Cyberscope

Audit Report

Royalty Black Card

April 2022

SHA256 3f6e17c32a2f9d08edd1164a66331d9fdf81099916ef0827c63c84b0570acf4d

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	Royalty_Black_Card
Symbol	RBC
Domain	royaltyblackcard.club

Source Files

Filename	**SHA256**
contract.sol	3f6e17c32a2f9d08edd1164a66331d9fdf81099916ef0827c63c84b0570acf4d

Audit Updates

Initial Audit	18th April 2022
Corrected	

Contract Analysis

Royalty Black Card is an NFT project where users have the ability to mint NFT at a predefined fixed price.

The whitelisted accounts can mint NFTs with a fixed price that is 0.17 of the native currency.

The regular users can mint new NFTs up to a maximum quantity with a fixed price that is 0.22 of the native currency.

Contract Owner Privileges

- The contract owner has the authority to mint NFTs out of charge to the owner’s address.
- The contract owner has the authority to whitelist addresses.
- The contract owner has the authority to change the maximum quantity of the NFTs that can be bought in one request.
- The contract owner has the authority to pay a specific amount to the holders of some NFTs. This amount is the contract's accumulated paid funds.
- The contract owner has the authority to withdraw all the contract's funds.
- The contract owner has the authority to distribute funds to all of the NFT holders.
- The contract owner has the authority to pause the mint process.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CR	Code Repetition
	MC	Missing Check
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L12	Using Variables before Declaration
	L15	Local Scope Variable Shadowing

CR - Code Repetition

Criticality	minor
Location	contract.sol#L1523,1537,1547,1599,1609

Description

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

```
address ownerofid= ownerOf(i);
payable(ownerofid).transfer(NewBalance);
```

```
for (uint i = 0; i < chosenAmount; i++) {
_safeMint(msg.sender, totalsupply());
tokenId++;
}
```

Recommendation

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

MC - Missing Check

Criticality	minor
Location	contract.sol#L1583

Description

The contract proceeds with the payment even if the contract's balance is not enough to cover the payment expenses.

```
function reimursment(uint256[] memory ids, uint256 _balance) public onlyOwner{
    require(address(this).balance >= _balance, "eth must be less than or equal
to total balance");
    _balance= _balance*100000000000000000;
    uint256 len = ids.length;
    // uint256 NewBal= _balance/len;
    for (uint i=0;i<len;i++){
        uint256 _ids = ids[i];
        address addofowner = ownerOf(_ids);

        payable(addofowner).transfer(_balance);
    }

}
```

Recommendation

The reimbursement should initially check if the contract's balance is enough to cover the total amount of the payment process.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L847,928,935,942,961,985,1002,1016,1308,1451,1455,1458,1464,1467,1478,1481,1486,1514,1529,1542,1554,1569,1574,1615

Description

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

```
withdraw
addWhiteList
addToWhiteList
tokensOfOwner
reservenfts
mint
whitelist_mint
reserveTokens
flipwlmintStatus
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L1429,1430

Description

Constant state variables should be declared constant to save gas.

```
wl_mint_price
mint_price
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1031,889,1423,1464,1467,1514,1554,1583,1603,1428,1429,1430,1440,1444

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.

```
_revelNFT
__whiteList
wl_mint_price
mint_price
mint_quantity
Newam
_balance
_owner
whitelist_mint
...
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L1428

Description

There are segments that contain unused state variables.

```
mint_quantity
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L487,377,387,406,420,466,476,439,449,352,1154,954,272,288

Description

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

```
toHexString
baseURI
_burn
sendValue
functionStaticCall
functionDelegateCall
functionCallWithValue
functionCall
_verifyCallResult
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1514,1529,1565

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
__whiteList[address_] == true
require(bool,string)(ismintPaused == false,Public sale not active)
require(bool,string)(iswlmintPaused == false,Pre sale not active)
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1226,1228

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
reason
retval
```

Recommendation

The variables should be declared before any usage of them.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L1451,1496,1501,1554

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_owner
baseURI
tokenId
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
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		
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		

