Cyberscope

Audit Report

RADIKAL

November 2022

Type	ERC20
Network	MATIC
Address	0xf78a1108bced9cf6a6e1f686fc537c976ee244cd
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	ERC20RDK
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Explorer	https://polygonscan.com/token/0xF78a1108Bced9CF6a6E1f686fC537c976ee244CD
Symbol	RDK
Decimals	18
Total Supply	5,000,000
Domain	radikalriders.app

Audit Updates

Initial Audit	10th December 2022 https://github.com/cyberscope-io/audits/blob/main/rdk/audit.pdf
Corrected	17th November 2022

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	75e3c97011e75627ffb36f4a2799a4e887e1a3e27ed427490e82d7b6f51cc5c9
@openzeppelin/contracts/token/ERC20/ERC20.sol	f7831910f2ed6d32acff6431e5998baf50e4a00121303b27e974aab0ec637d79
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	c2b06bb4572bb4f84bfc5477dadc0fcc497cb66c3a1bd53480e68bedc2e154a6
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
contracts/3. Token/ERC20RDK.sol	823c16f10122b2302ba6140a66e3fb418f4e035f32c16d1c8d858d6b707115ba

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Unresolved
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

ST - Stops Transactions

Criticality	medium
Location	contract.sol#L36,54
Status	Unresolved

Description

The contract owner has the authority to stop the transactions for all users excluding the radikalContracts.

Example

Addresses	balances	_balancesTransferable
Distributor Address	5,000,000	0
Address 1	0	0
Address 2	0	0

Initially, only the distributor can execute a transaction. So, let's assume the **distributor address** sends 500,000 tokens to **address 1**.

Addresses	balances	_balancesTransferable
Distributor Address	4,500,000	0
Address 1	500,000	500,000
Address 2	0	0

Now **address 1** sends the same amount to **address 2**.

Addresses	balances	_balancesTransferable
Distributor Address	4,500,000	0
Address 1	0	0
Address 2	500,000	0

The _balancesTransferable for **address 2** remains zero, but the ERC20 balance doesn't. So, if **address 2** tries to make a transaction it will fail. This can be prevented if the distributor adds **address 2** to the **radikalContracts** array.

```
    function _beforeTokenTransfer(address from, address to, uint256 amount)
internal virtual override {
        address[] memory _radikalContracts = radikalContracts;
        bool userToUser = true;
        for(uint i = 0; i < _radikalContracts.length; i++) {
           if(from == _radikalContracts[i] || to == _radikalContracts[i]) {
               userToUser = false;
           }
        }
        if(userToUser == true) {
            require(_balancesTransferable[from] >= amount, "ERC20: transfer
amount exceeds transferable balance");
        }
    }
```

```
    function _afterTokenTransfer(address from, address to, uint256 amount)
internal virtual override {
        address[] memory _radikalContracts = radikalContracts;
        bool fromContract = false;
        bool toContract = false;
        for(uint i = 0; i < _radikalContracts.length; i++) {
           if(from == _radikalContracts[i]) {
               fromContract = true;
           } else if(to == _radikalContracts[i]) {
               toContract = true;
           }
        }
        if(fromContract == false && toContract == false) {
            _balancesTransferable[from] -= amount;
        } else if(fromContract == true && toContract == false) {
            _balancesTransferable[to] += amount;
        } else if(fromContract == false && toContract == true) {
            uint balance = balanceOf(from);
            if(balance < _balancesTransferable[from]) {
                _balancesTransferable[from] = balance;
            }
        }
    }
```

Recommendation

The contract should allow the users to trade without limitation.

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.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	BLC	Business Logic Concern	Unresolved
●	L11	Unnecessary Boolean equality	Unresolved

BLC - Business Logic Concern

Criticality	minor / informative
Location	contracts/ERC20RDK.sol#L36,55
Status	Unresolved

Description

If both `from` and `to` addresses belong to the radical addresses, then the contract will assume that only the `from` address is issued from the radical addresses.

```
for(uint i = 0; i < _radikalContracts.length; i++) {
    if(from == _radikalContracts[i]) {
        fromContract = true;
        break;
    } else if(to == _radikalContracts[i]) {
        toContract = true;
        break;
    }
}
```

Recommendation

The contract should enable both from and to addresses if they belong to the radical addresses.

