Cyberscope

Audit Report

MC Games

May 2022

Type	BEP20
Network	BSC
Address	0xc2a18c458f17f93e27e8c63113d691f4b5725797
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MCG
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0xc2a18c458f17f93e27e8c63113d691f4b5725797
Symbol	MC Games
Decimals	9
Total Supply	10,000,000,000
Domain	mcgames.app

Source Files

Filename	**SHA256**
contract.sol	df740b0bbc47a033f1cc4d541e627216b308822d04e2787bfcf7f6aaee0a3802

Audit Updates

Initial Audit	25th 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

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L1960,2014

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `maxSellTransactionAmount` to a low value like 1. This will essentially prevent the users from selling.

```
if (
    maxSellTransactionAmount != 0 &&
    isSelling && // sells only by detecting transfer to automated market maker
pair
    from != address(uniswapV2Router) //router -> pair is removing liquidity
which shouldn't have max
) {
    require(
        amount <= maxSellTransactionAmount,
        "maxSellTransactionAmount."
    );
}
```

The contract owner has the authority to tax only the sales with an additional 50% fee. The owner may take advantage of it by setting the `MCGRewardsSellFee` to 50.

```
if (isSelling) {
    MCGRewardsFee = MCGRewardsSellFee;
} else if(to == stakingAddress) {
    MCGRewardsFee = 0;
    operationsFees = 0;
} else {
    MCGRewardsFee = MCGRewardsBuyFee;
}

uint256 totalFees = MCGRewardsFee.add(liquidityFee).add(operationsFees);
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount 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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1774

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `updateFees` function with a high 20, 20, 50, 50. As a result the total taxes will be increased to 90%.

```
function updateFees(
    uint256 operations,
    uint256 liquidity,
    uint256 MCGRewardsBuy,
    uint256 MCGRewardsSell
) public onlyOwner {
    require(operations <= 20, "operation fees can't exceed 20%");
    require(liquidity <= 20, "liquidity fees can't exceed 20%");
    require(
        MCGRewardsBuy >= 1 && MCGRewardsBuy <= 50,
        "MCG reward fees must be between 1 and 50"
    );
    require(
        MCGRewardsSell >= 1 && MCGRewardsSell <= 50,
        "MCG reward fees must be between 1 and 50"
    );

    operationsFees = operations;
    liquidityFee = liquidity;
    MCGRewardsBuyFee = MCGRewardsBuy;
    MCGRewardsSellFee = MCGRewardsSell;

    totalSellFees = MCGRewardsSellFee.add(liquidityFee).add(operationsFees);
    totalBuyFees = MCGRewardsBuyFee.add(liquidityFee).add(operationsFees);

    emit UpdateFees(operations, liquidity, MCGRewardsBuy, MCGRewardsSell);
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

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
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L12	Using Variables before Declaration
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope
●	L15	Local Scope Variable Shadowing

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L386,394,411,443,456,473,496,525,552,938,947,1301,1339,1358,1606,1612,1632,1639,1658,1669,1678,1697,1704,1711,1725,1732,1741,1745,1759,1768,1774,1814,1818,1826,2096,2104,2108,2115,2134,2365,2417,2503,2507,2518,2522,2526

Description

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

```
getAllowTokens
updateAllowTokens
getPayoutToken
updatePayoutToken
updateUniswapV2Router
process
getAccountAtIndex
forceSwapAndSendDividends
dividendTokenBalanceOf
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L79,81,112,1027,1339,1346,1358,1372,1242,1606,1777,1778,1477,1478,2324

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.

```
_account
MCGRewardsSellFee
MCGRewardsBuyFee
MCGRewardsSell
MCGRewardsBuy
_contractAddress
UpdatestakeAddress
magnitude
_owner
...
```

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#L959

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1658,1669,2134

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.

```
buyAmount = buyAmount.sub(fromBuy)
swapTokensAtAmount = newNum * (10 ** 9)
maxSellTransactionAmount = newNum * (10 ** 9)
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L1391,1307,194,1005

Description

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

```
abs
get
_withdrawDividendOfUser
_transfer
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L2026,2025,2027

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.

