Cyberscope

Audit Report

Baltic Miners

May 2022

Type	BEP20
Network	BSC
Address	0x2df5f68db45ad4f153ce2b1447589d8c17d1306e
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BMI
Compiler Version	v0.7.4+commit.3f05b770
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x2df5f68db45ad4f153ce2b1447589d8c17d1306e
Symbol	BMI
Decimals	5
Total Supply	3,539,520
Domain	balticminers.com

Source Files

Filename	**SHA256**
contract.sol	f4bbfa01900e3c3d2fd4817c0708b4feecc937f37f46935802e4b14c47c18697

Audit Updates

Initial Audit	16th May 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

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L589,560

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `wipeBlacklistAddress` function.

```
  require(!blacklist[sender] && !blacklist[recipient], "in_blacklist");
```

The contract owner can also add specific users to a whalelis. Users that have been added in that list can not trade more tokens than the variable set as buySellPercentForWhale. The owner can set that variable to a very small number and prevent them from selling their tokens.

```
        if ((whalelist[sender] || whalelist[recipient]) &&  (pair == sender ||
pair == recipient)){

require(_totalSupply.mul(buySellPercentForWhale).div(10000)>amount);
        }
```

Recommendation

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.

BT - Burn Tokens

Criticality	critical
Location	contract.sol#L872

Description

The contract owner has the authority to burn the balance of the user's tokens. The owner may take advantage of it by calling the `wipeBlacklistAddress` function.

```
    function wipeBlacklistAddress(address _addr) public onlyOwner {
        require(blacklist[_addr], "address is not blacklist");
        uint256 _balance = _gonBalances[_addr];
        _gonBalances[DEAD].add(_balance);
        _gonBalances[_addr] = 0;
        emit BlacklistAddressWiped(_addr);
        emit Transfer(_addr, address(0x0), _balance);
    }
```

Recommendation

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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L740

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `withdrawAllToTreasury` function.

```
    function withdrawAllToTreasury(address treasury) external swapping
onlyOwner {         uint256 amountToSwap =
_gonBalances[address(this)].div(_gonsPerFragment);
        require( amountToSwap > 0,"There is no gen token deposited in token
contract");
        address[] memory path = new address[](2);
        path[0] = address(this);
        path[1] = router.WETH();
        router.swapExactTokensForETHSupportingFeeOnTransferTokens(
            amountToSwap,
            0,
            path,
            treasury,
            block.timestamp
        );}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	MTS	Manipulate Total Supply
●	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
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope

MTS - Manipulate Total Supply

Criticality	minor
Location	contract.sol#L533

Description

Owner is able to manipulate total supply. This change will have a direct impact on the token price and Market Cap

```
    for (uint256 i = 0; i < times; i++) {
              _totalSupply = _totalSupply
                  .mul((10**RATE_DECIMALS).add(rebaseRate))
                  .div(10**RATE_DECIMALS);
         }
```

Recommendation

The contract owner should carefully manage the adjustment of the circulating supply (increases or decreases), according to the token’s price fluctuations.

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1

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 last else if statements will never be executed as the conditions are met in one of the first 2 blocks every time.

```
if (deltaTimeFromInit < (365 days)) {
            rebaseRate = 140452;
        } else if (deltaTimeFromInit >= (365 days)) {
            rebaseRate = 2110;
        } else if (deltaTimeFromInit >= ((15 * 365 days) / 10)) {
            rebaseRate = 140;
        } else if (deltaTimeFromInit >= (7 * 365 days)) {
            rebaseRate = 20;
        }
```

Recommendation

Remove the last 2 else if blocks.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L345,358,363,389,393,397,872,902,929

Description

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

```
setPairAddress
getLiquidityBacking
wipeBlacklistAddress
decimals
symbol
name
transferOwnership
renounceOwnership
owner
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L439,440,411,409,410,437,432,430,428,431,447,429

Description

Constant state variables should be declared constant to save gas.

```
treasuryFee
swapEnabled
sellFee
liquidityFee
genInsuranceFundFee
firePitFee
feeDenominator
_symbol
_name
...
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L147,148,165,185,789,798,861,872,891,892,893,894,912,916,920,924,929,933,409,410,411,414,439,440,462,463,464,465,466,467

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.

```
_totalSupply
_lastAddLiquidityTime
_lastRebasedTime
_initRebaseStartTime
_autoAddLiquidity
_autoRebase
ZERO
DEAD
_isFeeExempt
...
```

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#L13

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L945,41

Description

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

```
abs
isContract
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L514,627,902

Description

Performing divisions before multiplications may cause lose of prediction.

```
liquidityBalance = _gonBalances[pair].div(_gonsPerFragment)
_gonBalances[autoLiquidityReceiver] =
_gonBalances[autoLiquidityReceiver].add(gonAmount.div(feeDenominator).mul(_liqu
idityFee))
_gonBalances[address(this)] =
_gonBalances[address(this)].add(gonAmount.div(feeDenominator).mul(treasuryFee.a
dd(genInsuranceFundFee)))
_gonBalances[firePit] =
_gonBalances[firePit].add(gonAmount.div(feeDenominator).mul(firePitFee))
feeAmount = gonAmount.div(feeDenominator).mul(_totalFee)
times = deltaTime.div(28800)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L517

Description

The are variables that are defined in the local scope and are not initialized.

