



Cyberscope

Audit Report

Ecoverse Stake

March 2022

Type BEP20

Network BSC

Address 0x64070776D926C08dAE9197dD2886128a344eCDad

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Audit Updates	3
Contract Analysis	4
Contract Owner Privileges	5
Contract Diagnostics	6
OCTD - Owner Contract Tokens Drain	7
Description	7
Recommendation	7
CR - Code Repetition	8
Description	8
Recommendation	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L04 - Conformance to Solidity Naming Conventions	10
Description	10
Recommendation	10
L07 - Missing Events Arithmetic	11
Description	11
Recommendation	11
L09 - Dead Code Elimination	12
Description	12
Recommendation	12
Contract Functions	13
Contract Flow	16

Summary	17
Disclaimer	18
About Cyberscope	19

Contract Review

Contract Name	StakeEarn
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/address/0x64070776D926C08dAE9197dD2886128a344eCDad
Source	contract.sol

Audit Updates

Initial Audit	18th March 2022
Corrected	

Contract Analysis

The contract implements a typical staking functionality. The users have the ability to deposit tokens in order to withdraw their amount accumulated with interest in the future. The deposited amount cannot be withdrawn if the vesting period has not elapsed. The Ecoverse stake contract defines the staking period and the interest amount (reward).

Contract Owner Privileges

- The contract owner has the authority to pause the deposits. When the deposits are paused, the users can withdraw their rewards.
- The contract owner has the authority to change the staking duration. The change does not affect the current depositors.
- The contract owner has the authority to change the interest (reward amount). The change does not affect the current depositors.
- The contract owner has the authority to change the minimum allowed amount for staking.
- The contract owner has the authority to withdraw all the accumulated amount of the depositors.

Note

The staking distribution is based on the fact that the contract owner should keep the contract with a sufficient amount of tokens to cover the total withdrawal amount.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	OCTD	Owner Contract Tokens Drain
●	CR	Code Repetition
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L366

Description

The contract owner has the authority to drain all the tokens from the contract. This amount has been accumulated from the user's deposits. The contract owner has the ability to do that by setting the stakedToken address to the recoverWrongTokens function.

```
function recoverWrongTokens(address tokenAddress, uint256 tokenAmount) external  
onlyOwner {  
    IBEP20(tokenAddress).safeTransfer(address(msg.sender), tokenAmount);  
    emit AdminTokenRecovery(tokenAddress, tokenAmount);  
}
```

Recommendation

The contract could prevent the withdrawal of the stakedToken address, if the amount of tokens is more than the totalStaked + totalReward.

CR - Code Repetition

Criticality	minor
Location	contract.sol#L338

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

```
require(!paused, "Staking is not available right now");
```

Recommendation

The contract implements the ownership feature with two modifiers, the `whenNotPaused` and `whenPaused`. The contract could reuse these functions instead of directly accessing the pause variable.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L36,41,351,371,376,381,386

Description

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

```
getUserStaking  
getUserStats  
unpause  
pause  
withdraw  
transferOwnership  
renounceOwnership
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

contract.sol#L391,395,399,282,283

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  
Duration  
SetMinStakeAmount  
SetStakeBonus  
SetStakeDuration
```

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#L391,395,399

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.

```
minStake = minStakeAmount  
Bonus = bonus  
Duration = duration
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L183,191,212,216,202,206,177,254,264,259

Description

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

```
safeIncreaseAllowance
safeDecreaseAllowance
safeApprove
sendValue
functionStaticCall
functionDelegateCall
functionCallWithValue
functionCall
...
```

Recommendation

Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
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		
ReentrancyGuard	Implementation			
	<Constructor>	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	✓	
StakeEarn	Implementation	Ownable, Reentrancy Guard		

	<Constructor>	Public	✓	-
	deposit	External	✓	nonReentrant
	withdraw	Public	✓	-
	recoverWrongTokens	External	✓	onlyOwner
	pause	Public	✓	onlyOwner whenNotPaused
	unpause	Public	✓	onlyOwner whenPaused
	getUserStats	Public		-
	getUserStaking	Public		-
	SetStakeDuration	External	✓	onlyOwner
	SetStakeBonus	External	✓	onlyOwner
	SetMinStakeAmount	External	✓	onlyOwner

Contract Flow



Summary

Ecoverse Stake implements a typical staking functionality. The users have the ability to deposit tokens in order to withdraw the interest after a period of time. In this audit we focus on the contract owner's privileges, we mention some issues that could be produced, and we recommend some optimizations.

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 provides all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>