Cyberscope

Audit Report

Multi Finance Protocol

May 2022

Type	BEP20
Network	BSC
Address	0xA2d12A33Cff3131A1aC8fC5023E2AA17436f3c96
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MULTIFI
Compiler Version	v0.8.0+commit.c7dfd78e
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0xA2d12A33Cff3131A1aC8fC5023E2AA17436f3c96
Symbol	MLM
Decimals	10
Total Supply	1,400,000
Domain	multifinance.io

Source Files

Filename	**SHA256**
contract.sol	16cfd3386a1a172cb18ea51abbd9d92a6e17593274af29bacb3f372bbd023218

Audit Updates

Initial Audit	23rd 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

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
Medium	BLC	Business Logic Concern
Minor	MTS	Manipulate Total Supply
Minor	CO	Code Optimization
Medium	MC	Missing Check

BLC - Business Logic Concern

Criticality	medium
Location	contract.sol#L644, 869, 793, 804

Description

The business logic seems peculiar. The implementation may not follow the expected behavior. The contract allows all users to modify the stepRerferelAmout variable. Also, there is no check for the maximum value that can be set.

```
function setStepReferelAmount(uint256 amount) public {
   require(amount > 0);
   stepReferelAmount = amount;
}
```

```
function tokenStepPrice() public view returns (uint256) {
        (uint256 a0, uint256 a1, ) = pairContract.getReserves();
        uint256 price = 0;
        if(pairContract.token0() == address(this)) {
            price = a1 != 0 ? a0.mul(stepReferelAmount).div(a1) : 0;
        } else {
            price = a0 != 0 ? a1.mul(stepReferelAmount).div(a0) : 0;
        }
        return price;
    }
```

```
function reCheckLv(address _address, bool isBuyer) internal {
        uint256 price = tokenStepPrice();
```

```
reCheckLv(sender, false);
reCheckLv(recipient, true);
```

Recommendation

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

MTS - Manipulate Total Supply

Criticality	minor
Location	contract.sol#L662

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);
        }
_gonsPerFragment = TOTAL_GONS.div(_totalSupply);
```

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#L709

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.

```
bool flag = true;
address _address = _referel;
for (uint256 i = 0; i < 15; i++) {
    if (referelAddress[_address] == address(0)) {
        break;
    } else if (referelAddress[_address] == msg.sender) {
        flag = false;
        break;
    }
}
require(flag);
```

Recommendation

Rewrite some code segments so the runtime will be more performant. Remove the for loop and leave only the second else if case.

MC - Missing Check

Criticality	medium
Location	contract.sol#L843

Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues. In the following code segment, if there are more than 2 iterations in the for loop, the subtraction for _referelGon will overflow. Hence, the transaction will be reverted.