```
lastProcessedIndex
iterations
claims
```

Recommendation

The variables should be declared before any usage of them.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1916,2145

Description

Performing divisions before multiplications may cause lose of prediction.

```
dividendsFromSell =
address(this).balance.mul(sellAmount).div(totalAmount).mul(MCGRewardsSellFee).di
v(MCGRewardsSellFee.add(operationsFees))
dividendsFromBuy =
address(this).balance.mul(buyAmount).div(totalAmount).mul(MCGRewardsBuyFee).div(
MCGRewardsBuyFee.add(operationsFees))
swapAmountSold = contractTokenBalance.mul(sellAmount).div(totalBuySell)
swapAmountBought = contractTokenBalance.mul(buyAmount).div(totalBuySell)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L2536,2599,2027,2025,2026

Description

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

```
claims
iterations
lastProcessedIndex
success
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L1262

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_symbol
_name
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	**Type**	**Bases**		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IUniswapV2Pair	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	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IterableMapping	Library			
	get	Internal		
	getIndexOfKey	Internal		
	getKeyAtIndex	Internal		
	size	Internal		
	set	Internal	✓	
	remove	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-

	symbol	External		-
	decimals	External		-
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-

	withdrawDividend	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-

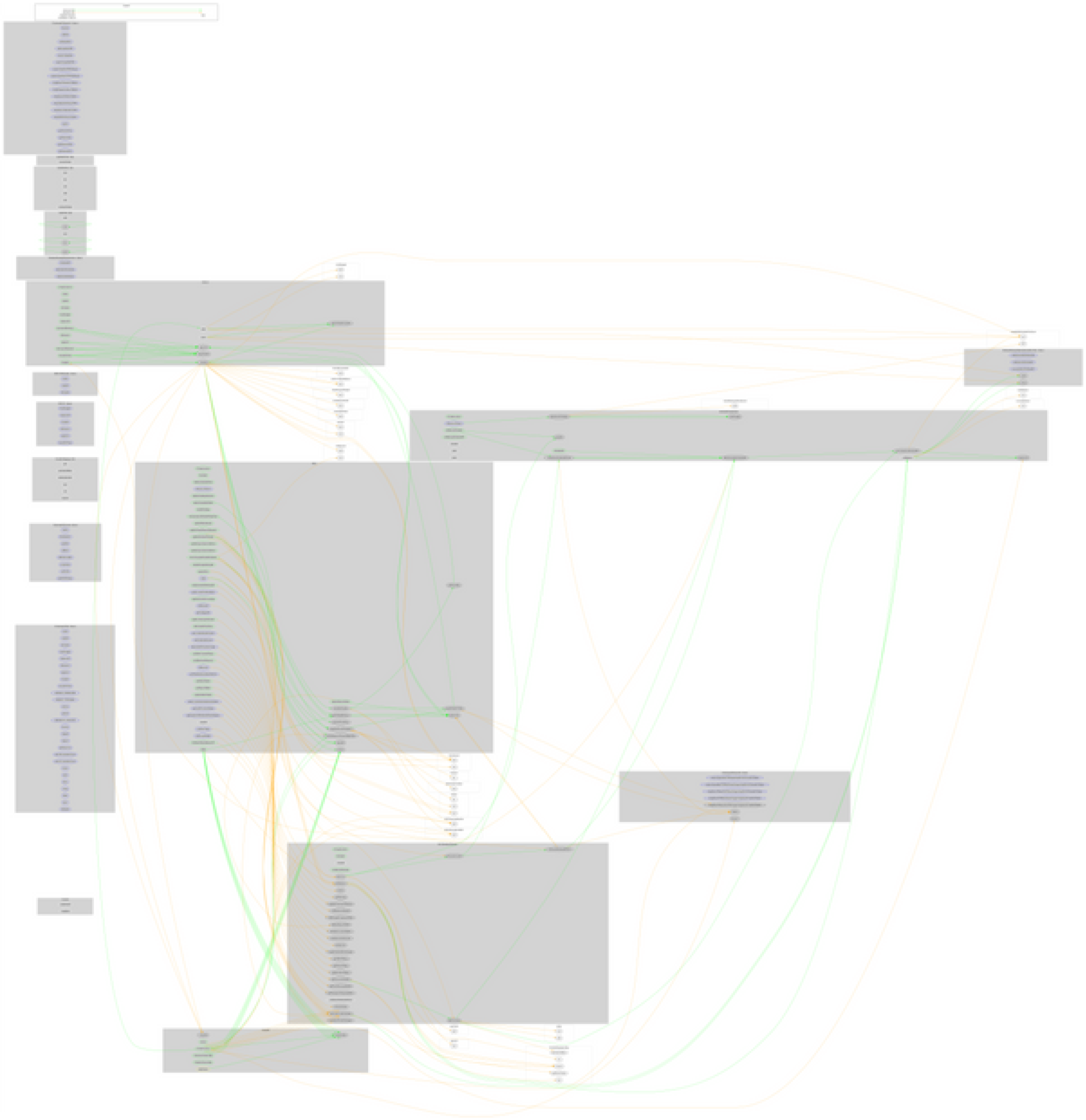
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-

	distributeDividends	Public	Payable	-
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
MCG	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	decimals	Public		-
	UpdatestakeAddress	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	updateStakingAmounts	Public	✓	onlyOwner
	setPresaleWallet	External	✓	onlyOwner
	enableStaking	Public	✓	onlyOwner
	stake	Public	✓	-
	updateMaxAmount	Public	✓	onlyOwner
	updateSwapTokensAtAmount	Public	✓	onlyOwner
	updateDividendTracker	Public	✓	onlyOwner
	updateOperations1Address	Public	✓	onlyOwner
	updateOperations2Address	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	enableSwapAndLiquify	Public	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	setAllowCustomTokens	Public	✓	onlyOwner
