GEMZ
AUDIT REPORT

Security Report Prepared On
February 22 2021

ICECREAMSWAP V2
CODE REVIEW

TABLE OF CONTENTS

SUMMARY

What follows is a code security audit of the IceCreamSwap V2 project on Binance Smart Chain.

Website: https://IceCreamSwap.finance
Telegram: https://t.me/IceCreamSwap

The information appearing in this report is for general purposes only and is not intended to provide any legal security guarantees to any individual or entity. It is advisable to conduct more reviews or/and audits as just one review or audit may miss important issues.

This report does not provide personalized investment advice or recommendations, nor does it provide advice to perform any transactions and it does not provide investment, financial, legal, or tax advice. We are not responsible or liable for any loss which results from the report. This report should not be considered as investment advice.

REPORT

FARM

https://github.com/IceCreamSwap/contracts/tree/main/farm-contracts

CreamToken.sol

**This is a clone of https://github.com/pancakeswap/pancake-farm/blob/master/contracts/CakeToken.sol.
No functional changes are present.**

MasterChef.sol

The owner of the contract has the ability to do the following:

```
add new pools
modify existing pools'_allocPoint
updateMultiplier
updateBonus
updateCreamPerBlock
setDevFee
setTaxAddr
setTax
```

The onlyOwner modifier enforces access control for the functions to be called by the owner. The checks in the previous version which use require and have more than one owner (governance and owner) address have all been removed. There is only one owner now.

Additional checks in updateMultiplier and updateCreamPerBlock are added. Multiplier cannot be set to be above 3, and creamPerBlock cannot be set to be above 4 * 1e18. This will limit setting the emission per block to an absurdly high number, but a pool with a very high allocPoint will still be able to get most of the entire farm's emission per block.

Additionally, the migrator function has been removed.

REPORT

MasterChef.sol Continued

Fix added to mint 100% of the block reward instead of 100% + 5% (dev fee is included as 5% of the 100%) like in other MasterChef clones.

```
        // fix: to avoid printing 105%
        uint256 creamDevReward = creamReward.div(20); // dev fee 5%
        uint256 creamUserReward = creamReward.sub(creamDevReward);
        cream.mint(devFee, creamDevReward );
        cream.mint(address(milkshake), creamUserReward);
```

Developers fixed the syrup (milkshake) bug in emergencyWithdraw by forcing a burn of Shake tokens equal to the same amount of ICS withdrawn by the user. If the user does not have enough Shake tokens, the function will revert.

```
297     // Withdraw without caring about rewards. EMERGENCY ONLY.
298     function emergencyWithdraw(uint256 _pid) public {
299         PoolInfo storage pool = poolInfo[_pid];
300         UserInfo storage user = userInfo[_pid][msg.sender];
301         if(_pid == 0) {
302             milkshake.burn(msg.sender, user.amount );
303         }
304         pool.lpToken.safeTransfer(address(msg.sender), user.amount);
305         emit EmergencyWithdraw(msg.sender, _pid, user.amount);
306         user.amount = 0;
307         user.rewardDebt = 0;
308     }
309
```

When the MasterChef contract was deployed, the owner was verified to be the time-lock contract and not an EOA.

MilkShake.sol

This is a clone of https://github.com/pancakeswap/pancake-farm/blob/master/contracts/SyrupBar.sol, but with some functional changes made.

One of these new functions includes the taxation of harvests. setTax and setTaxAddr are both functions used to set and change the tax amount and destination address.

These can only be called by owner (MasterChef contract), which is done with the onlyOwner modifier check. The maximum value for the tax can be set to 100 (10%).

REPORT

MilkShake.sol Continued

taxUser is a private function used in safeCreamTransfer to deduct an amount of harvested farm tokens which will be set to the taxAddr. This only affects the harvested cream, not the deposited LP token.

The event MilkShakeTransfer is defined twice with different arguments. While few things may need this event, it's still recommended to have a single compatible definition.

SmartChef.sol

SmartChef has only one pool instead of an array of pools. This means that each SmartChef deployed contract can only support one token to stake and one reward token.

The owner can call the following functions, with access control by using the onlyOwner modifier:

```
startReward
stopReward
emergencyRewardWithdraw
```

An explanation on why the SmartChef needs to have functions to arbitrarily stop or restart rewards would be helpful. A _governance address variable was added but does not appear to be used.

