



Cyberscope

Audit Report

Lava Token

April 2022

File	LavaToken.sol
Commit	605b9971b669eabf3e6727cb61d55f7cdd620e5a
Github	https://github.com/lavafinancial/LavaContracts
Audited by	© cyberscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Source Files	3
Contract Analysis	4
ELFM - Exceed Limit Fees Manipulation	5
Description	5
Recommendation	5
Update 13 April	6
BC - Blacklisted Contracts	7
Description	7
Recommendation	7
Update 13 April	7
Contract Diagnostics	8
Update 13 April	8
Unit Test	9
Contract Functions	10
Contract Flow	14
Summary	15
Update 13 April	15
Disclaimer	16
About Cyberscope	17

Contract Review

Github	LavaToken
commit	605b9971b669eabf3e6727cb61d55f7cdd620e5a
File	LavaToken.sol

Audit Updates

Initial Audit	9th April 2022
Corrected	13th April 2022

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	75e3c97011e75627ffb36f4a2799a4e887e1a3e27ed427490e82d7b6f51cc5c9
@openzeppelin/contracts/security/Pausable.sol	5b6abc290190f46b9941c674594eee083a3fe6b92d1828d0cfefacc94d1cac9a
@openzeppelin/contracts/token/ERC20/ERC20.sol	f7831910f2ed6d32acff6431e5998baf50e4a00121303b27e974aab0ec637d79
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	c2b06bb4572bb4f84bfc5477dadcd0fcc497cb66c3a1bd53480e68bedc2e154a6
@openzeppelin/contracts/token/ERC721/IERC721.sol	a88e8e63c7a737436f7ec62542620609ab07bb9a772e77146ed4dc98539e03d3
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5
contracts/interfaces/JoeRouter.sol	dd1c237f78c9eea8d6378ebdaa91c3002e85bb0a1b70517413e158c5f4bf9b48
contracts/LavaToken.sol	15a04df4e0541d7e30a87a55c1cbeb98197a6b073a9ee45cb1f444bce2174777

Contract Analysis

● Critical ● Medium ● Minor ● Pass

Severity	Code	Description	Resolved
●	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%)	Multisig Wallet
●	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	Multisig Wallet

ELFM - Exceed Limit Fees Manipulation

Criticality	minor
Location	contract.sol#L96
Status	Multisig Wallet

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 `setTaxes` function and setting the `_transferTax` to 5000. Additionally, there is an extra tax raised up to 50% in the sales if the issuer holds NFTs from the `lavaFinance`.

```
function setTaxes(uint _transferTax, uint _salesTax, uint _nodeSalesTax)
external onlyOwner {
    require(_salesTax <= 2500, "Cannot be more than 25%");
    require(_transferTax <= 5000, "Cannot be more than 50%");
    require(_nodeSalesTax <= 2000, "Cannot be more than 20%");
    transferTax = _transferTax;
    salesTax = _salesTax;
    nodeSalesTax = _nodeSalesTax;
}
```

```
uint totalSalesTax = salesTax;
if (nodeSalesTax > 0) {
    totalSalesTax += nodeSalesTax * IERC721(lavaFinance).balanceOf(from);
    if (totalSalesTax > 5000) {
        totalSalesTax = 5000;
    }
}
```

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.

Update 13 April

The team has noted that all administrative functionality is secured by a 3/5 multisig wallet with doxxed and KYC'd members.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L132
Status	Multisig Wallet

Description

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

```
require(!isBlacklisted(from), "Not allowed");
```

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.

Update 13 April

The team has noted that all administrative functionality is secured by a 3/5 multisig wallet with doxxed and KYC'd members.

Contract Diagnostics

Update 13 April

The team has resolved all the issues that were mentioned in the `d59617e3ac107eea6d7601aac6e73e7f45ee00eb` commit.

