Cyberscope

Audit Report

Dark Matter Metaverse

June 2022

Type	BEP20
Network	BSC
Address	0xb16870d072396f0069d9a9c98cd5d2b25db05ea9
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	DARKMATTER
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0xb16870d072396f0069d9a9c98cd5d2b25db05ea9
Symbol	DMMT
Decimals	9
Total Supply	500,000,000
Domain	https://spaceheroes.world/

Source Files

Filename	**SHA256**
contract.sol	4e1819136b17c3663f94902653f192daf810571e4979ca335c0ed5917a62b7f6

Audit Updates

Initial Audit	25th June 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#L1374,L11730,L1424

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the _minBuyTxAmount to minimum amount. By setting the _minBuyTxAmount to 1 it makes the buying transaction almost impossible.

```
require(
        amount >= _minBuyTxAmount,
        "Buy transfer amount must exceeds the minBuyTxAmount."
);
```

The owner can convert the contract into a honeypot and prevent users from selling by setting coolDownTimestamp to the maximum amount.

```
require(
        coolDownTimestamp < block.timestamp,
        "Wait till one hour Cool Down Pd"
);
```

The owner can also stop transactions by setting _liquidityFee and _marketingFee to zero. The sum for these values is combineFees which is the denominator of tokenForLiquidity. When combineFees is zero the transaction will be reverted.

```
combineFees = _liquidityFee.add(_marketingFee);
uint256 tokensForLiquidity = contractTokenBalance
            .mul(_liquidityFee)
            .div(combineFees);
```

Recommendation

The contract could embody a check for not allowing setting the _maxSellTxAmount and _minBuyTxAmount 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#L1656

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 setTaxFeePercent function with a high percentage value.

```
function setFees(
        uint256 taxFee,
        uint256 liquidityFee,
        uint256 marketingFee,
        uint256 burnFee
    ) external onlyOwner {
        _taxFee = taxFee;
        _liquidityFee = liquidityFee;
        _marketingFee = marketingFee;
        _burnFee = burnFee;
    }
```

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1365

Description

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

```
require(
            !isantiBotlisted[from] && !isantiBotlisted[to],
            "antiBotlisted address"
);
```

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
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L1077,995,1699,1085,1048,497,1338,987,991,983,1013,1644,1031,506,1004,515,1022,1640,1081,528,1097,1125,520,1061

Description

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

```
decreaseAllowance
lock
excludeFromReward
reflectionFromToken
unlock
totalFees
excludeFromFee
approve
geUnlockTime
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L925,909,915,914,913

Description

Constant state variables should be declared constant to save gas.

```
_name
_symbol
_decimals
_tTotal
deadAddress
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1308,923,608,942,638,1300,941,931,606,1699,1735,917,920,927,684

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.

```
WETH
_marketingFee
_liquidityFee
_taxFee
_seconds
_enabled
DOMAIN_SEPARATOR
BUSD
_minBuyTxAmount
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1671,1652,1735,1678,1683,1691

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.

```
_maxSellTxAmount = maxSellTxAmount * (10 ** 9)
_minBuyTxAmount = minBuyTxAmount * (10 ** 9)
maxWalletToken = maxWalletToken_ * (10 ** 9)
sellCooldownperiod = _seconds
_taxFee = taxFee
numTokensSellToAddToLiquidity = newAmt * (10 ** 9)
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L316,380,400,413,287,361,348

Description

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

```
functionCall
isContract
_functionCallWithValue
functionCallWithValue
sendValue
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1562,1580

Description

Performing divisions before multiplications may cause lose of prediction.