L11 - Unnecessary Boolean equality

Criticality	minor / informative
Location	contracts/ERC20RDK.sol#L36,55
Status	Unresolved

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.

```
userToUser == true
fromContract == true && toContract == false
fromContract == false && toContract == false
fromContract == false && toContract == true
```

Recommendation

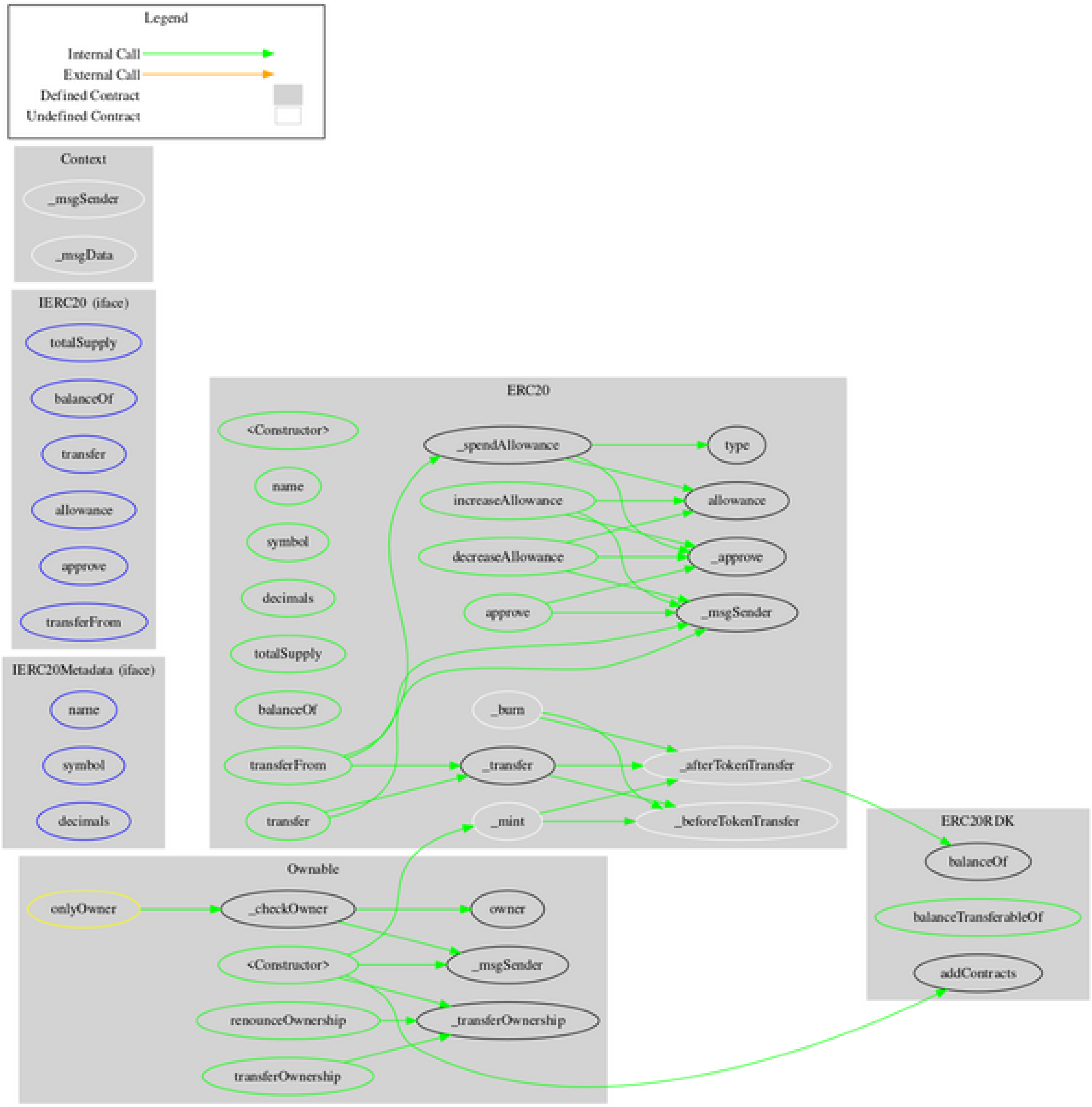
Remove the equality to the boolean constant.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	

IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC20RDK	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
	addContracts	Public	✓	onlyOwner
	addPair	Public	✓	onlyOwner
	balanceTransferableOf	Public		-

Contract Flow



Domain Info

Domain Name	radikalriders.app
Registry Domain ID	482839258-APP
Creation Date	2021-12-28T17:00:04Z
Updated Date	2022-06-28T11:21:18Z
Registry Expiry Date	2022-12-28T17:00:04Z
Registrar WHOIS Server	whois.nic.google
Registrar URL	https://www.dondominio.com/
Registrar	Soluciones Corporativas IP, SLU
Registrar IANA ID	1383

The domain was created 12 months before the creation of the audit. It will expire in 18 days.

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

Summary

The Smart Contract analysis reported one medium severity issue. The contract owner has the authority to stop transactions. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io