Cyberscope

Audit Report

Rodgo Coin

January 2024

Network	TRON
Address	TPh8djPzBEQfZojsoXnkT7VUFx3gt29v6f
Audited by	© cyberscope

Analysis

Critical Medium Minor / Informative Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	L09	Dead Code Elimination	Unresolved
	L19	Stable Compiler Version	Unresolved

Table of Contents

Review

Contract Name	Token
Testing Deploy	https://testnet.bscscan.com/address/0x4985ece71cfa2acfaf62e0ab5d255b7d99764965
Explorer	https://tronscan.org/#/address/tph8djpzbeqfzojsoxnkt7vufx3gt29v6f
Symbol	RodGO
Decimals	12
Total Supply	30,000,000

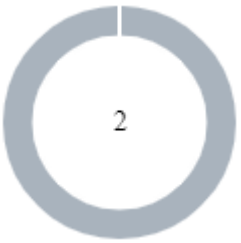
Audit Updates

Initial Audit	21 Jan 2024 https://github.com/cyberscope-io/audits/blob/main/rodgo/v1/audit.pdf
Corrected Phase 2	30 Jan 2024

Source Files

Filename	SHA256
contracts/testingDeploy/Token.sol	b640da100ad8fda5c0cb006447d65ef68d a28cf9b93d3c488e4c68e251ae5cfb
contracts/testingDeploy/TRC20Detailed.sol	b366b70aac301e3950fb180737aa9fc56e4 77673a0057242bd0b911cec954b89
contracts/testingDeploy/TRC20.sol	8e02cd170523d910b6bc9769eea166febb 4b5a4d83a48a0ac1fbf6dfbbd9de7b
contracts/testingDeploy/SafeMath.sol	38d61179ebe62a2bd1e77383348d057c64 b45aae42afd2ce3282071d081842f6
contracts/testingDeploy/ITRC20.sol	53416c3af0f3a2f9b8b7b410ff5a583458eb 26728ed7b54b83c03d8fcaa79775

Findings Breakdown



Critical	0
Medium	0
Minor / Informative	2

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	2	0	0	0

L09 - Dead Code Elimination

Criticality	Minor / Informative
Location	contracts/testingDeploy/TRC20.sol#L190,225
Status	Unresolved

Description

In Solidity, dead code is code that is written in the contract, but is never executed or reached during normal contract execution. Dead code can occur for a variety of reasons, such as:

- Conditional statements that are always false.
- Functions that are never called.
- Unreachable code (e.g., code that follows a return statement).

Dead code can make a contract more difficult to understand and maintain, and can also increase the size of the contract and the cost of deploying and interacting with it.

```
function _burn(address account, uint256 value) internal {
        require(account != address(0), "TRC20: burn from the zero
address");

        _totalSupply = _totalSupply.sub(value);
        _balances[account] = _balances[account].sub(value);
        emit Transfer(account, address(0), value);
    }

function _burnFrom(address account, uint256 amount) internal {
        _burn(account, amount);
        _approve(account, msg.sender,
_allowances[account][msg.sender].sub(amount));
    }
```

Recommendation

To avoid creating dead code, it's important to carefully consider the logic and flow of the contract and to remove any code that is not needed or that is never executed. This can help improve the clarity and efficiency of the contract.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	contracts/testingDeploy/TRC20Detailed.sol#L1 contracts/testingDeploy/TRC20.sol#L1 contracts/testingDeploy/Token.sol#L3 contracts/testingDeploy/SafeMath.sol#L1 contracts/testingDeploy/ITRC20.sol#L1
Status	Unresolved

Description

The `^` symbol indicates that any version of Solidity that is compatible with the specified version (i.e., any version that is a higher minor or patch version) can be used to compile the contract. The version lock is a mechanism that allows the author to specify a minimum version of the Solidity compiler that must be used to compile the contract code. This is useful because it ensures that the contract will be compiled using a version of the compiler that is known to be compatible with the code.

```
pragma solidity ^0.5.0;
```

