Cyberscope

Audit Report

Wakanda Launchpad

November 2022

Github	https://github.com/Bloceducare/WKDLaunchPad
Commit	d06ecf3db044c03d0af6d52ce946fad34debecde
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Launchpad
Compiler Version	v0.8.9+commit.e5eed63a
Testing Deploy	https://testnet.bscscan.com/token/0x9A17CF6099bD116BE0F9ad11FC0b06002C5619a3
Domain	https://wakandainu.com

Audit Updates

Initial Audit	8th November 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/utils/Address.sol	1e0922f6c0bf6b1b8b4d480dcabb691b1359195a297bde6dc5172e79f3a1f826
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
contracts/launchpad.sol	3a1e19723ab354603f77500c16c562f8551b3e7569e5bb53b403213bc5825fde
contracts/launchpadDeployer.sol	26e75e89ee28c9a76a68728af589d998c34b995177a582bdac8b2bd6e8b4cd16
contracts/utils/IBEP20.sol	dd74634844f948ee08aa6ff721d3cf2e8123d3b2206e78dd90d7c4f8ad4f024c
contracts/utils/SafeBEP20.sol	3ba408775d84a53acafaf05208ddc1c48e1a2c058a2bff7cb5dfa458bdd0dac5
contracts/WKDCommit.sol	007d9d87e4973de6454e75fc910d6ab1533832d091a15168705cb422e437d33a

Introduction

The Wakanda ecosystem implements a launchpad mechanism. The launchpad mechanism consists of three contracts. The launchpad, launchpadDeployer, and the WLDcommit contract.

- The launchpad contractimplements the core functionality of the launchpad mechanism.
- The LaunchpadDeployer contract implements a launchpad deployer mechanism.
- The WKDCommit contract implements a depositor contract. Where users can deposit and withdraw Wakanda tokens.

The launchpad has two user tiers for each launchpad contribution, Tier1 and Tie2. Each Tier offers a different percentage for claiming tokens. Tier percentages are initialized on contract creation.

Roles

Launchpad

The contract has an admin role. The admin has the authority to

- Recover any forgotten ERC20 tokens from the contract.
- Withdraw presale tokens.
- Finalize the presale procedure.

Users have the authority to

- Contribute to the presale during the presale period. The user's tier depends on the Wakanda tokens that he is holding.
- Claim the presale token once the presale period elapsed.

WKDCommit

The admin has the authority to recover any ERC20 tokens from the contract, except for the Wakanda token.

Users have the authority to deposit and withdraw Wakanda tokens.

LaunchpadDeployer

The contract has an owner role. The owner has the authority

- Create a new presale.
- Recover any forgotten ERC20 tokens from the contract.

Contract Diagnostics

🔴 Critical 🟠 Medium ● Minor / Informative

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
🔴	BLC	Business Logic Concern	Unresolved
●	CO	Code Optimization	Unresolved
●	MC	Missing Check	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol/launchpad.sol#L169,175,185,197,278
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
offeringToken.transfer(msg.sender, amount);

offeringToken.transfer(msg.sender, _offeringAmount);

offeringToken.transfer(msg.sender, offeringTokenAmount);

IBEP20(_tokenAddress).transfer(msg.sender, _tokenAmount);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

BLC - Business Logic Concern

Criticality	critical
Location	contract.sol/launchpad.sol#L102,153
Status	Unresolved

Description

The contract miscalculates the `launchPadInfo.tier2Amount`. The expression `_offeringAmount * (_tier2Percentage);` is not divided with the corresponding total percentage.

```
function initialize(
        address _offeringToken,
        uint256 _startBlock,
        uint256 _endBlock,
        address _adminAddress,
        address _projectOwner,
        address _wkdCommit,
        uint256 _offeringAmount,
        uint256 _raisingAmount,
        uint256 _launchPercentShare,
        uint256 _tier2Percentage,
        uint256 _minimumRequirementForTier2
    ) public {
        //..
        launchPadInfo.tier2Amount = _offeringAmount * (_tier2Percentage);
        launchPadInfo.tier1Amount = (_offeringAmount * (100 - _tier2Percentage)) / 100;
```

The contract accumulates the deposited funds twice to the user's amountDeposited.

```
user[msg.sender].amountDeposited += msg.value;
participants.push(msg.sender);
user[msg.sender].amountDeposited = user[msg.sender].amountDeposited +
msg.value;
```

Recommendation

- The launchPadInfo.tier1Amount should be divided by 100.

- The amountDeposited should be accumulated one.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol/launchpad.sol#L9,40,156
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

Redundant variable on the data structure.

```
contract Launchpad is Ownable {
    IBEP20 offeringToken;
    /..
    struct LaunchpadDetails {
        // offerinn token
        address offeringToken;
```

Redundant code statement.

```
function deposit() public payable {
        //..
        raisedAmount = raisedAmount += msg.value;
       //..
    }
```

Recommendation

Rewrite some code segments so the runtime will be more performant.

The contract could remove one offeringToken variable.

The contract could remove the first raisedAmount assignment.

