Cyberscope

Audit Report

Baltic Miners

July 2022

Type	BEP20
Network	BSC
Address	0xe247d974a7AdCBc097764c6d76C164211e50b3e0
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BalticFinancialToken
Compiler Version	v0.8.15+commit.e14f2714
Optimization	5000 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xe247d974a7AdCBc097764c6d76C164211e50b3e0
Symbol	BMFT
Decimals	5
Total Supply	4,000,000,000
Domain	http://www.balticminers.com/

Source Files

Filename	**SHA256**
contract.sol	bae3013b5f5ca523ec5894012f117e71e6f5cde41153f70527f6f403ed4cf221

Audit Updates

Initial Audit	7th July 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

ST - Stop Transactions

Criticality	medium
Location	contract.sol#L558

Description

The contract owner has the authority to stop the sales for specific users excluding the owner. The owner may take advantage of it by setting the limitPercent to zero. As a result, the sellLimitPerTime will be zero and the expression will always revert.

```
require(amount <= userLimits[from].sellLimitPerTime, "Limited wallet selling above limit.");
```

Recommendation

The contract could embody a check for not allowing setting the limitPercent less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

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#L676

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 `sweepContigency` method.

```
function sweepContingency() external onlyOwner {
        require(!_hasLiqBeenAdded, "Cannot call after liquidity.");
        payable(_owner).transfer(address(this).balance);
 }
```

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L697

Description

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

```
try antiSnipe.checkUser(from, to, amount) returns (bool check) {
    checked = check;
} catch {
     revert();
}

if(!checked) {
     revert();
}
```

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.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
Medium	US	Untrusted Source
Minor	STC	Succeeded Transfer Check
Minor	MTS	Manipulate Total Supply
Minor	L01	Public Function could be Declared External
Minor	L02	State Variables could be Declared Constant
Minor	L04	Conformance to Solidity Naming Conventions
Minor	L05	Unused State Variable
Minor	L07	Missing Events Arithmetic
Minor	L12	Using Variables before Declaration
Minor	L13	Divide before Multiply Operation
Minor	L14	Uninitialized Variables in Local Scope

US - Untrusted Source

Criticality	medium
Location	contract.sol#L697

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result it may produce security issues and harm the transactions.

```
try antiSnipe.checkUser(from, to, amount) returns (bool check) {
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L676

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.

```
function sweepContingency() external onlyOwner {
        require(!_hasLiqBeenAdded, "Cannot call after liquidity.");
        payable(_owner).transfer(address(this).balance);
 }
```

Recommendation

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

MTS - Manipulate Total Supply

Criticality	minor
Location	contract.sol#L580

Description

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

```
uint256 rebaseMinutes = rebaseTimeInMinutes * 1 minutes;
if(autoRebaseEnabled
    && block.timestamp >= autoRebaseLastTriggered + rebaseMinutes
) {
    uint256 deltaTime = block.timestamp - autoRebaseLastTriggered;
    uint256 rebaseAmounts = deltaTime / (rebaseMinutes);
    uint256 epoch = rebaseAmounts * rebaseTimeInMinutes;

    for (uint256 i = 0; i < rebaseAmounts; i++) {
        _tTotal = (_tTotal * ((10**_rateDecimals) + rebaseRate)) / (10**_rateDecimals);
    }
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L658,384,315,320,380

Description

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

```
isExcludedFromLimits
approve
transfer
isExcludedFromFees
enableTrading
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L192,191

Description

Constant state variables should be declared constant to save gas.

```
_maxTxAmount
_maxWalletSize
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L161,33,165,133,141,160,117,114,157,159,149,417,158,116,177,118,195,203

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.

```
_rateDecimals
_hasLiqBeenAdded
_decimals
_taxWallets
_name
_antiSnipe
maxSellTaxes
_antiBlock
_transferTaxes
...
```

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#L192,191

Description

There are segments that contain unused state variables.

```
_maxTxAmount
_maxWalletSize
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L463,457,500,484

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
rebaseRate = rate
limitTime = timeInMinutes * 60
swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor
piSwapPercent = priceImpactSwapPercent
```

Recommendation

Emit an event for critical parameter changes.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L693

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
check
```

Recommendation

The variables should be declared before any usage of them.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L726,532

Description

Performing divisions before multiplications may cause lose of prediction.

```
treasuryAmount = (feeAmount / currentFee.total) * currentFee.treasury
insuranceAmount = (feeAmount / currentFee.total) * currentFee.insurance
firePitAmount = (feeAmount / currentFee.total) * currentFee.firePit
rebaseAmounts = deltaTime / (rebaseMinutes)
feeAmount = (amount / masterTaxDivisor) * currentFee.total
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L693,692

