Cyberscope

Audit Report

Minetopia

August 2022

Type	ERC721
Network	RINKEBY ETH
Address	0x5248CE7CA895fac4645Eef9849835282F02Bc866
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	minetest17
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x5248CE7CA895fac4645Eef9849835282F02Bc866
Symbol	MT
Domain	minetopia.io

Source Files

Filename	**SHA256**
contract.sol	71d407461b84dbf9b5c0b6c618d302103d643a7f77dfda36ef5eda4e4045a8f5

Audit Updates

Initial Audit	26th August 2022 https://github.com/cyberscope-io/audits/blob/main/3-mine/v1/audit.pdf
Corrected	29th August 2022

Introduction

- The contract is taking advantage of the Merkle Proof algorithm to whitelist addresses without uploading all the addresses. This way, the contract owner saves gas for the whitelisting functionality.
- The maximum number of NFTs per user in the mint method is 10.
- There is no limit to how many NFTs can be produced, since the maxSupply is configurable.
- Only whitelisted users can transfer their NFTs.
- The users can mint NFTs by paying in native currency. The contract owner can change this amount.

Whitelist Verification

The verification process is based on an off-chain configuration. The contract owner is responsible for updating the in-chain “merkleRoot” in order to validate correctly the provided message. The verification algorithm is using the markle tree mechanism.

https://github.com/protofire/zeppelin-solidity/blob/master/contracts/MerkleProof.sol

According to the markle algorithm, the off-chain mechanism pre-defines all the index, recipient, token ids, the amount per token and the amount of tickets that will be burned combinations.

Hence, only predefined users have the transfer NFTs.

Contract Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	ST	Stops Transactions	Unresolved
●	MRA	Mint Role Access	Unresolved
●	UBI	User Balance Inconsistency	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L12	Using Variables before Declaration	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

ST - Stops Transactions

Criticality	minor / informative
Location	contract.sol#L1428
Status	Unresolved

Description

The contract owner has the authority to stop the transactions for all users including the owner. The owner may take advantage of it by calling the `pause` method.

```
 function mint(uint256 _mintCount) public payable onlyNotPaused {
        uint256 supply = totalSupply();
        uint256 tokenCount = balanceOf(msg.sender);
        require(tokenCount < maxCollection,           string(abi.encodePacked('You can only mint ',
maxCollection.toString(), ' cards per wallet')));
        require(supply < maxSupply,                   'No more left');
        require(_mintCount > 0,                       'Mint count cannot be 0');
        require(msg.value >= mintPrice,               'Ether value is too low');

        for (uint256 i = 1; i <= _mintCount; i++) {
            _safeMint(msg.sender, supply + i);
        }

        // require(payable(owner()).send(msg.value));
}
```

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.

MRA - Mint Role Access

Criticality	minor
Location	contract.sol#L1369
Status	Unresolved

Description

The address *0xdAb1a1854214684acE522439684a145E62505233* has the authority to mint tokens to any address.

```
function crossmint(address _to) public payable {
    uint256 supply = totalSupply();
    require(mintPrice == msg.value,                "Incorrect ETH value sent");
    require(supply < maxSupply,                    "No more left");
    require(msg.sender == 0xdAb1a1854214684acE522439684a145E62505233,
        "This function is for Crossmint only."
    );

    _safeMint(_to, supply+ 1);
}
```

Recommendation

The specific address 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.

UBI - User Balance Inconsistency

Criticality	minor
Location	contract.sol
Status	Unresolved

Description

The method “mint” does not allow the users to mint more than 10 NFTs. On the other hand, the “transferNFT” method does not contain any restriction. So, in the “transferNFT” a user can receive more than 10 NFTs and break the mint’s requirement.

```
require(tokenCount < maxCollection,            string(abi.encodePacked('You
can only mint ', maxCollection.toString(), ' cards per wallet')));
```

Recommendation

The contract should set the same restriction in all the cases that the user receives NFTs.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L709,716,723,742,766,783,1257,1265,1352,1367,1378,1406,1410,1414,1418,1424,1428,1432,1441,1445
Status	Unresolved

Description

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

```
name
symbol
tokenURI
approve
setApprovalForAll
transferFrom
renounceOwnership
transferOwnership
mint
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L812,1279,1317,1321,1326,1352,1367,1410,1414,1418,1424,1428
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.

```
_data
minetest17
_proof
_merkleRoot
_to
_tokenId
_mintCount
_mint_price
_max_collection
...
```

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#L1410
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.

```
mintPrice = _mint_price
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L423,433,452,466,512,522,485,495,398,539,735,935,314,330
Status	Unresolved

Description

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

```
functionCall
functionCallWithValue
functionDelegateCall
functionStaticCall
sendValue
verifyCallResult
_baseURI
_burn
toHexString
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1007,1009
Status	Unresolved

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.