```
tBurn = tAmount.div(100).mul(_burnFee)
tMarketing = tAmount.div(100).mul(_marketingFee)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	**Type**	**Bases**		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	

	functionCallWithValue	Internal	✓	
	_functionCallWithValue	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	geUnlockTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-

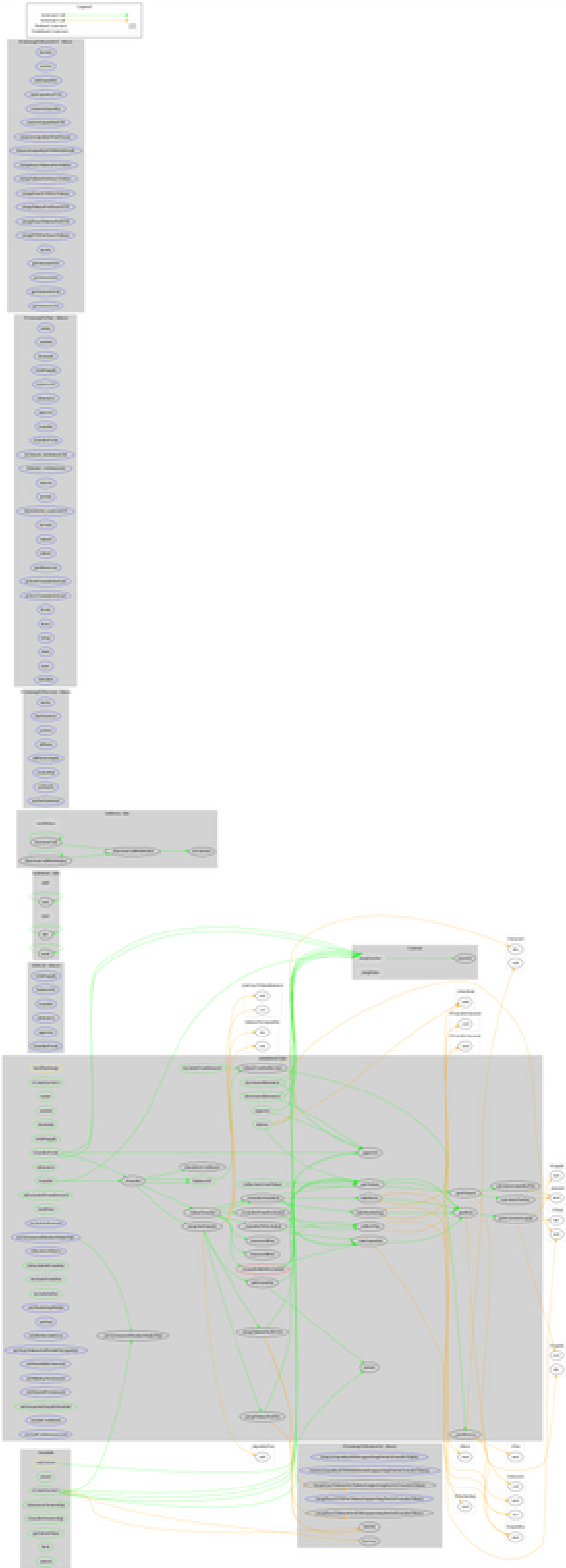
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
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	✓	-
DARKMATTER	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner

	includeInReward	External	✓	onlyOwner
	setAutomatedMarketMakerPair	External	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	_transferBothExcluded	Private	✓	
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateLiquidityFee	Private		
	calculateTaxFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForBUSD	Private	✓	
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	takeMarketing	Private	✓	
	takeBurn	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setFees	External	✓	onlyOwner
	antiBotlistAddress	External	✓	onlyOwner

	setNumTokensSellToAddToLiquidity	External	✓	onlyOwner
	setMaxWalletAmount	External	✓	onlyOwner
	setMinBuyTxAmount	External	✓	onlyOwner
	setMaxSellTxAmount	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	enableCooldown	External	✓	onlyOwner
	checkForCoolDown	Public	✓	-
	setSellCooldownperiod	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	spaceheroes.world
Registry Domain ID	5690d2c4e5014d1691191f9064f62931-DONUTS
Creation Date	2021-07-01T20:31:52Z
Updated Date	2021-07-06T20:31:57Z
Registry Expiry Date	2022-07-01T20:31:52Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created in 6 days 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 stopping transactions, manipulating fees and blacklisting addresses. 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.

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