COINSCOPE

Audit Report

Metacore nft

January 2022

Type	BEP20
Network	BSC
Address	0x8511Dc91Af5315f65a011D84F2F452c2bc7E811a
Audited by	© coinscope

Table of Contents

Table of Contents 1

Contract Review 3

Audit Updates 3

Contract Analysis 4

ST - Stop Transactions 5

Description 5

Recommendation 5

ELFM - Exceed Limit Fees Manipulation 6

Description 6

Recommendation 6

ULTW - Unlimited Liquidity to Team Wallet 7

Description 7

Recommendation 7

Contract Diagnostics 8

L01 - Public Function could be Declared External 9

Description 9

Recommendation 9

L02 - State Variables could be Declared Constant 10

Description 10

Recommendation 10

L05 - Unused State Variable 11

Description 11

Recommendation 11

L04 - Conformance to Solidity Naming Conventions 12

Description 12

Recommendation 12

L09 - Dead Code Elimination 13

Description 13

Recommendation 13

L11 - Unnecessary Boolean equality 14

Description 14

Recommendation 14

L07 - Missing Events Arithmetic 15

Description 15

Recommendation 15

Contract Functions 16

Contract Flow 18

Domain Info 19

Summary 20

Disclaimer 21

About Coinscope 22

Contract Review

Contract Name	MetaCore
Compiler Version	v0.8.5+commit.a4f2e591
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x8511Dc91Af5315f65a011D84F2F452c2bc7E811a
Symbol	MCORE
Decimals	9
Total Supply	1,000,000,000
Source	contract.sol
Domain	metacorenft.io

Audit Updates

Initial Audit	31st January 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#L1190

Description

The contract owner has the authority to stop sales or buys for all users excluding the owner. The owner may take advantage of it by setting the `buyLimit` or `sellLimit` to zero.

```
if (isBuyOrder(from, to)) require(amount < buyLimit, "Try a smaller amount");
if (isSellOrder(from, to)) require(amount < sellLimit, "Try a smaller amount");
```

Recommendation

The contract could embody a check for not allowing setting the buyLimit and sellLimit 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#L790,793,796,799,802,805

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 `setMarketingFee` function with a high percentage value.

```
function setMarketingFee(uint fee) public onlyOwner {
    marketingFee = fee;
}
```

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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L1

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been swapped from the swap & liquify feature. The owner may take advantage of it by:

1. Disable the *liquifyEnabled*
2. Disable the *buyBackEnabled*
3. When the contract has accumulated many funds, call the *withdraw()*

```
if (liquifyEnabled && amountToLiquify > 0)  {
    swapAndLiquify(amountToLiquify, false);
}
// swap for buyback later
uint256 rest = contractTokenBalance.sub(amountToLiquify);
uint256 buyBack = rest.mul(buyBackFee).div(marketingFee+buyBackFee);
swapTokensForEth(buyBack);
```

```
function withdraw (address payable to, uint256 amount) public onlyOwner() {
    transferToAddressETH(to, amount);
}
```

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
●	L02	State Variables could be Declared Constant
●	L05	Unused State Variable
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L11	Unnecessary Boolean equality
●	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L1177,L1170,L1166 and 39 more

Description

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

```
setDailySellLimit
setBuyLimit
withdraw
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L450,L433,L432 and 1 more

Description

Constant state variables should be declared constant to save gas.

```
maxTimeStakeCluster
_symbol
_name
...
```

Recommendation

Add the constant attribute to state variables that never change.

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L490,L475

Description

There are segments that contains unused state variable.

```
dailySales
stakeStartTime
```

Recommendation

Remove unused state variables.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L469,L468,L467 and 20 more

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.

