Cyberscope

Audit Report

WakandaPoolInitializable

September 2022

Github	https://github.com/Bloceducare/WakandaInu-Contracts/tree/main/src
Commit	627de2245bb43b0de4ce357b33b38be7fc3a130e
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	GenericStakeFactory
Compiler Version	v0.6.12+commit.27d51765
Testing Deploy	https://testnet.bscscan.com/token/0x4c66e786FfDc6B59A75316843804D84D02edf2A9
Domain	https://wakandainu.com

Audit Updates

Initial Audit	21st September 2022 https://github.com/cyberscope-io/audits/blob/main/wkd/wakandaPoolInitializable.pdf
Corrected	26th September 2022

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	b9f957b42bdcf3d3499be4c94558152e91658e34a1fe5a5e8f0972ce20e15ed7
@openzeppelin/contracts/math/SafeMath.sol	4a04d0a20a19e3ef1dcabae9cad9ba006430a4e7eec4d9b519db87999722c98a
@openzeppelin/contracts/utils/Address.sol	11ad5e3e21434e00c4ceba1f5a977b7a68bdd7d16b849276ce4ff4495129eec7
@openzeppelin/contracts/utils/Context.sol	9a3d1e5be0f0ace13e2d9aa1d0a1c3a6574983983ad5de94fc412f878bf7fe89
@openzeppelin/contracts/utils/ReentrancyGuard.sol	3fc7968f4a1937caf3c96dffbac350398f86faad96288502e02c3a2b9f245e39
contracts/farm/GenericStake.sol	ea2eea8000881188a76a3fd8285f836e210ff8335d7d58ac434f5bedcd098baf
contracts/farm/GenericStakeFactory.sol	01e8e25be68e095a3a0b2f6bb5beaaaf4ec14a1d4c3b9d2d0207cdc7cb1f1e77
contracts/helpers/IBEP20.sol	5f8366fc3b9a5a8e25a639f2cf8534b5e017ffdce91c597dd7668e557c2fe272
contracts/helpers/SafeBEP20.sol	fa16115d3837e0e87ec528b29a4fbc0ee0bb3078ac075d06dd7cbfa4864acdf0

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
●	CR	Code Repetition	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved

CR - Code Repetition

Criticality	minor / informative
Location	contract.sol#L332,366,138,182
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.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/farm/GenericStake.sol#L86,127,91,288,237,90,87,302,89,88,328,226,171,266,18,267,46,301,92 contracts/farm/GenericStakeFactory.sol#L34,33,29,35,32,30,31
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
_bonusEndBlock
_tokenAddress
_rewardToken
_admin
_tokenAmount
...
```

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	contracts/farm/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.

L08 - Tautology or Contradiction

Criticality	minor / informative
Location	contracts/farm/GenericStakeFactory.sol#L28
Status	Unresolved

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool)(_stakedToken.totalSupply() >= 0)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

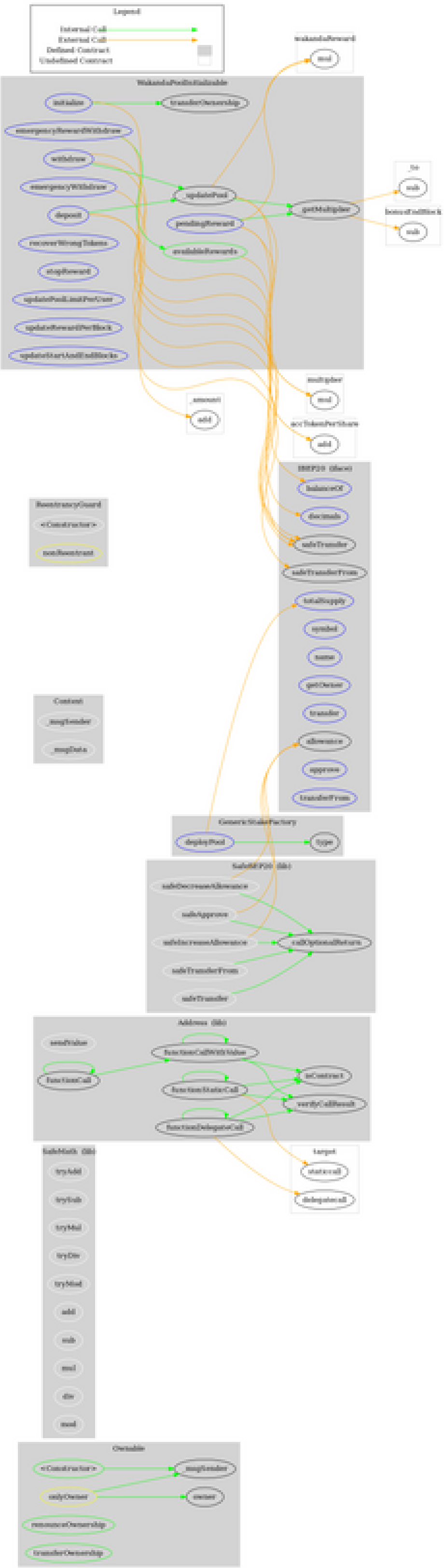
Contract Functions

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		
GenericStakeFactory	Implementation	Ownable		
	<Constructor>	Public	✓	-

	deployPool	External	✓	onlyOwner
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



Domain Info

Domain Name	wakandainu.com
Registry Domain ID	2650366346_DOMAIN_COM-VRSN
Creation Date	2021-10-26T11:48:53.00Z
Updated Date	2021-11-11T12:32:24.22Z
Registry Expiry Date	2026-10-26T11:48:53.00Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	http://www.namecheap.com
Registrar	NAMECHEAP INC
Registrar IANA ID	1068

The domain was created 11 months before the creation of the audit. It will expire in about 4 years.

There is no public billing information, the creator is protected by the privacy settings.

Summary

The Smart Contract analysis reported no compiler issues. 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