Cyberscope

Audit Report

WakandaPoolInitializable

September 2022

Type	BEP20
Network	BSC
Address	0x6141490c9540BbbF17f3eC79990Bbf9da30Ba57F
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	WakandaPoolInitializable
Compiler Version	v0.6.12+commit.27d51765
Optimization	99999 runs
Explorer	https://bscscan.com/token/0x6141490c9540BbbF17f3eC79990Bbf9da30Ba57F
Domain	-

Audit Updates

Initial Audit	21st September 2022
Corrected	

Source Files

Filename	SHA256
contracts/access/Ownable.sol	b9f957b42bdcf3d3499be4c94558152e91658e34a1fe5a5e8f0972ce20e15ed7
contracts/math/SafeMath.sol	f5421275a1e2b5a0dd0592a4ce1a09cefc339fceb42e822e0f711a5c66933ae8
contracts/utils/Address.sol	11ad5e3e21434e00c4ceba1f5a977b7a68bdd7d16b849276ce4ff4495129eec7
contracts/utils/Context.sol	9a3d1e5be0f0ace13e2d9aa1d0a1c3a6574983983ad5de94fc412f878bf7fe89
contracts/utils/ReentrancyGuard.sol	3fc7968f4a1937caf3c96dffbac350398f86faad96288502e02c3a2b9f245e39
src/farm/GenericStake.sol	06477467676480b4dd899e462bdd869f01d76a732d2c2524edbfa8b3d7bcf0c8
src/helpers/IBEP20.sol	5f8366fc3b9a5a8e25a639f2cf8534b5e017ffdce91c597dd7668e557c2fe272
src/helpers/SafeBEP20.sol	aba52f299dbdf14cfd80413ea56012290bfee253c99a09d5e9bc2bf09246d42e

Introduction

The WakandaPoolInitializable contract implements the WKD token pool.

The owner of the pool can:

- Withdraw all the rewards.
- Withdraw non relevant tokens from the contract.
- Stop the pool.
- Update the pool limit per user. This may happen if the pool has not been initialized with the variable hasUserLimit.
- Update reward per block before the pool launch.
- Update when the pool is launching and ending before the launch.

The users can:

- Join the pool by depositing staked tokens to the pool.
- Withdraw staked tokens and collect reward tokens if they have any available.
- Withdraw all the staked tokens without taking into consideration the rewards.
- Users can view pending rewards.

*Note: THis audit assumes that the *safeTransfer* and *safeTransferFrom* functions will transfer all the amount and revert in case of failure.

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	OCTD	Transfers Contract's Tokens	Unresolved
	BLC	Business Logic Concern	Unresolved
	CR	Code Repetition	Unresolved
	L01	Public Function could be Declared External	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L07	Missing Events Arithmetic	Unresolved

OCTD - Transfers Contract's Tokens

Criticality	minor / informative
Location	contract.sol#L1
Status	Unresolved

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `claimTokens` function.

```
function claimTokens() public onlyOwner {
  payable(_owner).transfer(address(this).balance);
}
```

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. That risk can be prevented by temporarily locking the contract or renouncing ownership.

BLC - Business Logic Concern

Criticality	critical
Location	contract.sol#L
Status	Unresolved

Description

The business logic seems peculiar. The implementation may not follow the expected behavior.

On emergency reward the contract has to update the stakedByUsers otherwise, it breaks the contract's balance.

```
function emergencyRewardWithdraw(uint256 _amount) external onlyOwner {
        rewardToken.safeTransfer(address(msg.sender), _amount);
    }
```

We assume that these function will transfer all the amount and revert in case of failure

1. safeTransfer
2. safeTransferFrom

This may underflow if the stakedToken has fees on transfers.

```
function availableRewards() public view returns (uint) {
        return stakedToken.balanceOf(address(this)) - stakedByUsers;
}
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

CR - Code Repetition

Criticality	minor / informative
Location	contract.sol#L334,368,140,184
Status	Unresolved

Description

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

```
uint256 multiplier = _getMultiplier(lastRewardBlock, block.number);
uint256 wakandaReward = multiplier.mul(rewardPerBlock);
uint256 adjustedTokenPerShare = accTokenPerShare.add(
    wakandaReward.mul(PRECISION_FACTOR).div(stakedTokenSupply)
);
```

