Audit Report

MetaMew

February 2022

Type	BEP20
Network	BSC
Address	0x4AB62aF01f49B91C2Cc82C16DAF0DACd84ed52e1
Audited by	© coinscope

Table of Contents

Contract Review

Contract Name	MetaMew
Compiler Version	v0.8.11+commit.d7f03943
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x4AB62aF01f49B91C2Cc82C16DAF0DACd84ed52e1
Symbol	MMEW
Decimals	9
Total Supply	1,000,000,000,000,000
Source	contract.sol
Domain	metamew.net

Audit Updates

Initial Audit	12th February 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#L962,1012

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 `_sellTaxFee` to a high value.

```
if (takeFee && to == _pancakeswapV2LiquidityPair) {
    // We will assume that the normal sell tax rate will apply
    uint256 fee = _sellTaxFee;
```

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 _maxTxAmount to zero.

```
if(from != owner() && to != owner()) {
    require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
}
```

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.

Read more about the stop sales recommendation in the limit fees manipulation section.

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#L722,726

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 `setSellTaxFeePerecent` function with a high percentage value.

```
function setSellTaxFeePerecent(uint256 taxFee) external onlyOwner() {
    _sellTaxFee = taxFee;
}
```

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#L1149

Description

The contract owner has the authority to transfer funds to the team wallet. These funds have been swiped from the swap & liquify feature. From the accumulated funds:

- 1/7 is going to the auto generated liquidity pool.
- 4/7 is going to the marketing wallet.
- The rest is accumulated in the contract.
- The buyback fee is 1% of the accumulated amount.
- The contract owner has the ability to withdraw the remaining amount by calling the withdrawBNB function.

```
function withdrawBNB(uint256 amount) public onlyOwner() {
    if(amount == 0) payable(owner()).transfer(address(this).balance);
    else payable(owner()).transfer(amount);
}
```

```
function swapAndLiquify(uint256 tokenAmount) private lockSwapping {
    // Split the contract balance into the swap portion and the liquidity
portion
    uint256 eigth      = tokenAmount.div(8);     // 1/8 of the tokens, used for
liquidity
    uint256 swapAmount = tokenAmount.sub(eigth); // 7/8 of the tokens, used to
swap for BNB

    // Capture the contract's current BNB balance so that we know exactly the
amount of BNB that the swap creates.
    // This way the liquidity event will not include any BNB that has been
collected by other means.
```

```
    uint256 initialBalance = address(this).balance;

    // Swap 7/8ths of MMEW tokens for BNB
    swapTokensForBNB(swapAmount);

    // How much BNB did we just receive
    uint256 receivedBNB = address(this).balance.sub(initialBalance);

    // A seventh of the received BNB will be paired with the eigth of tokens
left behind
    uint256 liquidityBNB = receivedBNB.div(7);

    // Add liquidity via the PancakeSwap V2 Router
    addLiquidity(eigth, liquidityBNB);

    // We now have 6/7ths left of BNB (converted from MMEW)
    // We want to send 4/7ths to the marketing wallet and keep 2/7ths within the
contract for buyback
    uint256 marketingBNB = receivedBNB.div(7).mul(4);

    // Send the remaining BNB to the marketing wallet
    transferBNBToAddress(_marketingAddress, marketingBNB);

    emit SwapAndLiquify(swapAmount, liquidityBNB, eigth);
}
```

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
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L07	Missing Events Arithmetic
	L15	Local Scope Variable Shadowing
	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L244,252,590,598,613,793 and 1 more

Description

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

```
reflect
reflectionFromToken
setRouterAddress
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L501,480,478,479,515

Description

Constant state variables should be declared constant to save gas.

```
_tTotal
_symbol
_name
...
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L297,298,315,336,483,484 and 22 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.

```
_lastRoll
_previousWinTime
_previousWonAmount
...
```

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#L186,123,127,135,143,171 and 8 more

Description

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

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

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L722,726,730,734,738,750 and 7 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.

```
_lottoMinimumSpend = minimumSpend
_lottoThreshold = threshold
_lottoChance = chance
...
```

Recommendation

Emit an event for critical parameter changes.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L653,678

Description

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

```
owner
```

Recommendation

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

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1149

Description

Performing divisions before multiplications may cause lose of prediction.

```
marketingBNB = receivedBNB.div(7).mul(4)
```

Recommendation

The multiplications should be prior to the divisions.

