Ethereum -
https://etherscan.io/address/0x877Ec50E446B74058CFA38935CCe83408B71D746#code

Methodology

1. Blind audit. Understand the structure of the code without reading any docs.
2. Questions to developers.
3. Execute static analyzers.
4. Identify problems with:

- backdoors;
- bugs;
- mathematical errors;
- potential fund leakage;
- potential contract lockup;
- argument and event validation;
- others.

5. Draft report formation and discussing results with the client.
6. Final report, ensuring all security vulnerabilities have been fixed.

Contract desrciption

This contract represents a decentralized farming platform focused on the NRFX token, which allows users to deposit various types of liquidity provider (LP) tokens into the platform in order to earn rewards. The platform supports multiple pools, each with its own allocation of rewards, represented by the NRFX token. Users can deposit their LP tokens into a specific pool, and the contract will distribute rewards to users based on their share of the total deposited tokens in the pool. The contract also includes functionality for referral rewards, early withdrawal fees, and other mechanisms to optimize reward distribution and encourage user participation.

The core functionality of the contract includes depositing LP tokens, withdrawing LP tokens, and harvesting rewards. Users can deposit their LP tokens into a specific pool, optionally specifying a referral address to receive a referral reward. The contract also allows for the withdrawal of LP tokens and harvesting of earned rewards. Additionally, the contract includes safeguards against various attack vectors, such as Reentrancy and Emergency situations, and ensures the proper accounting and distribution of rewards. The contract also has functionality to recover accidentally sent tokens, maintaining the overall safety and security of the platform.

Results

MAJOR-1.

At

- MasterChef.sol:219 https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L219

```
        return !isEarlyWithdraw;
```

should it be just “isEarlyWithdraw” not “!isEarlyWithdraw” otherwise the result will be opposite.

Status.

FIXED - the function was removed.

MAJOR-2.

At

- MasterChef.sol:226-229 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L226-L229
- MasterChef.sol:215-220 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L215-L220
- MasterChef.sol:204-209 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L204-L209
- MasterChef.sol:185-198 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L185-L198
- MasterChef.sol:291 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L291
- MasterChef.sol:269 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L269

```
    function getUserPoolSize(address _pairAddress, address _user) public view
        uint256 _pid = poolId[_pairAddress];
        return userInfo[_pid][_user].amount;
    }
```

```
    function getIsEarlyWithdraw(address _pairAddress, address _user) internal
        uint256 _pid = poolId[_pairAddress];
        UserInfo storage user = userInfo[_pid][_user];
        bool isEarlyWithdraw = block.timestamp - user.depositTimestamp < early
        return !isEarlyWithdraw;
    }
```

```
    function getIsUserCanHarvest(address _pairAddress, address _user) internal
        uint256 _pid = poolId[_pairAddress];
        UserInfo storage user = userInfo[_pid][_user];
        bool isEarlyHarvest = block.timestamp - user.harvestTimestamp < harves
        return !isEarlyHarvest;
    }
```

```
    function getUserReward(address _pairAddress, address _user) public view re
        uint256 _pid = poolId[_pairAddress];
        PoolInfo storage pool = poolInfo[_pid];
        UserInfo storage user = userInfo[_pid][_user];
        uint256 accRewardPerShare = pool.accRewardPerShare;
        uint256 lpSupply = pool.pairToken.balanceOf(address(this));
        if (block.number > pool.lastRewardBlock && lpSupply != 0) {
            uint256 blocks = block.number - pool.lastRewardBlock;
            uint256 totalReward = lastBalance * blocks - k * blocks**2 / 2;
            uint256 reward = totalReward * pool.allocPoint / totalAllocPoint;
            accRewardPerShare += reward * 1e12 / lpSupply;
        }
        return user.amount * accRewardPerShare / 1e12 - user.withdrawnReward +
    }
```

```
    function getPoolUserData(address _pairAddress, address _user) public view
```

```
        uint256 _pid = poolId[_pairAddress];
```

if there is no poolId for not-exist _pairAddress, then _pid=0, and the function will return amount of the user 0 pool (which has different LP token), consider adding _poolExists check. You can use it as a modifier like onlyExistPool(_pairAddress) and use it on every method with _pairAddress argument

Status.

FIXED - `onlyExistPool` modifier is placed on every public function with `_pairAddress` argument.

MAJOR-3.

At

- MasterChef.sol:409 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L409
- MasterChef.sol:430 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L430

- MasterChef.sol:443 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L443

```
    function withdraw(address _pairAddress, uint256 _amount) public {
```

```
    function emergencyWithdraw(address _pairAddress) public {
```

```
    function harvest(address _pairAddress) public {
```

nonReentrant modifier is forgotten.

Status.

FIXED - `nonReentrant` modifier is placed on every public function changing state, preventing reentry attacks.

MAJOR-4.

