Cyberscope

Audit Report

Telefy

March 2022

Type	ERC20
Network	ETH RINKEBY
Address	0x995CA0B86446Cd460C68d8Ac537b7dFbC86A9370
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MATToken
Compiler Version	v0.8.12+commit.f00d7308
Optimization	runs
Licence	MIT
Explorer	https://rinkeby.etherscan.io/address/0x995CA0B86446Cd460C68d8Ac537b7dFbC86A9370
Symbol	MA
Decimals	18
Total Supply	10000
Source	contract.sol
Domain	telefy.finance

Audit Updates

Initial Audit	12th March 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

MT - Mint Tokens

Criticality	critical
Location	contract.sol#L575

Description

The “minter” role has the authority to mint tokens. The minter may take advantage of it by calling the `mint` function. As a result the contract tokens will be highly inflated

```
function mint(address account, uint256 amount) external {
        require(_msgSender() == _minter, "MAT::mint: only the minter can mint");
        require(account != address(0), "MAT: mint to the zero address");

        _beforeTokenTransfer(address(0), account, amount);

        _totalSupply = safeAdd(_totalSupply, amount);
        balances[account] = safeAdd(balances[account], amount);
        emit Transfer(address(0), account, amount);
        _moveDelegates(address(0), _delegates[account], 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.

BT - Burn Tokens

Criticality	critical
Location	contract.sol#L1

Description

The “minter” role has the authority to burn tokens from a specific address. The minter may take advantage of it by calling the `burn` function. As a result the targeted contract address will lose the corresponding tokens.

```
function burn(address account, uint256 amount) external {
       require(_msgSender() == _minter, "MAT::burn: only the minter can mint");
       require(account != address(0), "MAT: burn from the zero address");

       _beforeTokenTransfer(account, address(0), amount);

       balances[account] = safeSub(balances[account], amount);
       _totalSupply = safeSub(_totalSupply, amount);
       emit Transfer(account, address(0), amount);
       _moveDelegates(address(0), _delegates[account], 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
	CR	Code Repetition
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination

CR - Code Repetition

Criticality	minor
Location	contract.sol#L383,405

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.
The functions transfer and transferMultiple share the same functionality underneath. There could be a function that will be commonly re-used from both functions.

```
balances[_msgSender()] = safeSub(balances[_msgSender()], tokens);
balances[to] = safeAdd(balances[to], tokens);
_moveDelegates(_delegates[_msgSender()], _delegates[to], tokens);
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L72,76,82,86,102,128,132,395,506,517 and 2 more

Description

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

```
transferAnyERC20Token
approveAndCall
decreaseAllowance
increaseAllowance
transferMultiple
acceptOwnership
transferOwnership
receiveApproval
transferFrom
...
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L128,301

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.

```
_delegates
_newOwner
```

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#L255,209,218,231,245,160,189,42,47,33

Description

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

```
safeMul
safeMod
safeDiv
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
...
```

Recommendation

Remove unused functions.

