Cyberscope

Audit Report
Ride2Earn

April 2022

Table of Contents

Contract Review

Contract Name	RD2E
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	GNU GPLv3
Explorer	https://bscscan.com/token/0x66E1e0EA96aBB155932aF031d41Bf1F21b235958
Symbol	RD2E
Decimals	18
Total Supply	127,120,000
Domain	ride2e.io

Source Files

Filename	**SHA256**
contract.sol	7ba0e6f552401200517d327a00da488f05e9a8af9e698054443a60395c00f32d

Audit Updates

Initial Audit	30th April 2022
Corrected	19th May 2022

The contract delegates the fees calculation logic to an external contract.
The address of the external contract can be changed.

Contract Analysis

● Critical   ● Medium   ● Minor   ● Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L1050

Description

The contract owner has the authority to stop transactions or increase the fees for all users excluding the owner. The owner may take advantage of it by setting a malicious implementation to the antiBot address.

```
uint256 fee = antiBot.protect(sender, recipient, amount);

if (fee > 0) {
    super._transfer(sender, _addressFee, fee);
}
```

Recommendation

The contract should allow setting the antiBot address once and the address should be verified so the investors will know the behaviour of this contract.

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.

MT - Mint Tokens

Criticality	critical
Location	contract.sol#L1004

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `mint` function with an amount lower than the total supply and later mint the remaining tokens. As a result the contract tokens will be highly inflated.

```solidity
function mint(address to, uint256 amount) public onlyOwner {
    require(totalSupply() + amount <= TOTAL_SUPPLY, "RD2E: max supply
exceeded");
    _mint(to, amount);
}
```

Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

Contract Diagnostics

● Critical   ● Medium   ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L531,539,556,570,582,605,627,650,670,874,889,949,957,1004

Description

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

```
mint
transferOwnership
renounceOwnership
burnFrom
burn
decreaseAllowance
increaseAllowance
transferFrom
approve
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1009,1013,1018,1025,1036

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.

```
_token
_amount
_dev
_enabled
_antiBot
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L87,116,176,186,149,159,62,386,410,401,370

Description

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

```
safeTransferFrom
safeIncreaseAllowance
safeDecreaseAllowance
safeApprove
sendValue
functionStaticCall
functionDelegateCall
functionCallWithValue
functionCall
...
```

Recommendation

Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metad ata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
SafeERC20	Library			
	safeTransfer	Internal	✓	

	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
Context	Implementation			
	_msgSender	Internal		
	_msgData	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	✓	
ERC20Burnable	Implementation	Context, ERC20		

	burn	Public	✓	-
	burnFrom	Public	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
AntiBot	Interface			
	protect	External	✓	-
RD2E	Implementation	ERC20, ERC20Burnable, Ownable		
	<Constructor>	Public	✓	ERC20
	mint	Public	✓	onlyOwner
	setAntiBotAddress	External	✓	onlyOwner
	setAntiBnotEnabled	External	✓	onlyOwner
	setDevAddress	External	✓	onlyOwner
	withdrawalStuck	External	✓	onlyOwner
	withdrawalStuckMax	External	✓	onlyOwner
	_transfer	Internal	✓	

Contract Flow

Domain Info

Domain Name	ride2e.io
Registry Domain ID	e31fb8f4e4744b29b55823552afa50f1-DONUTS
Creation Date	2022-04-01T16:29:28Z
Updated Date	2022-04-12T03:30:46Z
Registry Expiry Date	2023-04-01T16:29:28Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 29 days before the creation of the audit. It will expire in 11 months.

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

Summary

There are some functions that can be abused by the owner, like manipulating transactions and minting tokens. A stable antiBot address and 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

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