```
retval
reason
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1332,1334
Status	Unresolved

Description

The are variables that are defined in the local scope and are not initialized.

```
_holders
_holdersCount
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

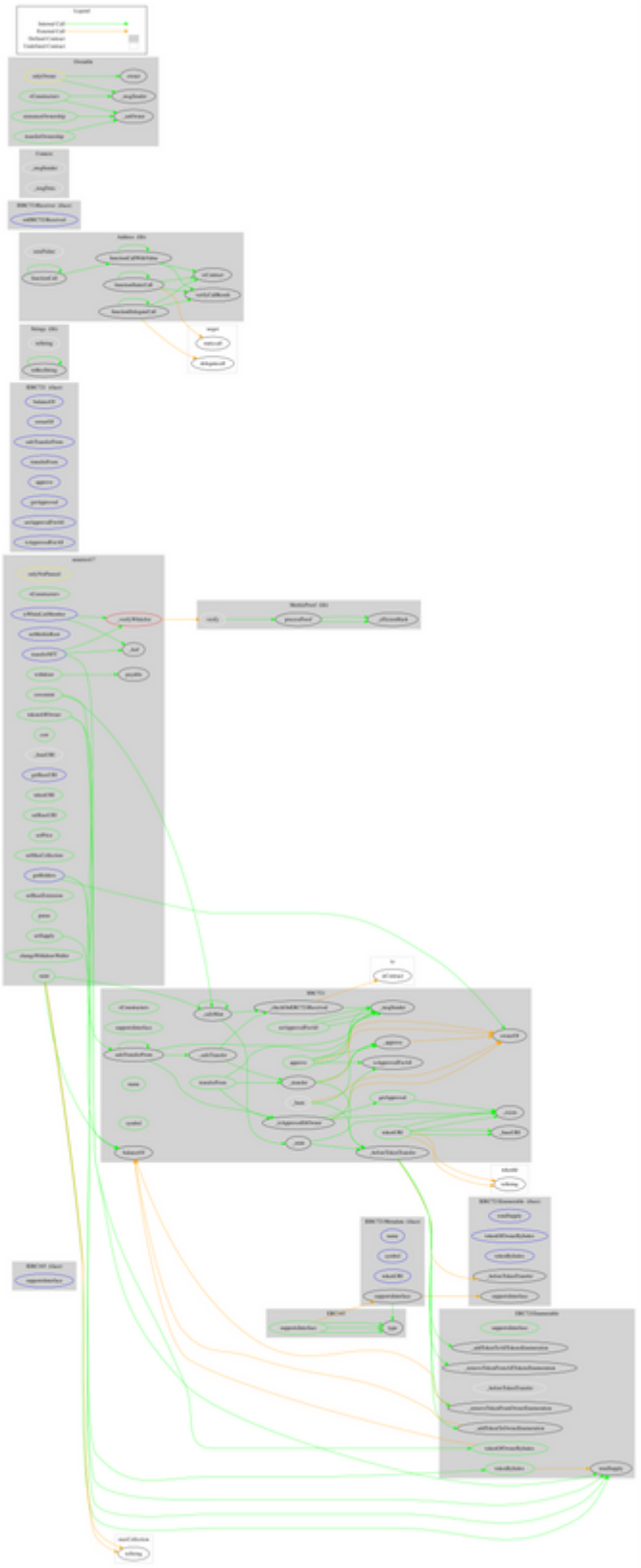
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC165	Interface			
	supportsInterface	External		-
MerkleProof	Library			
	verify	Internal		
	processProof	Internal		
	_efficientHash	Private		
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Enumerable	Interface	IERC721		
	totalSupply	External		-
	tokenOfOwnerByIndex	External		-
	tokenByIndex	External		-
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
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	Internal		
IERC721Metadata	Interface	IERC721		
	name	External		-
	symbol	External		-
	tokenURI	External		-
IERC721Receiver	Interface			
	onERC721Received	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
minetest17	Implementation	ERC721Enumerable, Ownable		
	<Constructor>	Public	✓	ERC721
	_leaf	Private		
	_verifyWhitelist	Private		
	isWhiteListMember	External		-
	setMerkleRoot	External	✓	onlyOwner
	transferNFT	External	✓	-
	getHolders	External		-
	mint	Public	Payable	onlyNotPaused
	crossmint	Public	Payable	-
	cost	Public		-
	_baseURI	Internal		
	getBaseURI	External		-
	tokenURI	Public		-
	setBaseURI	Public	✓	onlyOwner
	setPrice	Public	✓	onlyOwner
	setMaxCollection	Public	✓	onlyOwner
	setSupply	Public	✓	onlyOwner
	setBaseExtension	Public	✓	onlyOwner
	pause	Public	✓	onlyOwner
	tokensOfOwner	Public		-
	changeWithdrawWallet	Public	✓	onlyOwner
	withdraw	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	minetopia.io
Registry Domain ID	b61e07da90ef43c49da54500aff07edd-DONUTS
Creation Date	2022-06-14T14:40:22Z
Updated Date	2022-08-20T15:16:40Z
Registry Expiry Date	2023-06-14T14:40:22Z
Registrar WHOIS Server	whois.porkbun.com
Registrar URL	http://porkbun.com
Registrar	Porkbun LLC
Registrar IANA ID	1861

The domain was created 2 months before the creation of the audit. It will expire in 10 months.

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

Summary

This audit focuses on the business logic issues, the security concerns and the potential improvements. The contract implements an NFT contract where users can mint NFTs by paying in native currency. Users can transfer NFTs only if they are whitelisted.

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