Audit Report

Bernard masterchef

January 2022

Type	BEP20
Network	BSC
Address	0xCC097123af369385e13e1dfea18cD334f1D8273E
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	ChefBernard
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/address/0xCC097123af369385e13e1dfea18cD334f1D8273E
Source	contract.sol

Audit Updates

Initial Audit	30th January 2022
Corrected	

Contract Analysis

Exceed Limit Fees Manipulation

Criticality	minor
Location	contract.sol#L133

Description

The contract owner has the authority to increase the fees. The owner may take advantage of it by calling the function *updateTaxedContract* with a high percentage value. The rewardRate amount accumulates fees that are transferred to the owner's wallet.

```
function updateTaxedContract(address _addr, bool flag, uint256 _taxeFee) public
onlyOwner {
    taxedContract[_addr] = flag;
    taxeFee = _taxeFee;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

The owner should carefully manage the credentials of the owner’s account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one.

NFT Levels Manipulation

Criticality	minor
Location	contract.sol#L133

Description

The contract owner has the authority to change the level of an NFT. The owner may take advantage of it by calling the function *updateNftInfo* with a high percentage value. The NFT level is used in order to allow a user to claim his rewards.

```
function updateNftInfo(address _nftAddress, uint _tokenId, uint8 _newlevel)
public {
    require(nftManager == msg.sender, "nftManager: not permitted");
    require(_newlevel > 0, "nftManager: invalid level");
    require(_newlevel < 4, "nftManager: invalid level");
    require(nftLevel[_nftAddress][_tokenId] != _newlevel, "nftManager: invalid
level");
    nftLevel[_nftAddress][_tokenId] = _newlevel;
}
```

```
if( user.userLevel < pool.poolLevel) {
    pending = 0;
}

if(pending > 0) {
    safeBONESTransfer(msg.sender, pending);
}
```

Recommendation

The contract could embody a check for not allowing mutating the NFT level, when the holder of the NFT is participating in a pool.

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L1304,L1299,L1294 and 22 more

Description

Public functions that are never called by the contract should be declared external to save gas.

```
withdrawBEP20
updateTaxedContract
updateEmissionRate
...
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1052,L1050,L1048 and 36 more

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
BONUS_MULTIPLIER
BONESPerBlock
BONES
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L41,L51,L76 and 17 more

Description

Functions that are not used in the contract, and make the code's size bigger.

```
trySub
tryMul
tryMod
...
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1294,L1118,L1101 and 1 more

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
BONESPerBlock = _BONESPerBlock
totalAllocPoint = totalAllocPoint.sub(prevAllocPoint).add(_allocPoint)
totalAllocPoint = totalAllocPoint.add(_allocPoint)
...
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Address	Library			
	isContract	Internal		

	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
SafeBEP20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
BEP20	Implementation	Context, IBEP20, Ownable		
	<Constructor>	Public	✓	-
	getOwner	External		-
	name	Public		-
	symbol	Public		-

	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	mint	Public	✓	onlyOwner
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_burnFrom	Internal	✓	
IMigratorChef	Interface			
	migrate	External	✓	-
IERC165	Interface			
	supportsInterface	External		-
IERC721	Interface	IERC165		
	totalSupply	External		-
	balanceOf	External		-
	ownerOf	External		-
	tokenOfOwnerByIndex	External		-
ChefBernard	Implementation	Ownable		
	<Constructor>	Public	✓	-
	updateMultiplier	Public	✓	onlyOwner
	poolLength	External		-
	add	Public	✓	onlyOwner
	set	Public	✓	onlyOwner
	getMultiplier	Public		-

	pendingBONES	External		-
	massUpdatePools	Public	✓	-
	updatePool	Public	✓	-
	deposit	Public	✓	-
	withdraw	Public	✓	-
	emergencyWithdraw	Public	✓	-
	safeBONESTransfer	Internal	✓	
	addNftInfo	Public	✓	nonDuplicated
	updateNftInfo	Public	✓	-
	setNftManager	Public	✓	-
	getUserLevel	Public		-
	updateEmissionRate	Public	✓	onlyOwner
	updateTaxedContract	Public	✓	onlyOwner
	withdrawBEP20	Public	✓	onlyOwner

Contract Flow

Summary

Bernard masterchef is a stacking application. Users can deposit lp tokens in order to get BONES tokens as a reward. The award is proportional to the time period that has elapsed and the deposited amount. Users can claim their rewards any time. There are pools that require the users to hold NFTs with specific “level”. The level of the NFT is defined by the admins. There are some functions that can be abused by the contract owner. We state that the owner privileges are necessary and required for proper operation of the betting application. Thus, we emphasise the contract owner to be extra careful with the credentials.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Coinscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Coinscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Coinscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Coinscope team disclaims any liability for the resulting losses.

About Coinscope

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.

The Coinscope.co team

https://www.coinscope.co