```
_previousBuyBackFee
_previousMarketingFee
_previousLiquifyFeeBuy
...
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L79,L75,L10 and 7 more

Description

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

```
mod
_msgData
sendValue
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L767

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
devWallets[to] == true
```

Recommendation

Remove the equality to the boolean constant.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1174,L1170,L1122 and 13 more

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.

```
sellLimit = amountWithoutDecimals * 10 ** _decimals
buyLimit = amountWithoutDecimals * 10 ** _decimals
buyBackUpperLimit = buyBackLimit * 10 ** 18
...
```

Recommendation

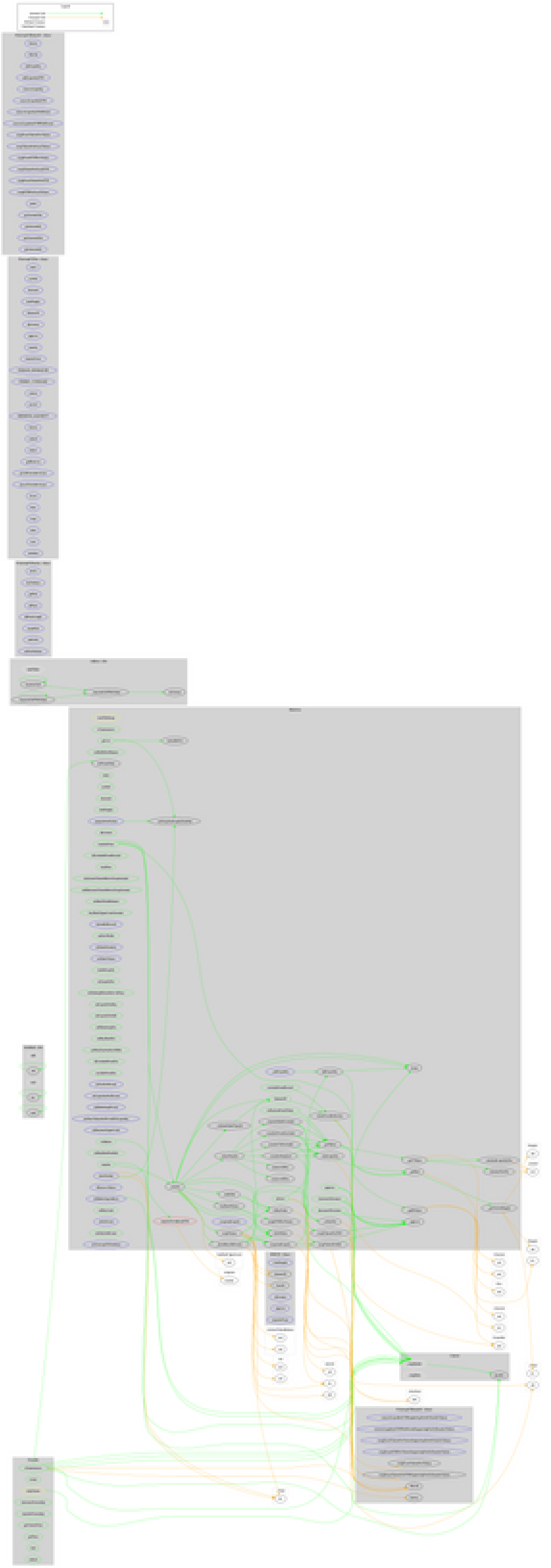
Emit an event for critical parameter changes.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		

	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
BaseToken	Implementation			
StandardToken	Implementation	IERC20, Ownable, BaseToken		
	<Constructor>	Public	Payable	-
	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	✓	
	_setupDecimals	Internal	✓	
	_beforeTokenTransfer	Internal	✓	

Contract Flow



Domain Info

Domain Name	metacorenft.io
Registry Domain ID	d3dc392a3ed64ec187e9deb0fb45725d-DONUTS
Creation Date	2022-01-26T15:04:21Z
Updated Date	2022-01-31T15:04:29Z
Registry Expiry Date	2023-01-26T15:04:21Z
Registrar WHOIS Server	whois.dynadot.com
Registrar URL	http://dynadot.com
Registrar	Dynadot, LLC
Registrar IANA ID	472

The domain has been created 5 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

Metacore is aiming to build an NFT-based game that is part of the play-to-earn metaverse. The Smart Contract analysis reported some issues. There are some functions that can be abused by the owner, like manipulating fees, transferring funds to the team's wallet and stopping transactions. The contract could operate as a honeypot if the contract owner abuses the configuratio. The contract contains an anti-bot system that is limited to a specific time period. It can prevent users from selling for a maximum of 10 hours. 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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.

The Coinscope.co team

https://www.coinscope.co