Description

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

```
checked
check
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

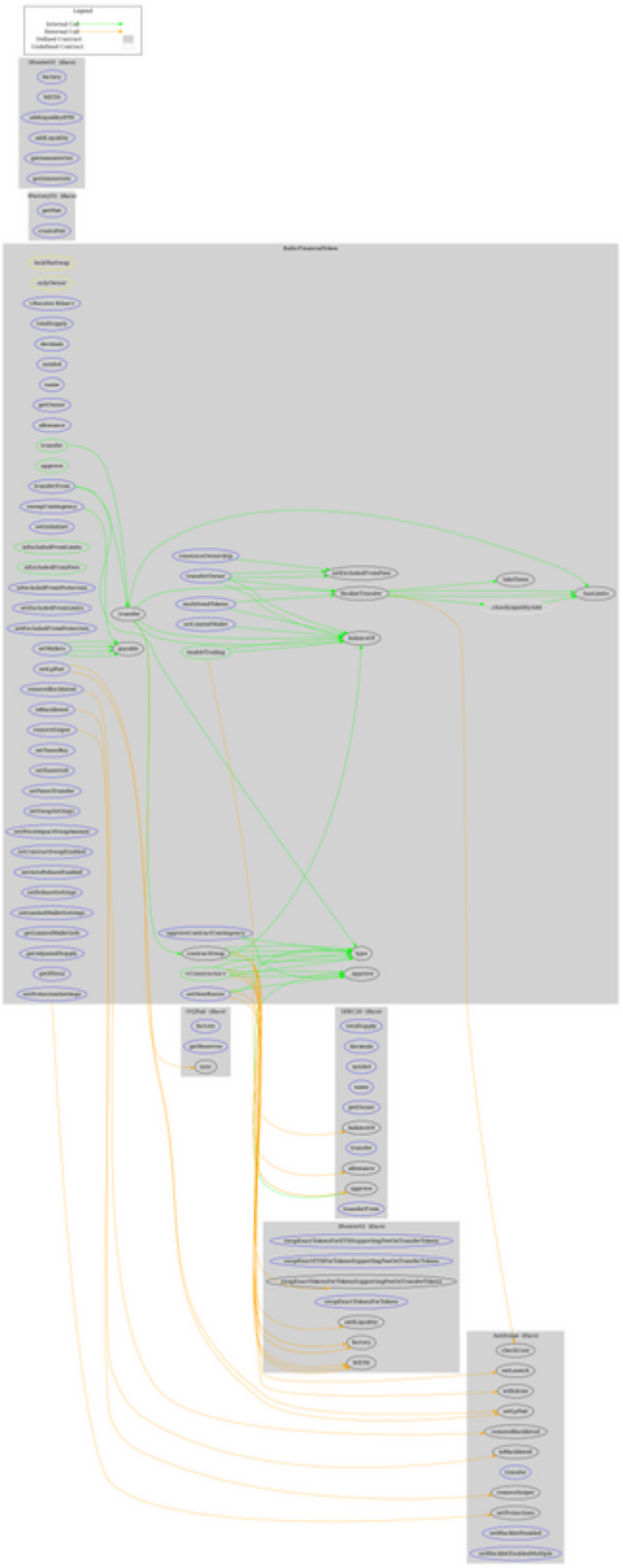
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IFactoryV2	Interface			
	getPair	External		-
	createPair	External	✓	-
IV2Pair	Interface			
	factory	External		-
	getReserves	External		-
	sync	External	✓	-
IRouter01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	addLiquidity	External	✓	-
	getAmountsOut	External		-
	getAmountsIn	External		-

IRouter02	Interface	IRouter01		
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokens	External	✓	-
AntiSnipe	Interface			
	checkUser	External	✓	-
	setLaunch	External	✓	-
	setLpPair	External	✓	-
	setProtections	External	✓	-
	removeSniper	External	✓	-
	removeBlacklisted	External	✓	-
	transfer	External	✓	-
	withdraw	External	✓	-
	isBlacklisted	External		-
	setBlacklistEnabled	External	✓	-
	setBlacklistEnabledMultiple	External	✓	-
BalticFinancialToken	Implementation	IERC20		
	<Constructor>	Public	Payable	-
	<Receive Ether>	External	Payable	-
	transferOwner	External	✓	onlyOwner
	renounceOwnership	External	✓	onlyOwner
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	allowance	External		-
	balanceOf	Public		-

	transfer	Public	✓	-
	approve	Public	✓	-
	_approve	Internal	✓	
	approveContractContingency	External	✓	onlyOwner
	transferFrom	External	✓	-
	setNewRouter	External	✓	onlyOwner
	setLpPair	External	✓	onlyOwner
	setInitializer	External	✓	onlyOwner
	isExcludedFromLimits	Public		-
	isExcludedFromFees	Public		-
	isExcludedFromProtection	External		-
	setExcludedFromFees	Public	✓	onlyOwner
	setExcludedFromLimits	External	✓	onlyOwner
	setExcludedFromProtection	External	✓	onlyOwner
	removeBlacklisted	External	✓	onlyOwner
	isBlacklisted	External		-
	removeSniper	External	✓	onlyOwner
	setProtectionSettings	External	✓	onlyOwner
	setWallets	External	✓	onlyOwner
	setTaxesBuy	External	✓	onlyOwner
	setTaxesSell	External	✓	onlyOwner
	setTaxesTransfer	External	✓	onlyOwner
	setSwapSettings	External	✓	onlyOwner
	setPriceImpactSwapAmount	External	✓	onlyOwner
	setContractSwapEnabled	External	✓	onlyOwner
	setAutoRebaseEnabled	External	✓	onlyOwner
	setRebaseSettings	External	✓	onlyOwner
	setLimitedWallet	External	✓	onlyOwner
	setLimitedWalletSettings	External	✓	onlyOwner
	getLimitedWalletInfo	External		-
	getAdjustedSupply	External		-
	getRTotal	External		-
	_hasLimits	Internal		
	_transfer	Internal	✓	

	contractSwap	Internal	✓	lockTheSwap
	_checkLiquidityAdd	Internal	✓	
	enableTrading	Public	✓	onlyOwner
	sweepContingency	External	✓	onlyOwner
	multiSendTokens	External	✓	onlyOwner
	finalizeTransfer	Internal	✓	
	takeTaxes	Internal	✓	

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 in 8 months before the creation of the audit.

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 transferring funds to the team’s wallet, blacklisting addresses and preventing specific users from selling. In addition, an untrusted source is used for core functionalities on the contract implementation. 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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io