```
uint256 _halfReferelGon = _referelGon.div(2);
while (
  _referelAddress != address(0) &&
  _lv >= count &&
  _halfReferelGon > 0
) {
  referelBalance[_referelAddress] = referelBalance[
    _referelAddress
  ].add(_halfReferelGon);
  count = count.add(1);
  _referelAddress = referelAddress[_referelAddress];
  _lv = referelLv[_referelAddress];
  _referelGon = _referelGon.sub(_halfReferelGon);
}
```

Recommendation

The contract should properly check the variables according to the required specifications. The contract could embody a check for not allowing the value of _referelGon to be less than zero.

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	✓	-
IPancakeSwapPair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-

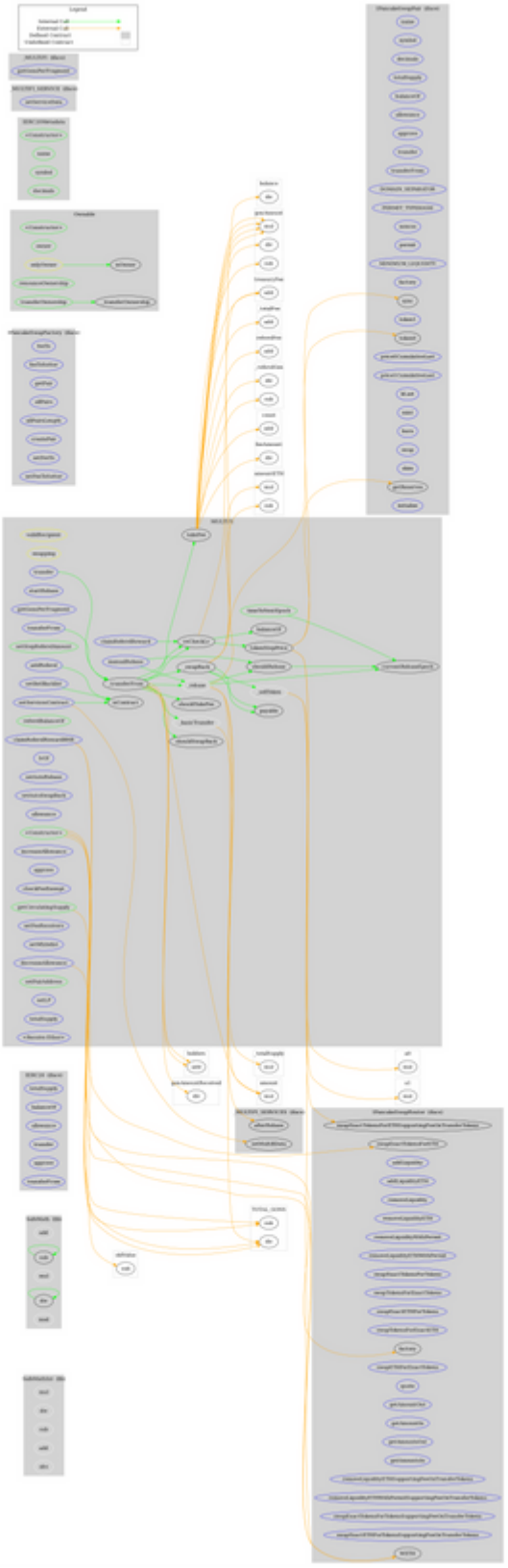
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	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	✓	-
IPancakeSwapFactory	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	✓	
IERC20Metadata	Implementation	IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
_MULTIFI_SERVICES	Interface			
	setMultifiData	External	✓	-
	afterRebase	External	✓	-
_MULTIFI_SERVICE	Interface			
	setServiceData	External	✓	-
_MULTIFI	Interface			
	getGonsPerFragment	External		-
MULTIFI	Implementation	IERC20Metadata, Ownable, _MULTIFI		
	<Constructor>	Public	✓	IERC20Metadata Ownable
	startRebase	External	✓	onlyOwner
	getGonsPerFragment	External		-
	setServicesContract	External	✓	onlyOwner
	setStepReferelAmount	Public	✓	-
	_rebase	Internal	✓	
	currentRebaseEpoch	Public		-
	timeToNextEpoch	Public		-
	shouldRebase	Internal		

	transfer	External	✓	validRecipient
	manualRebase	External	✓	-
	transferFrom	External	✓	validRecipient
	_basicTransfer	Internal	✓	
	addReferel	External	✓	-
	referelBalanceOf	Public		-
	claimReferelRewardBNB	External	✓	-
	lvOf	External		-
	claimReferelReward	External	✓	-
	_transferFrom	Internal	✓	
	reCheckLv	Internal	✓	
	takeFee	Internal	✓	
	tokenStepPrice	Public		-
	_sellToken	Internal	✓	
	_swapBack	Internal	✓	swapping
	shouldTakeFee	Internal		
	shouldSwapBack	Internal		
	setAutoRebase	External	✓	onlyOwner
	setAutoSwapBack	External	✓	onlyOwner
	allowance	External		-
	decreaseAllowance	External	✓	-
	increaseAllowance	External	✓	-
	approve	External	✓	-
	checkFeeExempt	External		-
	getCirculatingSupply	Public		-
	setFeeReceivers	External	✓	onlyOwner
	setWhitelist	External	✓	onlyOwner
	setBotBlacklist	External	✓	onlyOwner
	setPairAddress	Public	✓	onlyOwner
	setLP	External	✓	onlyOwner
	totalSupply	External		-
	balanceOf	Public		-
	isContract	Internal		
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	multifinance.io
Registry Domain ID	806351f00867412995d671decfa6d6da-DONUTS
Creation Date	2022-05-17T14:21:20Z
Updated Date	2022-05-22T14:21:46Z
Registry Expiry Date	2023-05-17T14:21:20Z
Registrar WHOIS Server	whois.namesilo.com
Registrar URL	http://www.namesilo.com
Registrar	NameSilo, LLC
Registrar IANA ID	1479

The domain has been created 6 days before the creation of the audit. It will expire in 12 months.

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

Summary

There is a function that can be called by all users and affect the contract’s transfer flow. The contract has some missing checks that could disturb users’ transactions. There is also a limit of max 20% fees.

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