IceCreamSwap

Security Code Review

https://twitter.com/VidarTheAuditor - 10 February 2021

Overview

Project Summary

Project Name	icecreamswap
Description	Clone of pancakeswap
Platform	**Binance Smart Chain**, Solidity
Contracts	https://github.com/IceCreamSwap/contracts commit 7e433aa1d2633665b95a12687a17fc84d2a9c1ac
	• **CreamToken Contract 0x58f651DDE51CAa87c4111B16ee0A6Fab061Ee564** • **MilkShakeContract 0x8Cf93F2b41bA17F9189Aa7a86576f2764A442eca** • **SousChefContract 0x73C522A54941a2222c01C1032c5ABD225D3A132E** • **MasterChefContract 0x78Bd56CA4D781d1Be3808a7AF0A8b5446048c1AC**

Executive Summary

Binance Smart Chain contracts were provided.

We have checked the codebase and deployed contracts against the prototypes (Uniswap/Pancake). We also have run manual checks and tests.

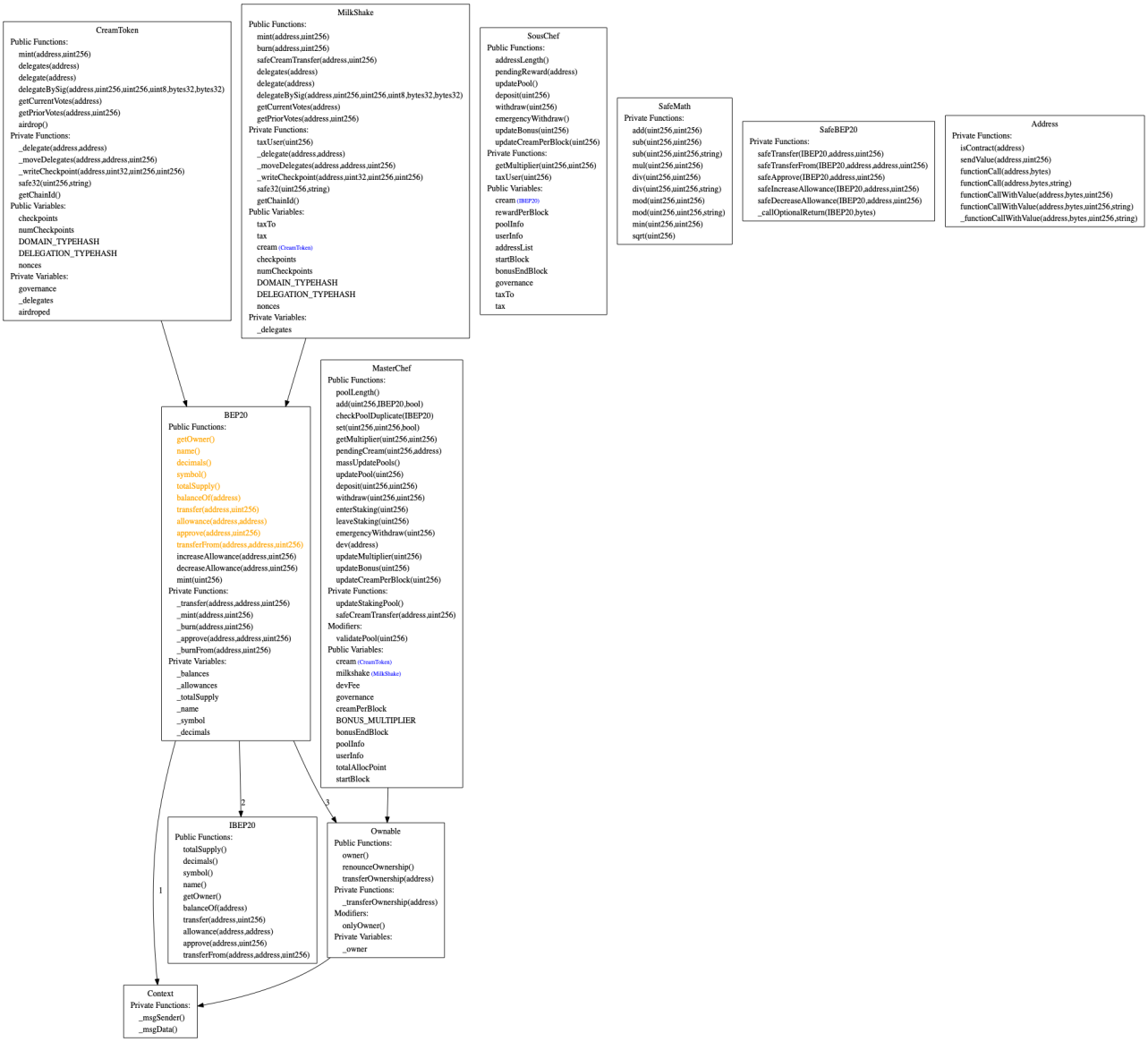
There is no high level issues with the currently deployed contracts.

