



Audit Report

LudoP2E

August 2022

Type	BEP20
Network	BSC
Address	0x1b1de097d1D241ad72D80467259492D60C359979
Audited by	© coinscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
Contract Diagnostics	5
ZD - Zero Division	6
Description	6
Recommendation	6
L01 - Public Function could be Declared External	7
Description	7
Recommendation	7
L04 - Conformance to Solidity Naming Conventions	8
Description	8
Recommendation	8
L07 - Missing Events Arithmetic	9
Description	9
Recommendation	9
L09 - Dead Code Elimination	10
Description	10
Recommendation	10
Contract Functions	11
Contract Flow	15
Summary	16
Disclaimer	17
About Coinscope	18

Contract Review

Contract Name	PlayLudo
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x1b1de097d1D241ad72D80467259492D60C359979
Symbol	PlayLudo
Decimals	18
Total Supply	1,000,000,000
Domain	http://www.playludogame.com

Source Files

Filename	SHA256
contract.sol	97ae32493ec9555c5f3754ae4cb10a31aa51fc755678ae9e500e8d44c2fd1710

Audit Updates

Initial Audit	1st August 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

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	ZD	Zero Division
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination

ZD - Zero Division

Criticality	critical
Location	contract.sol#L1111

Description

The contract is using variables that may be set to zero as denominators. As a result, the transactions will revert. The variable `totalFees` can be set to zero.

```
function swapAndLiquify(uint256 contractTokenBalance) private lockTheSwap {  
    uint256 tokensForLiquidity =  
contractTokenBalance.mul(liquidityFee).div(totalFees);  
    uint256 tokensForMarketing =  
contractTokenBalance.sub(tokensForLiquidity);
```

Recommendation

The contract should prevent those variables to be set to zero or should not allow to execute the corresponding statements.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L264,518,201,311,184,227,292,176,246

Description

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

```
approve  
name  
increaseAllowance  
transfer  
symbol  
decreaseAllowance  
decimals  
renounceOwnership  
transferFrom
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

contract.sol#L958,1027,1023,531,922

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.

```
marketingWallet  
WETH  
MarketingFee  
Account  
LiquidityFee
```

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#L958

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.

```
liquidityFee = LiquidityFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality

minor

Location

contract.sol#L389

Description

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

```
_burn
```

Recommendation

Remove unused functions.

Contract Functions

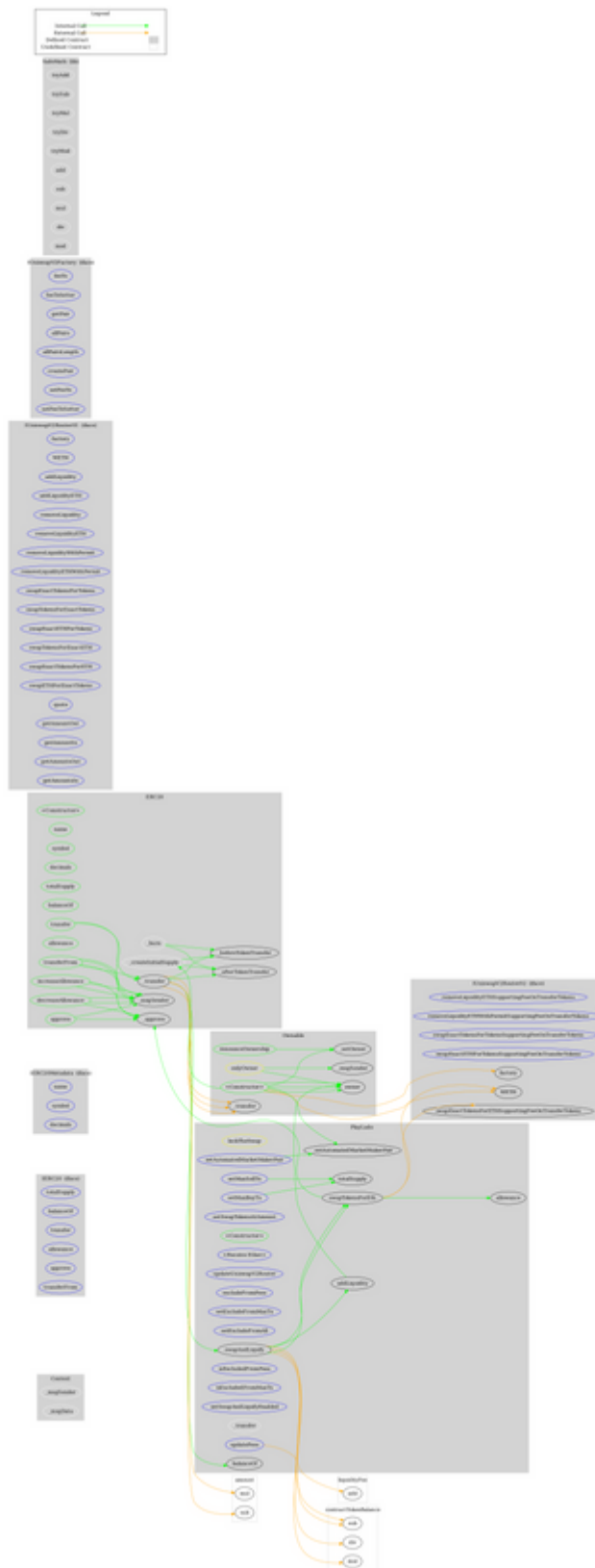
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
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	✓	
	_createInitialSupply	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
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	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
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		
PlayLudo	Implementation	ERC20, Ownable		
	updateFees	External	✓	onlyOwner
	setMaxSellTx	External	✓	onlyOwner
	setMaxBuyTx	External	✓	onlyOwner
	setSwapTokensAtAmount	External	✓	onlyOwner
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updateUniswapV2Router	External	✓	onlyOwner
	excludeFromFees	External	✓	onlyOwner
	setExcludeFromMaxTx	External	✓	onlyOwner
	setExcludeFromAll	External	✓	onlyOwner
	setAutomatedMarketMakerPair	External	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	isExcludedFromFees	External		-
	isExcludedFromMaxTx	External		-
	setSwapAndLiquifyEnabled	External	✓	onlyOwner
	_transfer	Internal	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	

Contract Flow



Summary

LudoP2E is aiming to build an online, multiplayer play-to-earn board game. The Smart Contract analysis reported one critical severity issue. 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>