Cyberscope

Audit Report

The ClubHouse

Staking Tier 2

August 2022

Type	BEP20
Network	BSC
Address	0xC92345c6D7eF5F0cB0859ae5d68E2DD5f84775b5
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Tier2_TCHStaking
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0xC92345c6D7eF5F0cB0859ae5d68E2DD5f84775b5
Domain	

Source Files

Filename	**SHA256**
contract.sol	cefadd7af377018d8e73907124e59625e75a09d1b5774ea92981f33558480e67

Audit Updates

Initial Audit	20th August 2022
Corrected	

Contract Diagnostics

🔴 Critical 🟠 Medium ⚪ Minor

Severity	Code	Description	Status
🔴	URUF	Users Receive Unlimited Funds	Unresolved
🟠	MAL	Diversified State Between Variables	Unresolved
⚪	DSM	Data Structure Misuse	Unresolved
⚪	OWCB	Owner Withdraws Contract Balance	Unresolved
⚪	L01	Public Function could be Declared External	Unresolved
⚪	L02	State Variables could be Declared Constant	Unresolved
⚪	L03	Redundant Statements	Unresolved
⚪	L04	Conformance to Solidity Naming Conventions	Unresolved
⚪	L07	Missing Events Arithmetic	Unresolved
⚪	L09	Dead Code Elimination	Unresolved

URUF - Users Receive Unlimited Funds

Criticality	critical
Location	contract.sol#L783
Status	Unresolved

Description

The users have the ability to execute the deposit method with zero amount and zero stake time. If a user has already staked, then he receives the “pending” reward. The contract is not keeping track that the “pending” reward has been transferred to the user. As a result, every time that the user executes the deposit method, he will receive rewards until the contract’s balance decreases to zero.

```
function _deposit(uint256 _amount, uint _stakeUntil) internal {

    PoolInfo storage pool = poolInfo[0];
    UserInfo storage user = userInfo[0][msg.sender];
    updatePool(0);

    if ( _stakeUntil != 0) {
        //deposit and relock case
        if(user.stakeUntil>0)require(_stakeUntil >= user.stakeUntil, "Not
possible to shorten the lock.");
        user.stakeUntil = _stakeUntil;
    }
```

Recommendation

The contract could update the rewardDebt variable so that the users will not receive the same rewards.

DSBV - Diversified State Between Variables

Criticality	medium
Location	contract.sol#L795
Status	Unresolved

Description

The `safeTCHTransfer` transfers the contract's balance instead of the expected amount if the amount is greater than the contract's balance. This flow produces two issues:

1. The caller of `safeTCHTransfer` is not aware of this. As a result, it assumes that the entire amount has been transferred and it updates the corresponding variables. Hence, the contract's variables keep a different state compared to reality.

2. The users will not receive the expected amount.

```
uint256 pending =
user.amount.mul(pool.accTokenPerShare).div(1e12).sub(user.rewardDebt);
if(pending > 0) {
    safeTCHTransfer(msg.sender, pending);
}
fundedBalance = fundedBalance.sub(pending);
```

Recommendation

The `safeTCHTransfer` should notify the caller about the actual amount that has been transferred.

DSM - Data Structure Misuse

Criticality	minor
Location	contract.sol#L679
Status	Unresolved

Description

The `userInfo` is defined as a mapping but it uses a singleton structure. The `poolInfo` is defined as an array but it uses a singleton structure.

```
mapping (uint256 => mapping (address => UserInfo)) public userInfo;
PoolInfo[] public poolInfo;
```

Recommendation

The contract could remove the mapping and array structure since it is redundant.

OWCB - Owner Withdraws Contract Balance

Criticality	minor
Location	contract.sol#L739
Status	Unresolved

Description

The contract owner has the authority to withdraw the funds that are indented to operate as the staking rewards. As a result, the users will not be able to unstake.

```
function withdrawTeam(uint256 _amount) public onlyOwner{
    require(_amount<=fundedBalance, 'Not enough tokens.');
    IBEP20(tchToken).safeTransfer(address(msg.sender), _amount);
    fundedBalance = fundedBalance.sub(_amount);
}
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L587,606,615,729,734,739,768,773,778,816,865,870
Status	Unresolved

Description

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

```
owner
renounceOwnership
transferOwnership
setTokenPerBlock
depositTeam
withdrawTeam
deposit
reDeposit
reLock
...
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L704,705,701,707,706
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
minimumLockPeriod
poolLimit
tchToken
userHighLimit
userLowLimit
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor
Location	contract.sol#L546
Status	Unresolved

Description

The contract contains statements that are not used and have no effect. As a result, those segments increase the code size of the contract unnecessarily.

```
Context
```

Recommendation

Remove the redundant statements in order to decrease the code size.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L669,729,734,739,746,751,768,773,778,816,849,874
Status	Unresolved

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.

```
Tier2_TCHStaking
_tokenPerBlock
_amount
_from
_to
_pid
_stakeUntil
_user
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L729,734,739
Status	Unresolved

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.