Some recommendations where issued in Deployment section.

Disclaimer: The analysis did not include any tokenomics analysis (e.g. APY rates etc).

Architecture & Standards

Please find below the calling architecture of the reviewed contracts.



CreamToken and MilkShake are fully BEP20 compatible.

```
# Check CreamToken

## Check functions
[✓] totalSupply() is present
        [✓] totalSupply() -> () (correct return value)
        [✓] totalSupply() is view
[✓] balanceOf(address) is present
        [✓] balanceOf(address) -> () (correct return value)
        [✓] balanceOf(address) is view
[✓] transfer(address,uint256) is present
        [✓] transfer(address,uint256) -> () (correct return value)
        [✓] Transfer(address,address,uint256) is emitted
[✓] transferFrom(address,address,uint256) is present
        [✓] transferFrom(address,address,uint256) -> () (correct return value)
        [✓] Transfer(address,address,uint256) is emitted
[✓] approve(address,uint256) is present
        [✓] approve(address,uint256) -> () (correct return value)
        [✓] Approval(address,address,uint256) is emitted
[✓] allowance(address,address) is present
        [✓] allowance(address,address) -> () (correct return value)
        [✓] allowance(address,address) is view
[✓] name() is present
        [✓] name() -> () (correct return value)
        [✓] name() is view
[✓] symbol() is present
        [✓] symbol() -> () (correct return value)
        [✓] symbol() is view
[✓] decimals() is present
        [✓] decimals() -> () (correct return value)
        [✓] decimals() is view

## Check events
[✓] Transfer(address,address,uint256) is present
        [✓] parameter 0 is indexed
        [✓] parameter 1 is indexed
[✓] Approval(address,address,uint256) is present
        [✓] parameter 0 is indexed
        [✓] parameter 1 is indexed


        [✓] CreamToken has increaseAllowance(address,uint256)
```

```
# Check MilkShake

## Check functions
[✓] totalSupply() is present
        [✓] totalSupply() -> () (correct return value)
        [✓] totalSupply() is view
[✓] balanceOf(address) is present
        [✓] balanceOf(address) -> () (correct return value)
        [✓] balanceOf(address) is view
[✓] transfer(address,uint256) is present
        [✓] transfer(address,uint256) -> () (correct return value)
        [✓] Transfer(address,address,uint256) is emitted
[✓] transferFrom(address,address,uint256) is present
        [✓] transferFrom(address,address,uint256) -> () (correct return value)
        [✓] Transfer(address,address,uint256) is emitted
[✓] approve(address,uint256) is present
        [✓] approve(address,uint256) -> () (correct return value)
        [✓] Approval(address,address,uint256) is emitted
[✓] allowance(address,address) is present
        [✓] allowance(address,address) -> () (correct return value)
        [✓] allowance(address,address) is view
[✓] name() is present
        [✓] name() -> () (correct return value)
        [✓] name() is view
[✓] symbol() is present
        [✓] symbol() -> () (correct return value)
        [✓] symbol() is view
[✓] decimals() is present
        [✓] decimals() -> () (correct return value)
        [✓] decimals() is view

## Check events
[✓] Transfer(address,address,uint256) is present
        [✓] parameter 0 is indexed
        [✓] parameter 1 is indexed
[✓] Approval(address,address,uint256) is present
        [✓] parameter 0 is indexed
        [✓] parameter 1 is indexed


        [✓] MilkShake has increaseAllowance(address,uint256)
```

Findings

Number of contracts: 10+11+14 (including inherited ones)

Use: SafeMath

PancakeSwap Cloned Contracts:

