Audit Report

Octaverse Games

December 2021

Type	BEP20
Address	0xd08bBD57BEee1f4eFE1A9306495BFcfc01dAf8f4
Audited by	© coinscope

Table of Contents

Table of Contents 1
Contract Review 3
Audit Updates 3
Contract Analysis 4
OTUT - Owner Transfer User's Tokens 5
Description 5
Recommendation 5
ELFM - Exceed Limit Fees Manipulation 6
Description 6
Recommendation 6
Contract Diagnostics 7
L08 - Tautology or Contradiction 8
Description 8
Recommendation 8
L07 - Missing Events Arithmetic 9
Description 9
Recommendation 9
L09 - Dead Code Elimination 10
Description 10
Recommendation 10
L04 - Conformance to Solidity Naming Conventions 11
Description 11
Recommendation 11
L03 - Redundant Statements 12
Description 12
Recommendation 12

L02 - State Variables could be Declared Constant 13
Description 13
Recommendation 13
L01 - Public Function could be Declared External 14
Description 14
Recommendation 14
Contract Functions 15
Contract Flow 20
Domain Info 21
Summary 22
Disclaimer 23
About Coinscope 24

Contract Review

Contract Name	Token
Compiler Version	v0.8.6+commit.11564f7e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xd08bBD57BEee1f4eFE1A9306495BFcfc01dAf8f4
Symbol	OVG
Decimals	18
Total Supply	1,000,000,000,000
Source	contract.sol
Domain	octaversegames.com

Audit Updates

Initial Audit	31st December 2021
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	Contract Owner is not able to 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

OTUT - Owner Transfer User's Tokens

Criticality	critical
Location	contract.sol#L1328

Description

The contract owner has the authority to transfer the balance of a user's contract to the owner's contract. The owner may take advantage of it by calling the `recoverBEP20` function.

```
function recoverBEP20(address tokenAddress, uint256 tokenAmount) public
onlyOwner {
    // do not allow recovering self token
    require(tokenAddress != address(this), "Self withdraw");
    IERC20(tokenAddress).transfer(owner(), tokenAmount);
}
```

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.

ELFM - Exceed Limit Fees Manipulation

Criticality	medium
Location	contract.sol#L935

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 `setAllFeePercent` function with a high percentage value. There is an upper limit on the total fee since every fee has 10% maximum value. Hence, the maximum total fee can be 50%.

```
    function setAllFeePercent(uint8 taxFee, uint8 liquidityFee, uint8 burnFee,
uint8 walletFee, uint8 buybackFee) external onlyOwner() {
        require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
        require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
        require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
        require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
        require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
        _taxFee = taxFee;
        _liquidityFee = liquidityFee;
        _burnFee = burnFee;
        _buybackFee = buybackFee;
        _walletFee = walletFee;
    }
```

Recommendation

The contract could embody a check for the maximum acceptable value. For instance, it could check if the sum of the fees is less than 25%.

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
	L08	Tautology or Contradiction
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L04	Conformance to Solidity Naming Conventions
	L03	Redundant Statements
	L02	State Variables could be Declared Constant
	L01	Public Function could be Declared External

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L932,L933,L934

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(burnFee >= 0 && burnFee <= maxBurnFee,BF err)
require(bool,string)(walletFee >= 0 && walletFee <= maxWalletFee,WF err)
require(bool,string)(taxFee >= 0 && taxFee <= maxTaxFee,TF err)
...
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L932,L949,L953 and 1 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.

```
_taxFee = taxFee
buyBackUpperLimit = buyBackLimit * 10 ** 18
_maxTxAmount = _tTotal.mul(maxTxPercent).div(10 ** 2)
...
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L359,L319,L329 and 13 more

Description

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

```
_functionCallWithValue
functionCall
functionCall
...
```

Recommendation

Remove unused functions.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L560,L967,L1032 and 11 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.

```
WETH
_enabled
_amount
...
```

Recommendation

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

L03 - Redundant Statements

Criticality	minor
Location	contract.sol#L239

Description

Detect the usage of redundant statements that have no effect.

Context

Recommendation

Remove redundant statements if they congest code but offer no value.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L703,L707,L709 and 7 more

Description

Constant state variables should be declared constant to save gas.