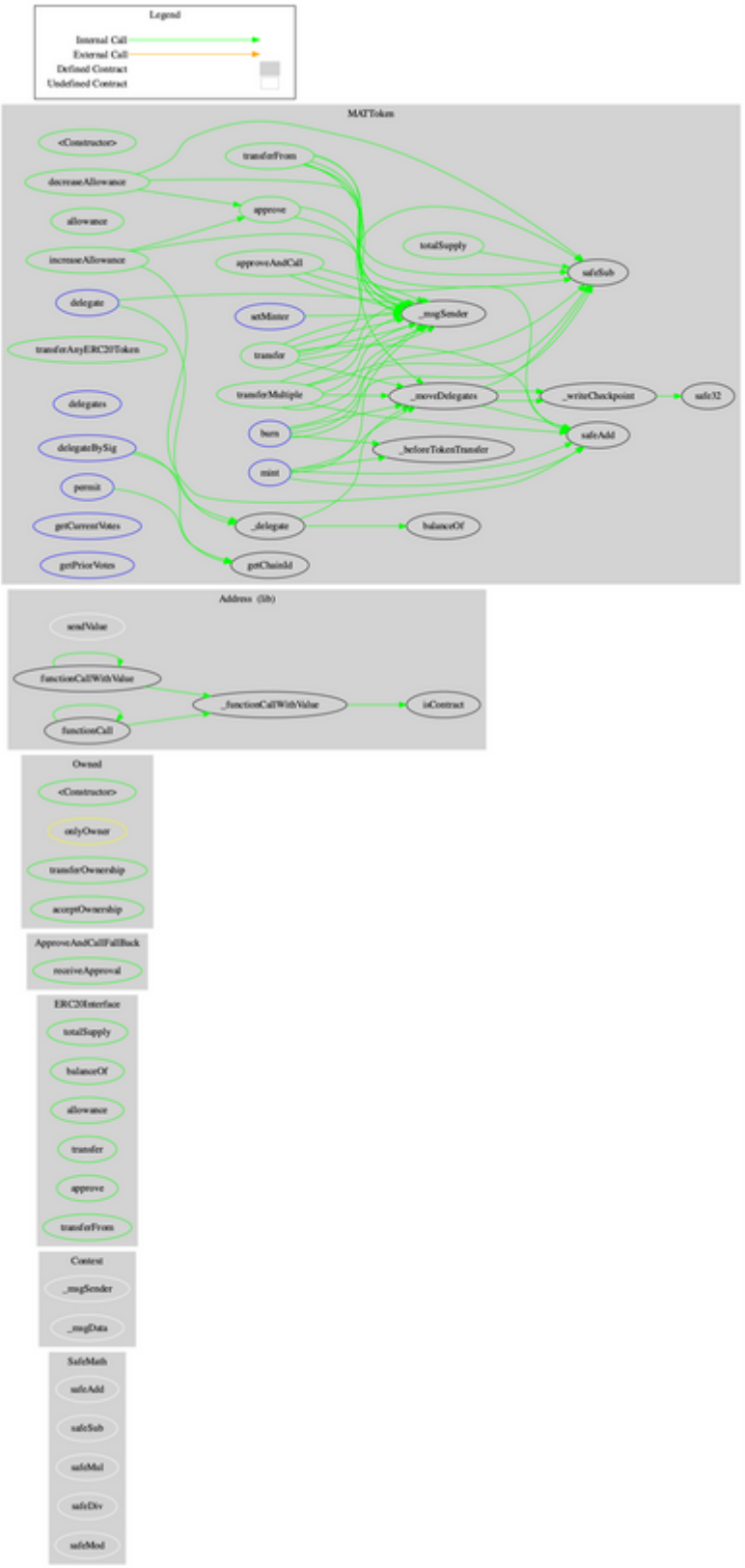
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Implementation			
	safeAdd	Internal		
	safeSub	Internal		
	safeMul	Internal		
	safeDiv	Internal		
	safeMod	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC20Interface	Implementation			
	totalSupply	Public		-
	balanceOf	Public		-
	allowance	Public		-
	transfer	Public	✓	-
	approve	Public	✓	-
	transferFrom	Public	✓	-
ApproveAndCallFallBack	Implementation			
	receiveApproval	Public	✓	-
Owned	Implementation			
	<Constructor>	Public	✓	-
	transferOwnership	Public	✓	onlyOwner
	acceptOwnership	Public	✓	-

Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	_functionCallWithValue	Private	✓	
MATToken	Implementation	ERC20Interface, Owned, SafeMath, Context		
	<Constructor>	Public	✓	-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	transferMultiple	Public	✓	-
	approve	Public	✓	-
	permit	External	✓	-
	transferFrom	Public	✓	-
	allowance	Public		-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	approveAndCall	Public	✓	-
	transferAnyERC20Token	Public	✓	onlyOwner
	burn	External	✓	-
	mint	External	✓	-
	setMinter	External	✓	-
	delegates	External		-
	delegate	External	✓	-
	delegateBySig	External	✓	-
	getCurrentVotes	External		-
	getPriorVotes	External		-
	_delegate	Internal	✓	
	_moveDelegates	Internal	✓	

	_writeCheckpoint	Internal	✓	
	safe32	Internal		
	getChainId	Internal		
	_beforeTokenTransfer	Internal	✓	

Contract Flow



Domain Info

Domain Name	telefy.finance
Registry Domain ID	e1860f22e64043d1a4b018459b915ca2-DONUTS
Creation Date	2021-08-31T17:05:43Z
Updated Date	2021-09-05T17:05:52Z
Registry Expiry Date	2022-08-31T17:05:43Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 6 months before the creation of the audit. It will expire in 6 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 minting and burning tokens to arbitrary addresses. The contract also implements a governance functionality similar to the compound protocol that does not affect the contract transaction flow. 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