Cyberscope

Audit Report

BABY RATSCOIN

June 2022

Type	BEP20
Network	BSC
Address	0xc5f7c32d1b8ab89d742609325b213624282aaff1
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BRATS
Compiler Version	v0.8.10+commit.fc410830
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0xc5F7C32D1B8Ab89d742609325B213624282Aaff1
Symbol	BRATS
Decimals	9
Total Supply	1,000,000,000,000,000
Domain	babyratscoin.org

Source Files

Filename	**SHA256**
contract.sol	38ee57148baa3064e1e5ce5aa4b4bc41daf9b9161e0552888c1b6b30fce5fc9f

Audit Updates

Initial Audit	2nd 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
	BLC	Business Logic Concern
	FSA	Fixed Swap Address
	CO	Code Optimization
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable

BLC - Business Logic Concern

Criticality	minor
Location	contract.sol#L750, 732, 683

Description

The business logic seems peculiar. The implementation may not follow the expected behavior. In the following code segments, the order of the parameters from the callee `_getTValues` function does not match the order of the parameters that are in the declaration of the same function.

```
function _getTValues(
        uint256 tAmount,
        uint256 liquidityFee,
        uint256 teamFee
    )
        private
        pure
        returns (
            uint256,
            uint256,
            uint256
        )
    {
        uint256 tFee = tAmount.mul(liquidityFee).div(100);
        uint256 tTeam = tAmount.mul(teamFee).div(100);
        uint256 tTransferAmount = tAmount.sub(tFee).sub(tTeam);
        return (tTransferAmount, tFee, tTeam);
    }
```

```
(uint256 tTransferAmount, uint256 tFee, uint256 tTeam) = _getTValues(
            tAmount,
            _taxFee,
            _liquidityFee
        );
```

```
(
            uint256 rAmount,
            uint256 rTransferAmount,
            uint256 rFee,
            uint256 tTransferAmount,
            uint256 tFee,
            uint256 tTeam
        ) = _getValues(tAmount);
        _rOwned[sender] = _rOwned[sender].sub(rAmount);
        _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount);
        _takeTeam(tTeam);
        _reflectFee(rFee, tFee);
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

FSA - Fixed Swap Address

Criticality	minor
Location	contract.sol#L384

Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade.

```
IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(
            0x10ED43C718714eb63d5aA57B78B54704E256024E
        );
        uniswapV2Router = _uniswapV2Router;
        uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory())
            .createPair(address(this), _uniswapV2Router.WETH());
```

Recommendation

It could be better to allow the swap address mutation in case of future swap updates.

CO - Code Optimization

Criticality	minor
Location	contract.sol#L339, 601, 631, 734

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations. The `_taxFee` is fixed to zero and is not assigned to a non zero value. Hence, the usage of it will have no impact on the transactions.

```
uint256 private _taxFee = 0;
```

```
require(amount > 0, "Transfer amount must be greater than zero");

_taxFee = 0;
```

```
if (
    (_isExcludedFromFee[from] || _isExcludedFromFee[to]) ||
    (from != uniswapV2Pair && to != uniswapV2Pair)
) {
    _taxFee = 0;
```

```
(uint256 tTransferAmount, uint256 tFee, uint256 tTeam) = _getTValues(
            tAmount,
            _taxFee,
            _liquidityFee
        );
```

Recommendation

Rewrite some code segments so the runtime will be more performant. Try to remove the declaration and usage of the `_taxFee` variable.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L285,299,465,477,496,521,814,821

Description

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

```
excludeFromFees
toggleSwap
transferFrom
approve
allowance
transfer
transferOwnership
renounceOwnership
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L350,346,348

Description

Constant state variables should be declared constant to save gas.

```
_marketingAddress
_devAddress
_autoLiquidityReceiver
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L62,814,342,343,344

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.

```
_decimals
_symbol
_name
_feeSwap
WETH
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L330

Description

There are segments that contain unused state variables.

```
_tOwned
```

Recommendation

Remove unused state variables.