```
tokenPerBlock = _tokenPerBlock
fundedBalance = fundedBalance.add(_amount * (10 ** 9))
fundedBalance = fundedBalance.sub(_amount)
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L360,389,403,334,478,503,494,171,176,631,662,651
Status	Unresolved

Description

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

```
functionCall
functionCallWithValue
sendValue
safeApprove
safeDecreaseAllowance
safeIncreaseAllowance
min
sqrt
safeTransferBNB
...
```

Recommendation

Remove unused functions.

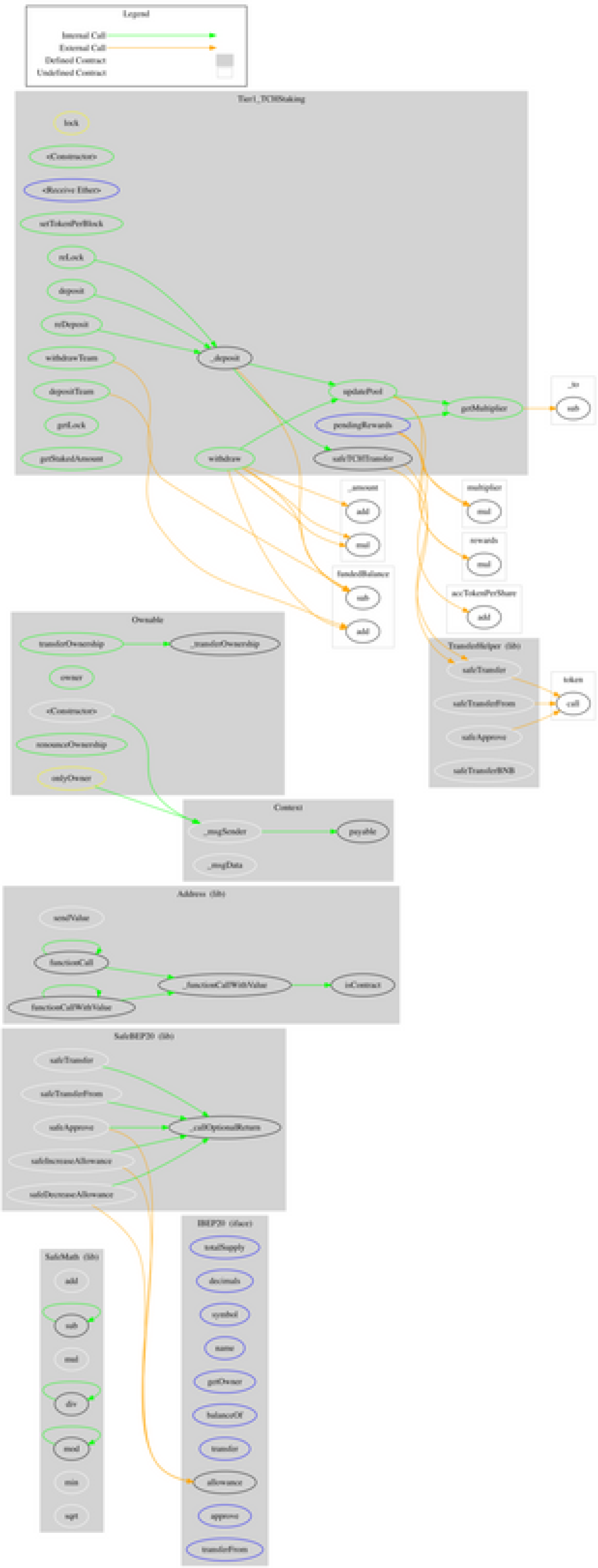
Contract Functions

Contract	**Type**	**Bases**		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
	min	Internal		
	sqrt	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	✓	
	_functionCallWithValue	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
	_transferOwnership	Internal	✓	
TransferHelper	Library			
	safeApprove	Internal	✓	
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeTransferBNB	Internal	✓	
Tier2_TCHStaking	Implementation	Ownable		
	<Constructor>	Public	✓	-
	<Receive Ether>	External	Payable	-
	setTokenPerBlock	Public	✓	onlyOwner
	depositTeam	Public	✓	onlyOwner
	withdrawTeam	Public	✓	onlyOwner

	getMultiplier	Public		-
	updatePool	Public	✓	-
	deposit	Public	✓	lock
	reDeposit	Public	✓	lock
	reLock	Public	✓	lock
	_deposit	Internal	✓	
	withdraw	Public	✓	lock
	pendingRewards	External		-
	getLock	Public		-
	getStakedAmount	Public		-
	safeTCHTransfer	Internal	✓	

Contract Flow



Summary

The ClubHouse Staking Tier 2 implements a staking functionality. This audit focuses on potential vulnerabilities, business logic concerns and improvements.

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.

Cyberscope 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 Cyberscope 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 Cyberscope 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 Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

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.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io