Name	# functions	ERCS	ERC20 info	Complex code	Features
PancakeFactory	15			No	Tokens interaction Assembly
PancakePair	62	ERC20	∞ Minting Approve Race Cond.	No	Ecrecover Tokens interaction Assembly
IERC20	9	ERC20	No Minting Approve Race Cond.	No	
IPancakeCallee	1			No	
Math	2			No	
SafeMath	3			No	
UQ112x112	3			No	

Name	# functions	ERCS	ERC20 info	Complex code	Features
PancakeMigrator	4			No	Receive ETH Send ETH Tokens interaction
PancakeRouter	51			No	Receive ETH Send ETH Tokens interaction
PancakeRouter01	40			No	Receive ETH Send ETH Tokens interaction
IERC20	9	ERC20	No Minting Approve Race Cond.	No	
IPancakeFactory	8			No	
IPancakePair	27	ERC20	∞ Minting Approve Race Cond.	No	
IWETH	3			No	Receive ETH
IUniswapV1Exchange	5			No	Receive ETH
IUniswapV1Factory	1			No	
PancakeLibrary	8			No	Tokens interaction
SafeMath	3			No	

Name	# functions	ERCS	ERC20 info	Complex code	Features
IWBNB	3			No	Receive ETH
BnbStaking	24			No	Receive ETH Send ETH
CakeToken	52	ERC20	∞ Minting Approve Race Cond.	Yes	Tokens interaction Ecrecover Assembly
LotteryRewardPool	15			No	Tokens interaction
MasterChef	29			No	Tokens interaction
Timelock	10			No	Receive ETH Send ETH
Migrations	2			No	
Multicall	8			No	AbiEncoderV2
PancakeVoterProxy	7			No	Tokens interaction Proxy
WBNB	8	ERC20	No Minting Approve Race Cond.	No	Receive ETH Send ETH

IceCreamSwap Contracts

Name	# functions	ERCS	ERC20 info	Complex code	Features
CreamToken	52	ERC20	∞ Minting Approve Race Cond.	Yes	Ecrecover Assembly
MasterChef	29			No	Tokens interaction
MilkShake	53	ERC20	∞ Minting Approve Race Cond.	Yes	Ecrecover Tokens interaction Assembly
SousChef	11			No	Send ETH Tokens interaction
SafeMath	10			No	
SafeBEP20	6			No	Send ETH Tokens interaction
Address	7			No	Send ETH Assembly

Static Analysis Findings

High issues: None

Medium issues:

Dangerous strict equality:

```
BnbStaking.updatePool(uint256) (contracts/farm-contracts/BnbStaking.sol#156-170) uses a dangerous strict equality:
        - lpSupply == 0 (contracts/farm-contracts/BnbStaking.sol#162)
MasterChef.updatePool(uint256) (contracts/farm-contracts/MasterChef.sol#198-214) uses a dangerous strict equality:
        - lpSupply == 0 (contracts/farm-contracts/MasterChef.sol#204)
SmartChef.updatePool(uint256) (contracts/farm-contracts/SmartChef.sol#142-156) uses a dangerous strict equality:
        - lpSupply == 0 (contracts/farm-contracts/SmartChef.sol#148)
SousChef.updatePool() (contracts/farm-contracts/SousChef.sol#130-144) uses a dangerous strict equality:
        - creamSupply == 0 (contracts/farm-contracts/SousChef.sol#135)
```

Use of strict equalities that can be easily manipulated by an attacker.

[Manual Check] As it checks only the totalSupply, which can not go below 0, that does not possesses any risks.

Manual Checks

Swap Contracts

The codebase is clone of UNISWAP codebase used for example in Pancake Swap.

```
contract UniswapV2ERC20 {
    using SafeMathUniswap for uint;

    string public constant name = 'iCream-LP';
    string public constant symbol = 'ICLP';
    uint8 public constant decimals = 18;
    uint  public totalSupply;
    mapping(address => uint) public balanceOf;
    mapping(address => mapping(address => uint)) public allowance;
```

```
contract UniswapV2ERC20 is IUniswapV2ERC20 {
    using SafeMath for uint;

    string public constant name = 'Uniswap V2';
    string public constant symbol = 'UNI-V2';
    uint8 public constant decimals = 18;
    uint  public totalSupply;
    mapping(address => uint) public balanceOf;
    mapping(address => mapping(address => uint)) public allowance;
```

