Cyberscope

Audit Report

Crazy Treasure Token

June 2022

SHA256 99953d29ddf5a93cfdc4b9b7427f438f922775ccba1346f7616796f6b67bf8fc

Audited by © cyberscope

Table of Contents

Source Files

Filename	SHA256
contract.sol	99953d29ddf5a93cfdc4b9b7427f438f922775ccba1346f7616796f6b67bf8fc

Audit Updates

Initial Audit	12th June 2022
Corrected	

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

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	STC	Succeeded Transfer Check
●	MC	Missing Check
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L625,635

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.

```
token.transfer(msg.sender, token.balanceOf(address(this)));
//
token.transfer(msg.sender, tokensToWithdraw);
```

Recommendation

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

MC - Missing Check

Criticality	minor
Location	contract.sol#L609,1017

Description

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

The constructor mints tokens according to the arguments. The initialAccounts length should be equal with the initialBalances length. Each initialBalances entry should be greater than zero.

```
constructor(address[] memory initialAccounts, uint256[] memory initialBalances)
payable ERC20("Crazy Treasure Token", "CTT") {
    for(uint8 i = 0; i < initialAccounts.length; i++) {
        require(initialAccounts[i] != address(0));
        _mint(initialAccounts[i], initialBalances[i]);
    }
}
```

The values that are initialized on the constructor are used as diviators in the expressions. For instance, the interval property should not be zero since it will revert the transactions.

```
constructor(address _beneficiary, uint256 _start, uint256 _duration, uint256
_interval,uint256 _initialTokens) {
    beneficiary = _beneficiary;
    start = _start;
    duration = _duration;
    interval = _interval;
    initialTokens = _initialTokens;
}
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L551,559,617,622,705,713,730,737,744,756,775,793,815,834,896,911

Description

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

```
burnFrom
burn
decreaseAllowance
increaseAllowance
transferFrom
approve
transfer
balanceOf
totalSupply
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L1038,1042,1046,1036,1044,1040,1048

Description

Constant state variables should be declared constant to save gas.

```
TEAM_OWNER_ADDRESS
MARKETING_OWNER_ADDRESS
LIQUIDITY_OWNER_ADDRESS
GAME_POOL_OWNER_ADDRESS
EXCHANGE_OWNER_ADDRESS
COMMUNITY_OPERATIONS_OWNER_ADDRESS
COMMUNITY_GOVERNANCE_OWNER_ADDRESS
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1036,1038,1040,1042,1044,1046,1048

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.

```
TEAM_OWNER_ADDRESS
EXCHANGE_OWNER_ADDRESS
LIQUIDITY_OWNER_ADDRESS
COMMUNITY_OPERATIONS_OWNER_ADDRESS
MARKETING_OWNER_ADDRESS
COMMUNITY_GOVERNANCE_OWNER_ADDRESS
GAME_POOL_OWNER_ADDRESS
```

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#L78,88,107,121,167,177,140,150,25,53,194

Description

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

```
verifyCallResult
sendValue
isContract
functionStaticCall
functionDelegateCall
functionCallWithValue
functionCall
...
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L622

Description

Performing divisions before multiplications may cause lose of prediction.

```
tokensByPart = initialTokens.div(parts)
```

Recommendation

The multiplications should be prior to the divisions.

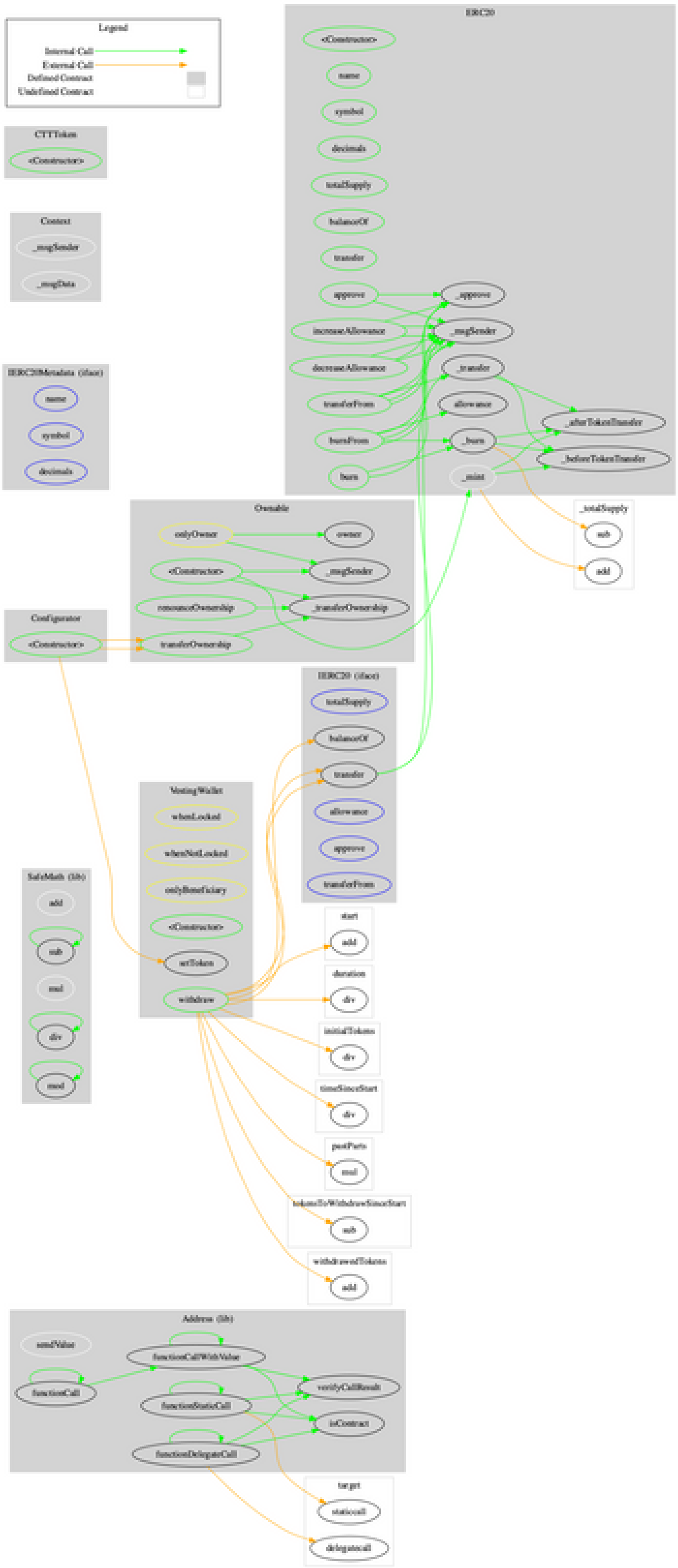
Contract Functions

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		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-

	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
VestingWallet	Implementation	Ownable		
	<Constructor>	Public	✓	-
	setToken	Public	✓	onlyOwner whenNotLocked
	withdraw	Public	✓	onlyBeneficiary whenLocked
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	Public	✓	-
	burnFrom	Public	✓	-
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
CTTToken	Implementation	ERC20, Ownable		
	<Constructor>	Public	Payable	ERC20
Configurator	Implementation			
	<Constructor>	Public	✓	-

Contract Flow



Domain Info

Domain Name	crazy-treasure.com
Registry Domain ID	2689177276_DOMAIN_COM-VRSN
Creation Date	2022-04-14T03:04:01Z
Updated Date	2022-04-14T03:20:20Z
Registry Expiry Date	2023-04-14T03:04:01Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	https://www.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created in 10 months before the creation of the audit.

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

Summary

Crazy Treasure Token is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions.

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