```
        if (user.amount > 0) {
            uint256 pending = user
                .amount
                .mul(accTokenPerShare)
                .div(PRECISION_FACTOR)
                .sub(user.rewardDebt);
            if (pending > 0) {
                rewardToken.safeTransfer(address(msg.sender), pending);
            }
        }

        if (_amount > 0) {
            user.amount = user.amount.add(_amount);
            stakedToken.safeTransferFrom(
                address(msg.sender),
                address(this),
                _amount
            );
        }
        user.rewardDebt = user.amount.mul(accTokenPerShare).div(
            PRECISION_FACTOR
        );
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract/GenericStake.sol#L200
Status	Unresolved

Description

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

```
availableRewards
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract/GenericStake.sol#L86,127,91,290,235,90,87,304,89,88,330,225,171,268,18,269,46,303,92
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.

```
_stakedToken
_amount
_poolLimitPerUser
_rewardPerBlock
_tokenAddress
_bonusEndBlock
_rewardToken
_tokenAmount
_startBlock
...
```

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 / informative
Location	contract/GenericStake.sol#L85
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.

```
rewardPerBlock = _rewardPerBlock
```

Recommendation

Emit an event for critical parameter changes.

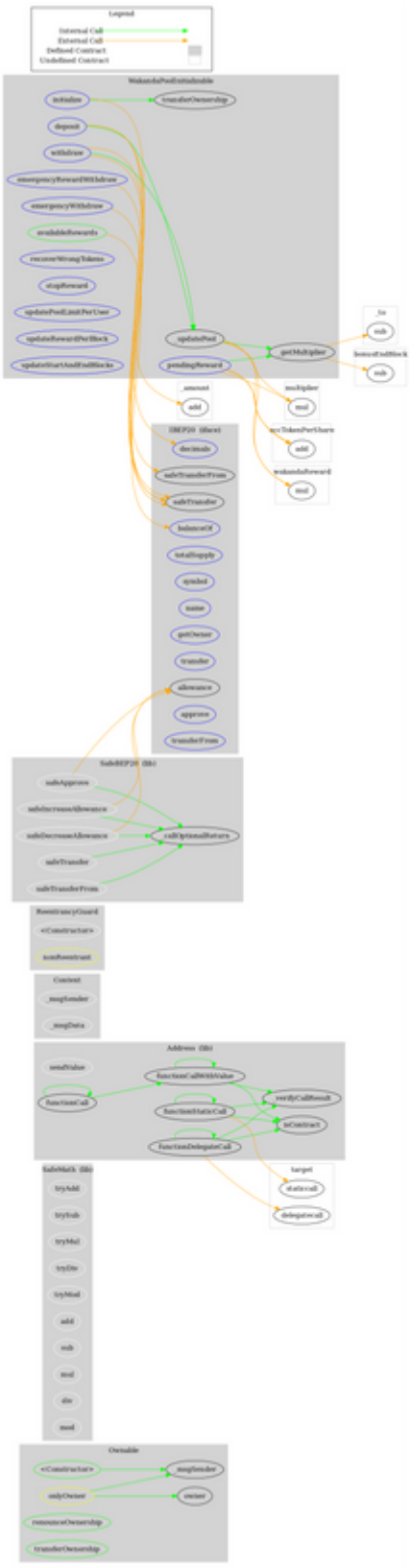
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
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		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ReentrancyGuard	Implementation			
	<Constructor>	Internal	✓	
WakandaPoolInitializable	Implementation	Ownable, ReentrancyGuard		
	<Constructor>	Public	✓	-
	initialize	External	✓	-
	deposit	External	✓	nonReentrant
	withdraw	External	✓	nonReentrant
	availableRewards	Public		-
	emergencyWithdraw	External	✓	nonReentrant
	emergencyRewardWithdraw	External	✓	onlyOwner
	recoverWrongTokens	External	✓	onlyOwner
	stopReward	External	✓	onlyOwner
	updatePoolLimitPerUser	External	✓	onlyOwner
	updateRewardPerBlock	External	✓	onlyOwner
	updateStartAndEndBlocks	External	✓	onlyOwner
	pendingReward	External		-
	_updatePool	Internal	✓	
	_getMultiplier	Internal		
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-

	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeBEP20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	

Contract Flow



Summary

The Smart Contract analysis reported no compiler issues. The contract owner has the authority to drain the contract's tokens. Other than that, the contract owner can access some admin functions that can not be used maliciously to disturb the users’ transactions. This audit investigates the security aspects and mentions some potential 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