The following changes have been identified:

- Exchange fees are distributed: 0.15% for liquidity providers, and 0.15% for the treasury

```
        if (rootK > rootKLast) {
            uint numerator = totalSupply.mul(rootK.sub(rootKLast));
            uint denominator = rootK.mul(15).add(rootKLast);
            uint liquidity = numerator / denominator;
```

```
        if (rootK > rootKLast) {
            uint numerator = totalSupply.mul(rootK.sub(rootKLast));
            uint denominator = rootK.mul(5).add(rootKLast);
            uint liquidity = numerator / denominator;
```

Farm Contracts

The codebase is clone of Pancake Swap Farm contracts.

```
// CreamToken with Governance.
contract CreamToken is BEP20('Ice Cream', 'iCream') {
    address private governance;
    constructor( address _governance ) public {

        governance = _governance;

        // mint 1 token to deployer to test pools
        mint(msg.sender, 1 ether);

        // mint 1k iCream to add in the initial liquidity pool
        mint(governance, 1000 ether);

    }
```

```
// CakeToken with Governance.
contract CakeToken is BEP20('PancakeSwap Token', 'Cake') {
```

The following changes has been identified:

- Governance has been added as a role
    - - updateMultiplier: allow governance to change the multiplier of the pool.
    - - updateBonus: allow governance to change bonus period of the pool.
    - - updateIceCreamPerBlock: allow governance to change the amount of IceCream tokens minted in each block as reward.
- Harvest fee is set to 10% - hardcoded

```
    constructor(CreamToken _cream , address _taxTo ) public {
        taxTo = _taxTo ;
        cream = _cream ;

        // %10!!! it's div/1000 bellow:
        tax = 100; // Defaults to 10%. 1=0.1%

    }
```

- There is an issue with Milkshake aka Syrup bug. It is possible to unstable iCream without burning Milkshake and emergencyWithdraw() has an issue with burning Milkshake that would prevent leaving iCream.
    - **The team has confirmed that they are not planning to use Milkshake tokens and they advice users to avoid emergencyWithdraw() - https://icecreamswap.medium.com/important-notice-icecreamswap-transparency-report-and-new-parnership-e3332f402fde**
- SousChef: is not used

Deployment & Contract Ownership

The contracts are currently deployed on BSC Mainnet:

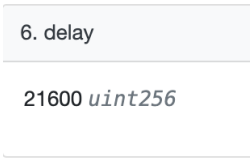
farm-contracts

- BnbStaking not used.
- CreamToken the iCream token. 0x58f651DDE51CAa87c4111B16ee0A6Fab061Ee564.
- LotteryRewardPool not used.
- MasterChef main pool contract. 0x78Bd56CA4D781d1Be3808a7AF0A8b5446048c1AC.
- MilkShake: MilkShake iCream Pool contract. 0x8Cf93F2b41bA17F9189Aa7a86576f2764A442eca.
- SmartChef: MilkShake BNB Pool contract.
- SousChef: not used. 0x73C522A54941a2222c01C1032c5ABD225D3A132E.
- Timelock: time lock used in MasterChef contract. 0x1140A764DFB67821dFa3f9C65B44818a2ce781D7.

swap-contracts

- UniswapV2Router02 router. 0x6728f3c8241C44Cc741C9553Ff7824ba9E932A4A.
- UniswapV2Pair ICLP pair.
- UniswapV2Factory factory. 0xc8c9aB92AB70E954aF23c49f98aaCc1f94EBEeD7.
- UniswapV2ERC20 erc20 uniswap pair.

The owner of MasterChef contract is Timelock contract (https://bscscan.com/address/0x1140a764dfb67821dfa3f9c65b44818a2ce781d7#code). Current delay is set to 6 hours.



[Recommendation] As the community of the project is located worldwide it is advisable to set delay to minimum 24h.

The liquidity of the iCream is not locked.

[Recommendation] Lock the whole iCream liquidity owned by the team.

Disclaimer

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. As one review is not enough to provide 100% security against any attacks or bugs, it is advisable to conduct more reviews or/and audits.

The report does not provide personalised investment advice or recommendations, especially does not provide advice to conclude 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.

The report should not be considered as an investment advice.