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		
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	✓	

	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
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		-

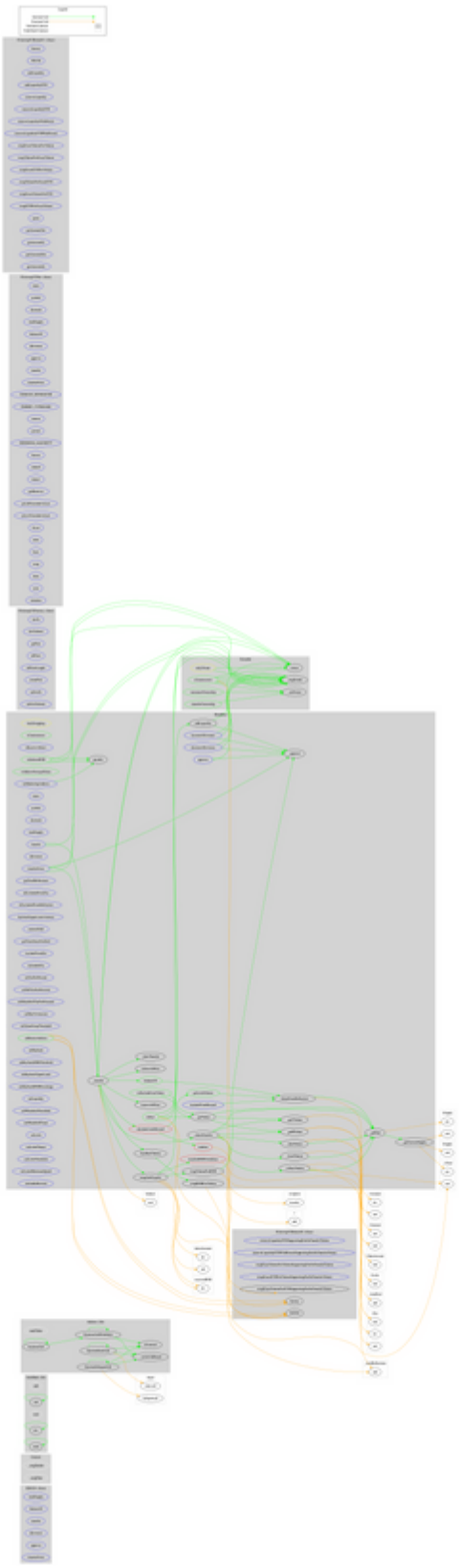
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	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	✓	-
MetaMew	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	<Receive Ether>	External	Payable	-
	withdrawBNB	Public	✓	onlyOwner
	withdrawForeignToken	Public	✓	onlyOwner
	transferBNBToAddress	Private	✓	
	setRouterAddress	Public	✓	onlyOwner
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	Public		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	_approve	Private	✓	

	getTotalReflections	External		-
	isExcludedFromFee	External		-
	isExcludedFromReflection	External		-
	getLottoTokens	Public		-
	buybackUpperLimitAmount	External		-
	amountSold	External		-
	getTimeSinceFirstSell	External		-
	excludeFromFee	External	✓	onlyOwner
	includeInFee	External	✓	onlyOwner
	setTaxFeePercent	External	✓	onlyOwner
	setSellTaxFeePerecent	External	✓	onlyOwner
	setWhaleSellTaxFeePerecent	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	setTokenSwapThreshold	External	✓	onlyOwner
	setMarketingAddress	External	✓	onlyOwner
	setBuyback	External	✓	onlyOwner
	setBuybackBNBThreshold	External	✓	onlyOwner
	setBuybackUpperLimit	External	✓	onlyOwner
	setBuybackBNBPercentage	External	✓	onlyOwner
	setLiquidity	External	✓	onlyOwner
	setWhaleSellThreshold	External	✓	onlyOwner
	setWhaleSellTimer	External	✓	onlyOwner
	setLotto	External	✓	onlyOwner
	setLottoChance	External	✓	onlyOwner
	setLottoThreshold	External	✓	onlyOwner
	setLottoMinimumSpend	External	✓	onlyOwner
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	random	Private	✓	
	removeAllFees	Private	✓	
	restoreAllFees	Private	✓	
	calculateLottoReward	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		

	_getRate	Private		
	_getCurrentSupply	Private		
	excludeFromReward	External	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	_lottoTransfer	Private	✓	
	_transfer	Private	✓	
	_tokenTransfer	Private	✓	
	_burnTokens	Private	✓	
	_reflectTokens	Private	✓	
	_takeTokens	Private	✓	
	buyBackTokens	Private	✓	lockSwapping
	swapAndLiquify	Private	✓	lockSwapping
	swapTokensForBNB	Private	✓	
	swapBNBForTokens	Private	✓	
	addLiquidity	Private	✓	
	reflect	Public	✓	-

Contract Flow



Domain Info

Domain Name	metamew.net
Registry Domain ID	2663454029_DOMAIN_NET-VRSN
Creation Date	2021-12-23T10:36:27.000Z
Updated Date	2022-01-01T14:59:47.000Z
Registry Expiry Date	
Registrar WHOIS Server	whois.ionos.com
Registrar URL	http://ionos.com
Registrar	IONOS SE
Registrar IANA ID	83

The domain has been created about 2 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 manipulating fees, stopping transactions and transferring funds to the team's wallet. If the fees are abused by the contract owner, the contract could operate as a honeypot. 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 also contains a lottery logic. Every buy transaction has the chance to win the awarded amount. The awarded amount is accumulated from a dedicated fee. The winning chance and the awarded amount can be configured by the contract owner.

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