Unit Test

- ✓ Test blacklist (75ms)
- ✓ Test pausable (62ms)
- ✓ Test max wallet (64ms)
- ✓ Test transfer tax (118ms)
- ✓ Test sales tax (618ms)
- ✓ Test lp buy (139ms)

Contract Functions

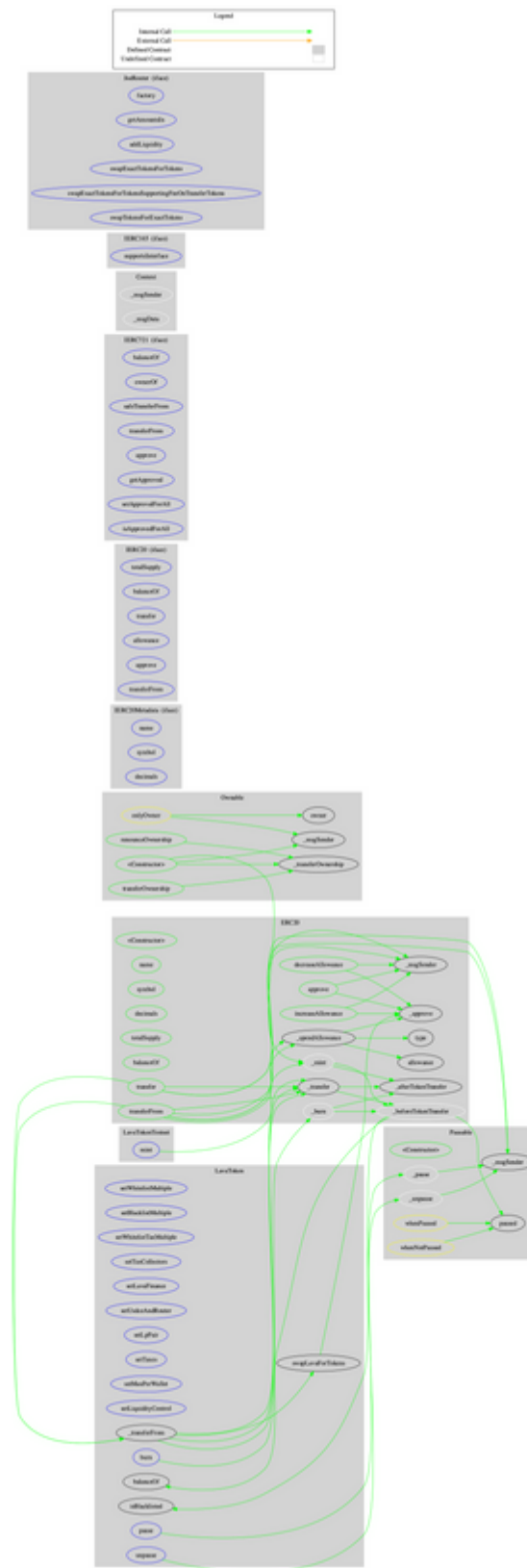
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
Pausable	Implementation	Context		
	<Constructor>	Public	✓	-
	paused	Public		-
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
ERC20	Implementation	Context, IERC20, IERC20Meta data		
	<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	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		

IERC165	Interface			
	supportsInterface	External		-
JoeRouter	Interface			
	factory	External		-
	getAmountsIn	External		-
	addLiquidity	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
LavaToken	Implementation	ERC20, Ownable, Pausable		
	<Constructor>	Public	✓	ERC20
	setWhitelistMultiple	External	✓	onlyOwner
	setBlacklistMultiple	External	✓	onlyOwner
	setWhitelistTaxMultiple	External	✓	onlyOwner
	setTaxCollectors	External	✓	onlyOwner
	setLavaFinance	External	✓	onlyOwner
	setUsdcAndRouter	External	✓	onlyOwner
	setLpPair	External	✓	onlyOwner
	setTaxes	External	✓	onlyOwner
	setMaxPerWallet	External	✓	onlyOwner
	setLiquidityControl	External	✓	onlyOwner
	pause	External	✓	onlyOwner
	unpause	External	✓	onlyOwner
	burn	External	✓	-
	isBlacklisted	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	transferFrom	Public	✓	-
	transfer	Public	✓	-
	_transferFrom	Internal	✓	
	swapLavaForTokens	Private	✓	

LavaTokenTest net	Implementation	LavaToken		
	<Constructor>	Public	✓	LavaToken
	mint	External	✓	-

Contract Flow



Summary

Lava Token is an interesting project that has a friendly and growing community. The Smart Contract analysis reported one issue. The contract owner can blacklist contracts. Other than that, the contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 25% fees.

Update 13 April

The team has noted that all administrative functionality is secured by a 3/5 multisig wallet with doxxed and KYC'd members.

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 provides all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>