	setAllowAutoReinvest	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateLiquidityWallet	Public	✓	onlyOwner

	updateGasForProcessing	Public	✓	onlyOwner
	updateFees	Public	✓	onlyOwner
	getStakingInfo	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	setAutoClaim	External	✓	-
	setReinvest	External	✓	-
	setDividendsPaused	External	✓	onlyOwner
	isExcludedFromAutoClaim	External		-
	isReinvest	External		-
	_transfer	Internal	✓	
	getStakingBalance	Private		
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	updatePayoutToken	Public	✓	-
	getPayoutToken	Public		-
	updateAllowTokens	Public	✓	onlyOwner
	getAllowTokens	Public		-
	addLiquidity	Private	✓	
	forceSwapAndSendDividends	Public	✓	onlyOwner
	swapAndSendDividends	Private	✓	
MCGDividend Tracker	Implementation	DividendPayingToken, Ownable		
	<Constructor>	Public	✓	DividendPayingToken
	decimals	Public		-

	_transfer	Internal		
	withdrawDividend	Public		-
	isExcludedFromAutoClaim	External		onlyOwner
	isReinvest	External		onlyOwner
	setAllowCustomTokens	External	✓	onlyOwner
	setAllowAutoReinvest	External	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	setAutoClaim	External	✓	onlyOwner
	setReinvest	External	✓	onlyOwner
	setDividendsPaused	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	updatePayoutToken	Public	✓	onlyOwner
	getPayoutToken	Public		-
	updateAllowTokens	Public	✓	onlyOwner
	getAllowTokens	Public		-
	_reinvestDividendOfUser	Private	✓	
	_withdrawDividendOfUser	Internal	✓	

Contract Flow



Domain Info

Domain Name	mcgames.app
Registry Domain ID	48BEFB4E4-APP
Creation Date	2022-04-20T22:12:48Z
Updated Date	2022-04-25T22:12:48Z
Registry Expiry Date	2023-04-20T22:12:48Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	https://www.godaddy.com/
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created about 1 month 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

There are some functions that can be abused by the owner like stopping transactions and manipulating fees. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

The contract offers a staking option. When the users stake their tokens, the dividend tracker is sharing a bonus amount. During the staking period, the users can buy but cannot sale their holdings.

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