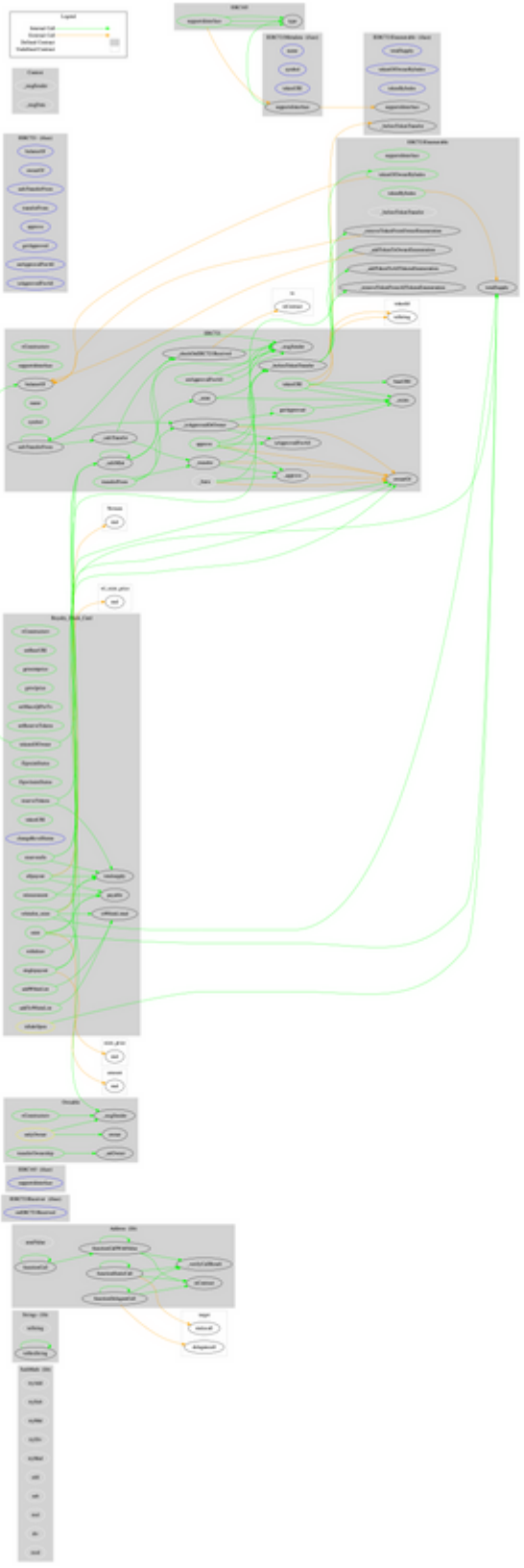
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
IERC721Receiver	Interface			
	onERC721Received	External	✓	-
IERC165	Interface			
	supportsInterface	External		-
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Enumerable	Interface	IERC721		
	totalSupply	External		-
	tokenOfOwnerByIndex	External		-
	tokenByIndex	External		-
IERC721Metadata	Interface	IERC721		
	name	External		-
	symbol	External		-
	tokenURI	External		-

Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
ERC721	Implementation	Context, ERC165, IERC721, IERC721Metadata		
	<Constructor>	Public	✓	-
	supportsInterface	Public		-
	balanceOf	Public		-
	ownerOf	Public		-
	name	Public		-
	symbol	Public		-
	tokenURI	Public		-
	baseURI	Internal		
	approve	Public	✓	-
	getApproved	Public		-
	setApprovalForAll	Public	✓	-
	isApprovedForAll	Public		-
	transferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	_safeTransfer	Internal	✓	
	_exists	Internal		
	_isApprovedOrOwner	Internal		
	_safeMint	Internal	✓	
	_safeMint	Internal	✓	
	_mint	Internal	✓	

	_burn	Internal	✓	
	_transfer	Internal	✓	
	_approve	Internal	✓	
	_checkOnERC721Received	Private	✓	
	_beforeTokenTransfer	Internal	✓	
ERC721Enumerable	Implementation	ERC721, IERC721Enumerable		
	supportsInterface	Public		-
	tokenOfOwnerByIndex	Public		-
	totalSupply	Public		-
	tokenByIndex	Public		-
	_beforeTokenTransfer	Internal	✓	
	_addTokenToOwnerEnumeration	Private	✓	
	_addTokenToAllTokensEnumeration	Private	✓	
	_removeTokenFromOwnerEnumeration	Private	✓	
	_removeTokenFromAllTokensEnumeration	Private	✓	
Royalty_Black_Card	Implementation	ERC721Enumerable, Ownable		
	<Constructor>	Public	✓	ERC721
	setBaseURI	Public	✓	onlyOwner
	getmintprice	Public		-
	getwlprice	Public		-
	setMaxxQtPerTx	Public	✓	onlyOwner
	setReserveTokens	Public	✓	onlyOwner
	flipmintStatus	Public	✓	onlyOwner
	flipwlmintStatus	Public	✓	onlyOwner
	reserveTokens	Public	✓	onlyOwner
	tokenURI	Public		-
	changeRevelStatus	External	✓	onlyOwner
	whitelist_mint	Public	Payable	isSaleOpen
	mint	Public	Payable	isSaleOpen

	reservenfts	Public	✓	onlyOwner
	tokensOfOwner	Public		-
	isWhiteListed	Public		-
	addToWhiteList	Public	✓	onlyOwner
	addWhiteList	Public	✓	onlyOwner
	reimursment	Public	✓	onlyOwner
	singlepayout	Public	✓	onlyOwner
	allpayout	Public	✓	onlyOwner
	withdraw	Public	✓	onlyOwner
	totalsupply	Private		

Contract Flow



Domain Info

Domain Name	royaltyblackcard.club
Registry Domain ID	D0CF0E77068AB442EBA4D747415A8B779-GDREG
Creation Date	2022-02-10T06:44:26Z
Updated Date	2022-02-15T06:44:26Z
Registry Expiry Date	2024-02-10T06:44:26Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	whois.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 2 months before the creation of the audit. It will expire in almost 2 years.

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

Summary

Royalty Black Card is an NFT project. It contains a list of whitelisted addresses. These addresses can mint NFTs at a different rate than the regular users. This audit focuses on the business logic, security concerns and 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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io