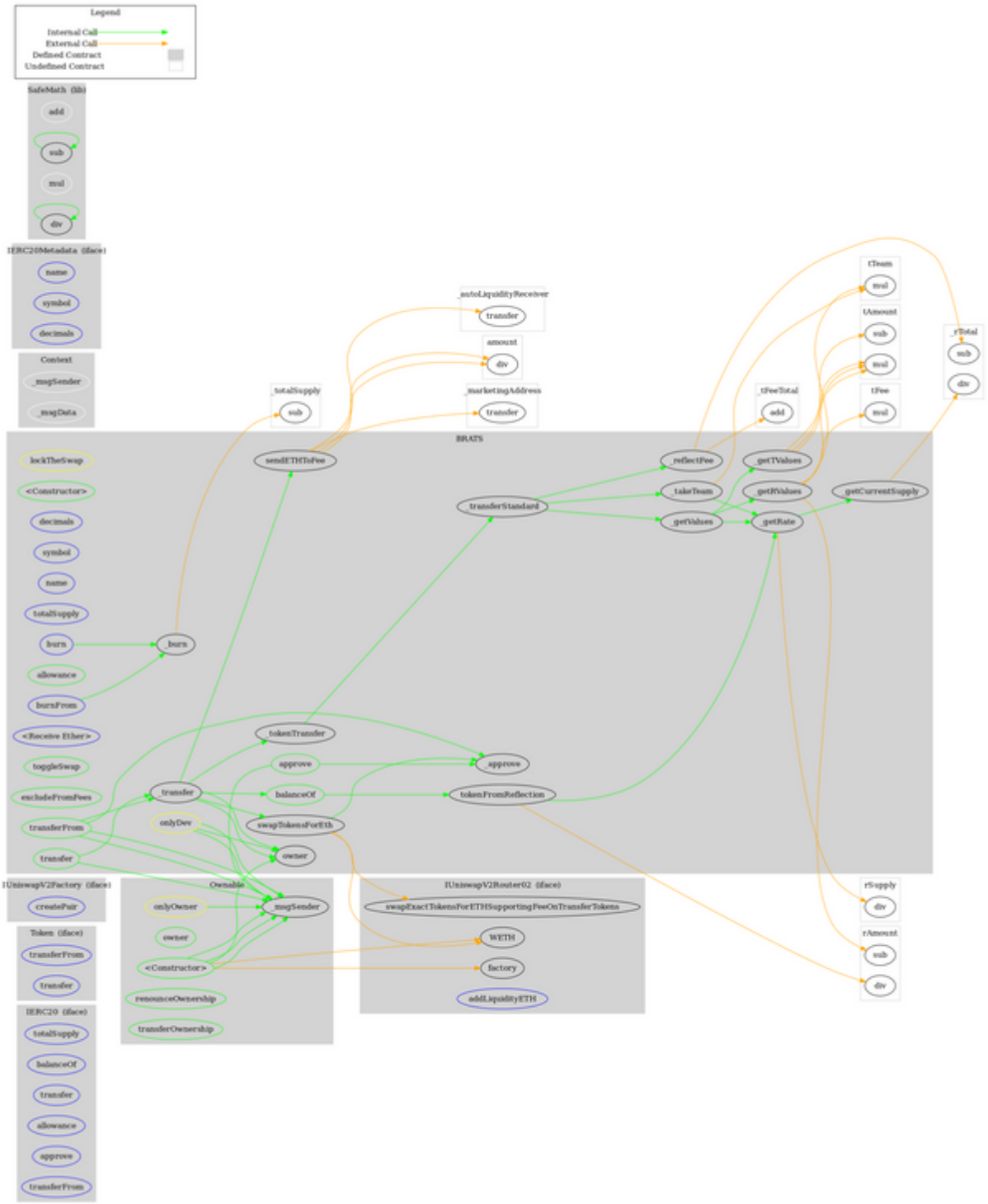
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Token	Interface			
	transferFrom	External	✓	-
	transfer	External	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router02	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20Metadata	Interface	IERC20		

	name	External		-
	symbol	External		-
	decimals	External		-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
BRATS	Implementation	Context, IERC20, Ownable, IERC20Metadata		
	<Constructor>	Public	✓	-
	decimals	External		-
	symbol	External		-
	name	External		-
	totalSupply	External		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	tokenFromReflection	Private		
	_approve	Private	✓	
	_transfer	Private	✓	

	swapTokensForEth	Private	✓	lockTheSwap
	sendETHToFee	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_takeTeam	Private	✓	
	_reflectFee	Private	✓	
	<Receive Ether>	External	Payable	-
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	toggleSwap	Public	✓	onlyDev
	excludeFromFees	Public	✓	onlyDev
	burn	External	✓	-
	_burn	Internal	✓	
	burnFrom	External	✓	-

Contract Flow



Domain Info

Domain Name	babyratscoin.org
Registry Domain ID	6a1052f89f164b4c8bb0866f38e10de5-LROR
Creation Date	2022-05-10T13:29:03Z
Updated Date	2022-05-10T13:56:47Z
Registry Expiry Date	2023-05-10T13:29:03Z
Registrar WHOIS Server	whois.porkbun.com
Registrar URL	https://porkbun.com
Registrar	Porkbun, LLC
Registrar IANA ID	1861

The domain has been created 23 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

BABY RATSCOIN 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. The fees are fixed to 4% and can not be changed.

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