CreamRecovery.sol

After deployment, ownership was renounced to 0x0 to prevent additional minting.

Timelock.sol

There is a 24 hour time lock. The minimum delay is 6 hours, meaning it is possible to lower it to that. However, the setDelay is also time-locked, and requires it to be called from the time-lock contract itself.

Previously multiple admins could time-lock, but a change was made to only allow one administrator now.

REPORT

SWAP

https://github.com/IceCreamSwap/contracts/tree/main/swap-contracts (Compared with https://github.com/sushiswap/sushiswap/blob/master/contracts/uniswapv2/)

UniswapV2ERC20.sol

This has the same functionality as Sushiswap.

UniswapV2Factory.sol

This has the same functionality as Sushiswap, but with the migrator related code removed.

UniswapV2Pair.sol

This is similar to Sushiswap.

uint denominator = rootK.mul(15).add(rootKLast); in _mintFee. Using 15 instead of 5, which is used in Sushiswap, which means out of the 0.30% fees, 0.15% is for liquidity providers, and 0.15% is for the dev fund.

This is clarified here: https://ice-cream-swap.gitbook.io/icecreamswap/roadmap/icecreamswap-exchange

The migrator code portions have been removed.

UniswapV2Router02.sol

This is the same as Sushiswap.

RISK FACTORS

ISSUES

ICS-001: devFee address variable is misnamed as devFee

Severity: Info

The devFee variable is actually the address that the devFee is sent to, instead of the amount of devFee percentage.

Recommendations

It is recommended to rename devFee to devFeeAddr to reflect this.

Resolution

Contract code was modified to rename the variable, and matches the published mainnet contract.

ICS-002: Unused governance address in SmartChef.sol

Severity: Info

A _governance address variable was added to SmartChef.sol, but does not appear to be used.

Recommendations

Remove the governance address variable from the smart contract.

Resolution

The governance address was removed from the code and matches the published mainnet contract.

ICS-003: Remove unnecessary commented out code

Severity: Info

In UniswapV2Pair.sol, there is a fee_to variable that is commented out.

Recommendations

Remove unused comments such as fee_to for code readability purposes.

Resolution

The commented lines were removed from the code and matches the published mainnet contract.

RISK FACTORS

ISSUES

ICS-004: SmartChef owner can arbitrarily stop or restart rewards

Severity: Info

In SmartChef.sol, there are 2 functions; startReward and endReward. Both of which can be called by the owner to set the value of startBlock and endBonusBlock.

Recommendations

If there is no need for such functionality to stop/restart rewards, the above mentioned functions should be removed. Otherwise, explain in the documentation why such functionality is required.

Resolution

These methods were removed from the code and matches the published mainnet contract.

ICS-005: MilkshakeTransfer event is defined twice

Severity: Info

The MilkshakeTransfer event is declared twice, with different function arguments.

```
event MilkShakeTransfer(address indexed user, uint256
amount, uint256 tax);

event MilkShakeTransfer(address _to, uint256 _total, uint256
_amount, uint256 _tax, uint256 creamBal);
```

In the code, the latter is used.

Recommendations

Remove the first instance duplicate event.

Resolution

The more-simplistic Event signature was removed in favor of MilkShakeTransfer(address _to, uint256 _total, uint256 _amount, uint256 _tax, uint256 creamBal); and matches the published mainnet contract.

RISK FACTORS
ISSUES

ADDITIONAL RISKS

5% of the minted ICS tokens go to the dev fund.

5% of ICS harvests are taxed and go to a tax address. This could be raised up to 10% of ICS harvests.

In the documentation (https://ice-cream-swap.gitbook.io/icecreamswap/roadmap/tokens-distributions), it is mentioned that taxed funds are burned. The taxed funds could be used for purposes other than burning if the destination address is an account that can arbitrarily initiate token transfers.

There is a lack of any test coverage for any of the smart contracts provided.

RECOMMENDATIONS

It is recommended to send the tokens to a burn address. This will ensure that the funds will definitely be burned.

As this is a fork of Pancake Swap's code, which already has some test cases (https://github.com/pancakeswap/pancake-farm/tree/master/test), it is recommended to build on top of the existing tests and add test coverage, especially for custom code changes (e.g. tax feature).