```
dead
maxBurnFee
maxBuybackFee
...
```

Recommendation

Add the constant attribute to state variables that never change.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L500,L509,L515 and 23 more

Description

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

```
renounceOwnership
transferOwnership
geUnlockTime
...
```

Recommendation

Use the external attribute for functions never called from the contract

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
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		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

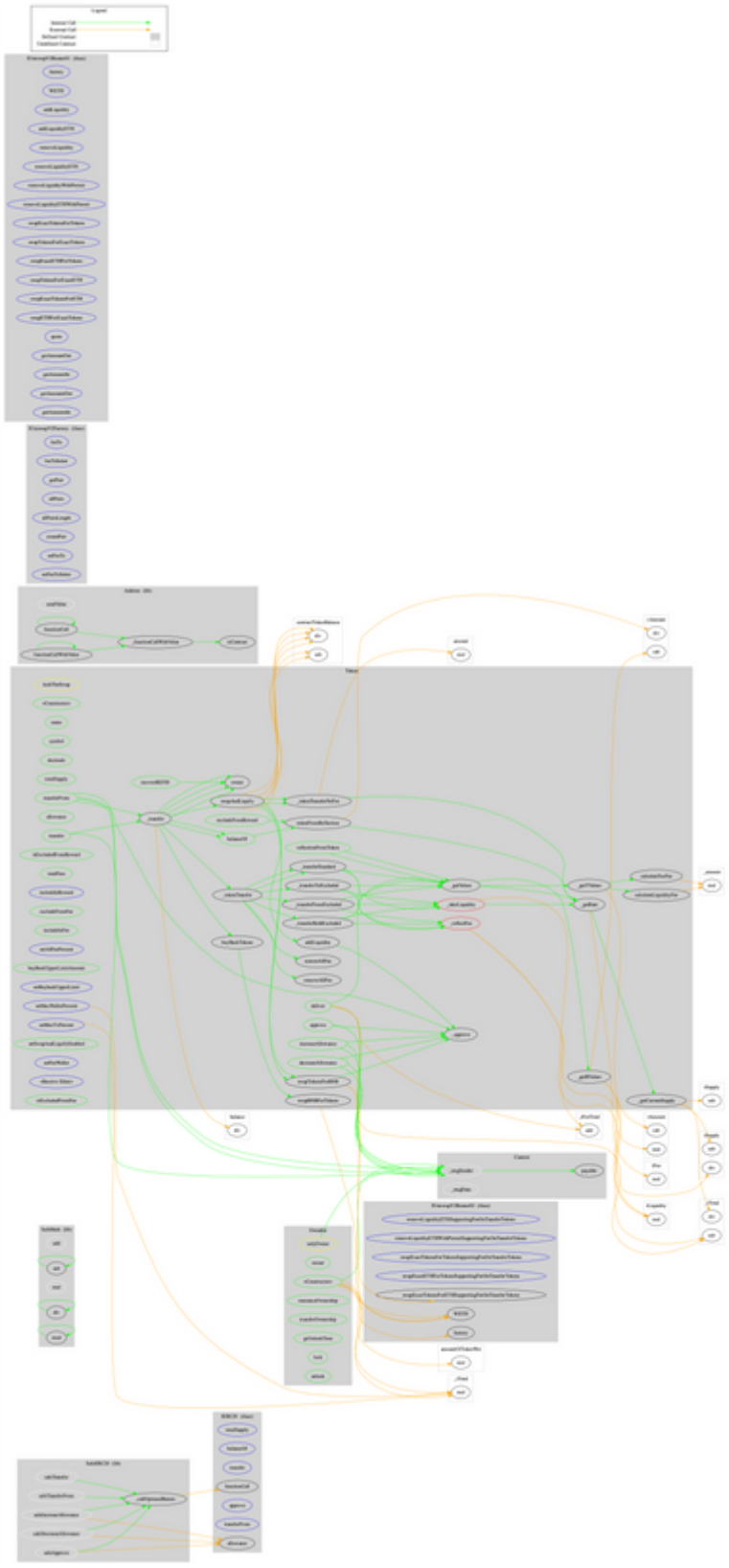
	_functionCallWithValue	Private	✓	
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	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	✓	-
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	✓	-
Token	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
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setAllFeePercent	External	✓	onlyOwner
	buyBackUpperLimitAmount	Public		-
	setBuybackUpperLimit	External	✓	onlyOwner
	setMaxTxPercent	External	✓	onlyOwner
	setMaxWalletPercent	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	setFeeWallet	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-

	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	buyBackTokens	Private	✓	lockTheSwap
	swapTokensForBNB	Private	✓	
	swapBNBForTokens	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_tokenTransferNoFee	Private	✓	
	recoverBEP20	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	octaversegames.com
Registry Domain ID	2658998645_DOMAIN_COM-VRSN
Creation Date	2021-12-02T14:14:42Z
Updated Date	2021-12-02T14:14:42Z
Registry Expiry Date	2022-12-02T14:14:42Z
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	http://www.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 29 days 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

Octaverse Games is aiming to create a decentralized Gaming developer on the binance Smart Chain, building games that support NFT trading and Play-To-Earn. The token has a friendly and growing community. The contract analysis reported one major and one medium risk issue. The contract owner is able to transfer tokens from any address to the owner's wallet. The contract owner can potentially set the total fees to 50%. It could have a check for a more reasonable maximum value like 25%. 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