There could be unfair rewards distribution in case if some used left his reward locked for a long time and during this time other users have harvested all available rewards. In this case long-lock user will receive no reward which is unfair.

Status.

FIXED - `endBlock` was introduced to guarantee that rewards on pool with the current `rewardsPerBlock` will be fairly distributed till `endBlock`

WARNING-1.

At

- MasterChef.sol:66 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L66

```
    uint256 public lastBalance;
```

why we need it if we can use narfex.balanceOf(address(this))

Status.

FIXED - the attribute was removed.
There was another `lastRewardTokenBalance` added to account new rewards.

WARNING-2.

At

- MasterChef.sol:157 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L157

```
        if (_withUpdate) {
```

At

- MasterChef.sol:173 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L173

```
        if (_withUpdate) {
```

should it be required?

Status.

FIXED - `_massUpdatePools` is always called.

WARNING-3.

At

- MasterChef.sol:353 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L353
- MasterChef.sol:190 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L190

```
        uint256 lpSupply = pool.pairToken.balanceOf(address(this));
```

```
        uint256 lpSupply = pool.pairToken.balanceOf(address(this));
```

im not sure but what will happen if someone just transfer LP to the balance of this contract (not with deposit but just with erc20 transfer), will it ruin the calculations and lead to mistakes in reward sharing?

Status.

FIXED - `pool.totalDeposited` is used instead.
Moreover, LP tokens occasionaly sent to the contract may be recovered via `recoverERC20` (except of deposited).

WARNING-4.

At

- MasterChef.sol:436 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L436

```
            user.amount = 0;
```

better to reset storage before external call

Status.

FIXED -

1. `nonReentrant` modifier is used.
2. storage reset placed before external call.

LOW-1.

At

- MasterChef.sol:98 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L98

```
        /// @notice Returns the soil fertility
```

what is “soil fertility”

Status.

FIXED - rephrased

LOW-2.

At

- MasterChef.sol:110 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L110

- MasterChef.sol:421 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L421
- MasterChef.sol:434 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L434

```
        rewardToken.safeTransfer(address(msg.sender), amount);
```

```
            pool.pairToken.safeTransfer(address(msg.sender), _amount);
```

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

msg.sender is already an address no conversions needed

Status.

ACKNOWLEDGED

LOW-3.

At

- MasterChef.sol:193 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L193

```
            uint256 totalReward = lastBalance * blocks - k * blocks**2 / 2;
```

unclear math. Consider the usage of some constant-speed rewards distribution.

Status.

FIXED - constant-speed rewards distribution is used.

LOW-4.

At

- MasterChef.sol:194 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L194

```
            uint256 reward = totalReward * pool.allocPoint / totalAllocPoint;
```

it should check that totalAllocPoint!=0

Status.

ACKNOWLEDGED - totalAllocPoint may never ne zero because the owner will add pools with positive alloc points right after the deploy.

LOW-5.

At

- MasterChef.sol:204 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L204

```
    function getIsUserCanHarvest(address _pairAddress, address _user) internal
```

prefix with _

Status.

FIXED - declared public

LOW-6.

At

- MasterChef.sol:215 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L215

```
    function getIsEarlyWithdraw(address _pairAddress, address _user) internal
```

prefix with _

Status.

FIXED - removed

LOW-7.

At

- MasterChef.sol:226 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L226

```
    function getUserPoolSize(address _pairAddress, address _user) public view
```

better to rename to getUserPoolAmount

Status.

ACKNOWLEDGED

LOW-8.

At

- MasterChef.sol:462 - https://github.com/narfex/masterchef/blob/916751ad7f8679c54ee5de2b5c87d72958bd7e7e/contracts/MasterChef.sol#L462

```
    function rewardTransfer(UserInfo storage user, uint256 _amount, bool isWit
```

prefix with _

Status.

FIXED - renamed

LOW-9.

At

- MasterChef.sol:757-762 - https://github.com/narfex/masterchef/blob/b6544d0f7501208b4c841e09f031b392bddda5f0/contracts/MasterChef.sol#L757

```
/// @notice Sets the early harvest commission
/// @param percents Early harvest commission in percents denominated by 10000
function setEarlyHarvestCommission(uint percents) external onlyOwner nonReentr
    earlyHarvestCommission = percents;
    emit EarlyHarvestCommissionSet(percents);
}
```

- MasterChef.sol:737-740 - https://github.com/narfex/masterchef/blob/b6544d0f7501208b4c841e09f031b392bddda5f0/contracts/MasterChef.sol#L737

```
function setEarlyHarvestCommissionInterval(uint interval) external onlyOwner n
    earlyHarvestCommissionInterval = interval;
    emit EarlyHarvestCommissionIntervalSet(interval);
}
```

global attributes `earlyHarvestCommission`, `earlyHarvestCommissionInterval` are not used, consider removal

Status.

ACKNOWLEDGED - it’s not a functional problem