```
rebaseRate
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	transferFrom	External	✓	-
IPancakeSwap Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-

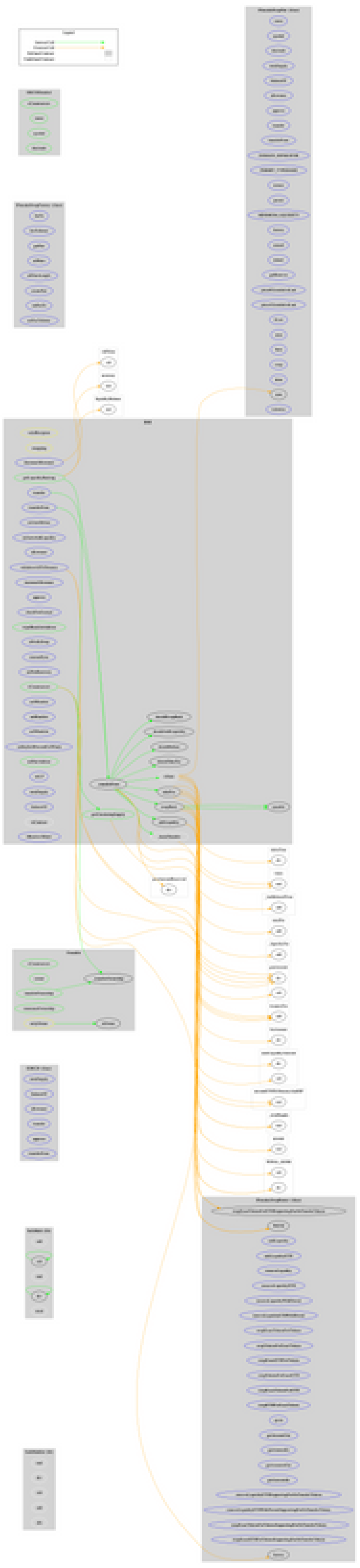
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IPancakeSwap Router	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-

	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IPancakeSwap Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	isOwner	Public		-
	renounceOwnership	Public	✓	onlyOwner

	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20Detailed	Implementation	IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
BMI	Implementation	ERC20Detailed, Ownable		
	<Constructor>	Public	✓	ERC20Detailed
	rebase	Internal	✓	
	transfer	External	✓	validRecipient
	transferFrom	External	✓	validRecipient
	_basicTransfer	Internal	✓	
	_transferFrom	Internal	✓	
	takeFee	Internal	✓	
	addLiquidity	Internal	✓	swapping
	swapBack	Internal	✓	swapping
	withdrawAllToTreasury	External	✓	swapping onlyOwner
	shouldTakeFee	Internal		
	shouldRebase	Internal		
	shouldAddLiquidity	Internal		
	shouldSwapBack	Internal		
	setAutoRebase	External	✓	onlyOwner
	setAutoAddLiquidity	External	✓	onlyOwner
	allowance	External		-
	decreaseAllowance	External	✓	-
	increaseAllowance	External	✓	-
	approve	External	✓	-
	checkFeeExempt	External		-
	getCirculatingSupply	Public		-
	wipeBlacklistAddress	Public	✓	onlyOwner

	isNotInSwap	External		-
	manualSync	External	✓	-
	setFeeReceivers	External	✓	onlyOwner
	getLiquidityBacking	Public		-
	setWhitelist	External	✓	onlyOwner
	setBlacklist	External	✓	onlyOwner
	setWhalelist	External	✓	onlyOwner
	setBuySellPercentForWhale	External	✓	onlyOwner
	setPairAddress	Public	✓	onlyOwner
	setLP	External	✓	onlyOwner
	totalSupply	External		-
	balanceOf	External		-
	isContract	Internal		
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	balticminers.com
Registry Domain ID	2676005082_DOMAIN_COM-VRSN
Creation Date	2022-02-18T13:35:33Z
Updated Date	2022-02-18T13:35:33Z
Registry Expiry Date	2023-02-18T13:35:33Z
Registrar WHOIS Server	whois.1api.net
Registrar URL	http://www.1api.net
Registrar	1API GmbH
Registrar IANA ID	1387

The domain has been created 3 months before the creation of the audit. It will expire in 9 months.

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

Summary

The Smart Contract analysis reported some issues. The contract owner has the authority to blacklist addresses and by doing that he will burn all of their token balances. The owner can also prevent specific users from transactions, and transfer the tokens accumulated by fees from the contract into the team's wallet. 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