MC - Missing Check

Criticality	minor / informative
Location	contract.sol/launchpad.sol#L102 contract.sol/WKDcommit.sol#L32
Status	Unresolved

Description

The contract is processing variables that have not been properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

The constructor arguments have not been properly sanitized.

```
function initialize(
        address _offeringToken,
        uint256 _startBlock,
        uint256 _endBlock,
        address _adminAddress,
        address _projectOwner,
        address _wkdCommit,
        uint256 _offeringAmount,
        uint256 _raisingAmount,
        uint256 _launchPercentShare,
        uint256 _tier2Percentage,
        uint256 _minimumRequirementForTier2
    ) public {
        if (msg.sender != owner()) revert NotPermitted();
        if (isInitialized) revert NotInitialized();
        if (_launchPercentShare > 100) revert InvalidPercentage();
        if (_tier2Percentage > 100) revert InvalidPercentage();
```

The admin argument is not sanitized properly.

```
constructor(address _admin) {
        admin = _admin;
}
```

The variable _tier2Percentage is not properly sanitized.

```
function createLaunchpad(
        //...
        uint256 _tier2Percentage,
        uint256 _minimumRequirementForTier2
    )
```

Recommendation

The contract should properly check the variables according to the required specifications.

- The address arguments _offeringToken, _adminAddress, _projectOwner, _wkdCommit, and admin should not be set to zero address.

- The variable _startBlock should be greater than the current timestamp.

- The variable _endBlock should be greater than the _startBlock .

- The variable _tier2Percentage should be greater than zero and lower than 100.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contracts/launchpad.sol#L17
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
totalTokensOffered
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/launchpadDeployer.sol#L15,18,56,14,16,13,20,12,21,17,22,19 contracts/WKDCommit.sol#L36,30,73,44,63,53 contracts/launchpad.sol#L231,106,107,57,103,109,104,275,181,108,15,105,111,13,75,256,226,113,112,110,239,62
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.

```
_adminAddress
_wkd
_user
_offeringAmount
_projectOwner
userTiers
_offeringToken
removeWkdCommit
_startBlock
...
```

Recommendation

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

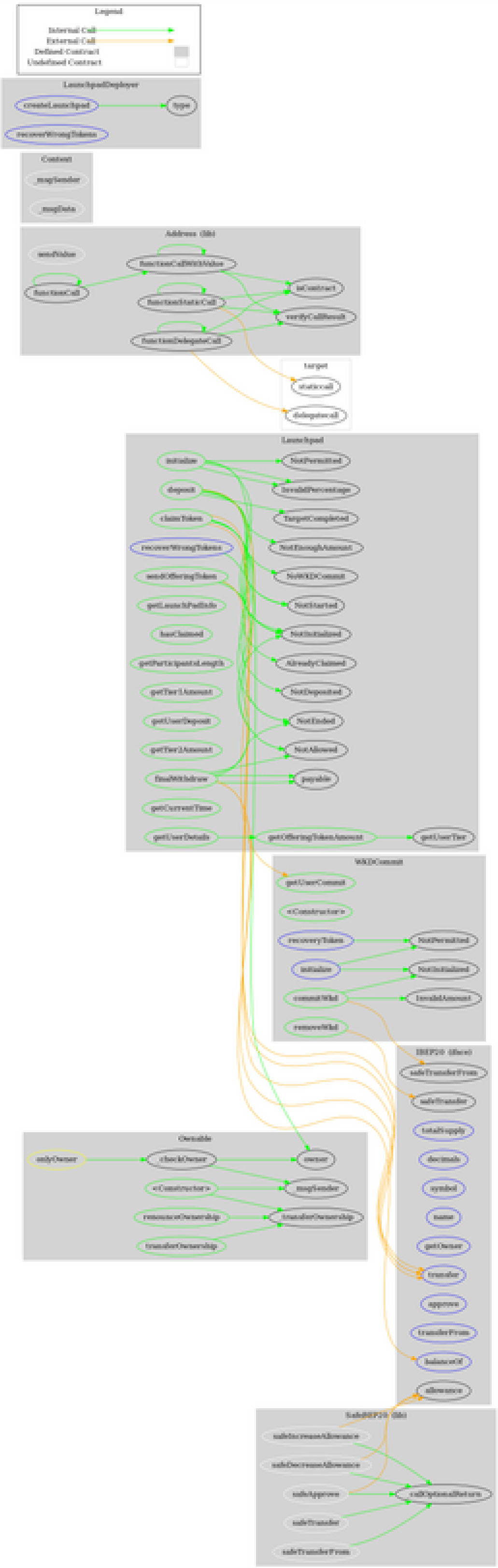
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Launchpad	Implementation	Ownable		
	initialize	Public	✓	-
	deposit	Public	Payable	-

	claimToken	Public	✓	-
	sendOfferingToken	Public	✓	-
	finalWithdraw	Public	✓	-
	getLaunchPadInfo	Public		-
	getOfferingTokenAmount	Public		-
	hasClaimed	Public		-
	getParticipantsLength	Public		-
	getUserTier	Public		-
	getTier1Amount	Public		-
	getUserDeposit	Public		-
	getTier2Amount	Public		-
	getUserDetails	Public		-
	getCurrentTime	Public		-
	recoverWrongTokens	External	✓	-
LaunchpadDeployer	Implementation	Ownable		
	createLaunchpad	External	✓	onlyOwner
	recoverWrongTokens	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	✓	
WKDCommit	Implementation			
	<Constructor>	Public	✓	-
	initialize	External	✓	-
	commitWkd	Public	✓	-
	removeWkd	Public	✓	-
	getUserCommit	Public		-
	recoveryToken	External	✓	-

Contract Flow



Domain Info

Domain Name	https://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 about 1 year before the creation of the audit. It will expire in almost 4 years.

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

Summary

The Launchpad contract operates as a launchpad. We state that admin privileges are necessary and required for proper protocol operations. Thus, we emphasize 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.

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