Recommendation

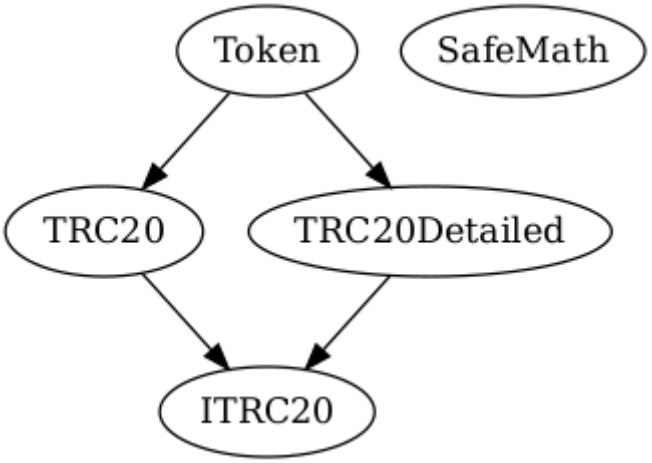
The team is advised to lock the pragma to ensure the stability of the codebase. The locked pragma version ensures that the contract will not be deployed with an unexpected version. An unexpected version may produce vulnerabilities and undiscovered bugs. The compiler should be configured to the lowest version that provides all the required functionality for the codebase. As a result, the project will be compiled in a well-tested LTS (Long Term Support) environment.

Functions Analysis

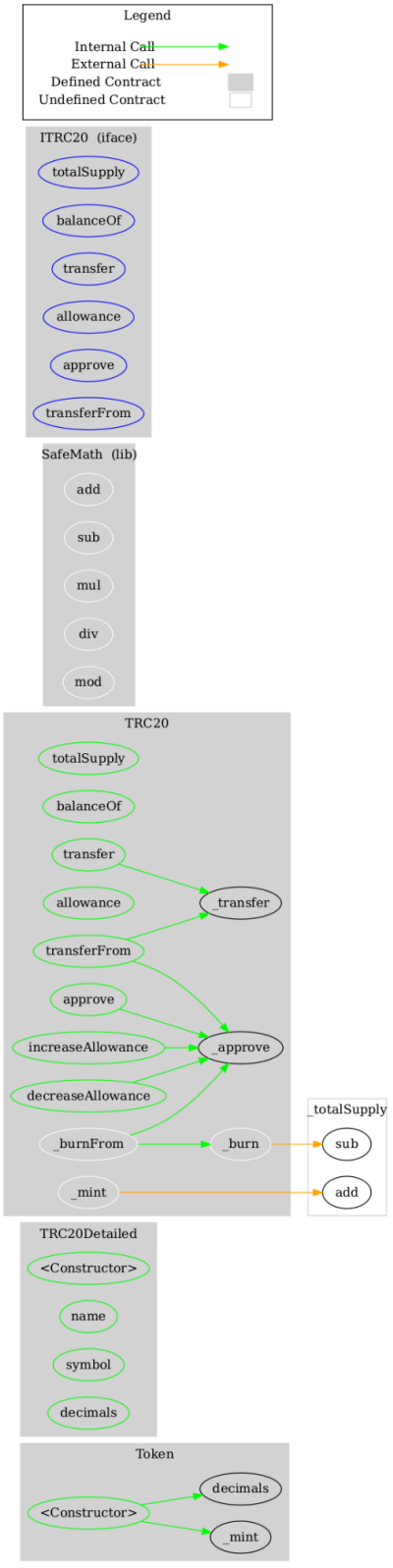
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Token	Implementation	TRC20, TRC20Detailed		
		Public	✓	TRC20Detailed
TRC20Detailed	Implementation	ITRC20		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
TRC20	Implementation	ITRC20		
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-

	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_burnFrom	Internal	✓	
SafeMath	Library			
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
ITRC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
ITRC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
SafeMath (lib)
add
sub
mul
div
mod
TRC20
totalSupply
balanceOf
transfer
allowance
transferFrom
approve
increaseAllowance
decreaseAllowance
_burnFrom
_mint
_transfer
_approve
_burn
_totalSupply
sub
add
TRC20Detailed
<Constructor>
name
symbol
decimals
Token
<Constructor>
decimals
_mint

Summary

Rodgo Coin contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. Rodgo Coin is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors 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

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