Cyberscope

Audit Report

Lava G Token

April 2022

File	gLava.sol
Commit	d59617e3ac107eea6d7601aac6e73e7f45ee00eb
Github	https://github.com/lavafinancial/LavaContracts
Audited by	© cyberscope

Table of Contents

Contract Review

Github	LavaFinance
commit	d59617e3ac107eea6d7601aac6e73e7f45ee00eb
File	gLava.sol

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	75e3c97011e75627ffb36f4a2799a4e887e1a3e27ed427490e82d7b6f51cc5c9
@openzeppelin/contracts/token/ERC20/ERC20.sol	f7831910f2ed6d32acff6431e5998baf50e4a00121303b27e974aab0ec637d79
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	c2b06bb4572bb4f84bfc5477dadc0fcc497cb66c3a1bd53480e68bedc2e154a6
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
contracts/gLava.sol	dc843376c768ab46e1e7e335c91e02a2d5e07e1a6620bfcd0e04af7271402d00

Audit Updates

Initial Audit	10th April 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

MT - Mint Tokens

Criticality	critical
Location	contract.sol#L28

Description

The lavaFinance address as a “minter” role has the authority to mint tokens. The “minter” role may take advantage of it by calling the `mint` function. As a result the contract tokens will be highly inflated. The contract owner can set the minter role.

```
function mint(address recipient, uint amount) external onlyMinters virtual {
    _mint(recipient, amount);
}
```

Recommendation

The owner should carefully manage the credentials of the owner’s account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L46

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(!blacklisted[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.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L06	Missing Events Access Control

L01 - Public Function could be Declared External

Criticality	minor
Location	contracts/gLava.sol#L38

Description

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

```
votingPower
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contracts/gLava.sol#L8,23

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.

```
_lavaFinance
gLava
```

Recommendation

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

L06 - Missing Events Access Control

Criticality	minor
Location	contracts/gLava.sol#L23

Description

Detected missing events for critical access control parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
lavaFinance = _lavaFinance
```

Recommendation

Emit an event for critical parameter changes.

Unit Test

✓ Test minting (228ms)

✓ Test blacklist transfer (105ms)

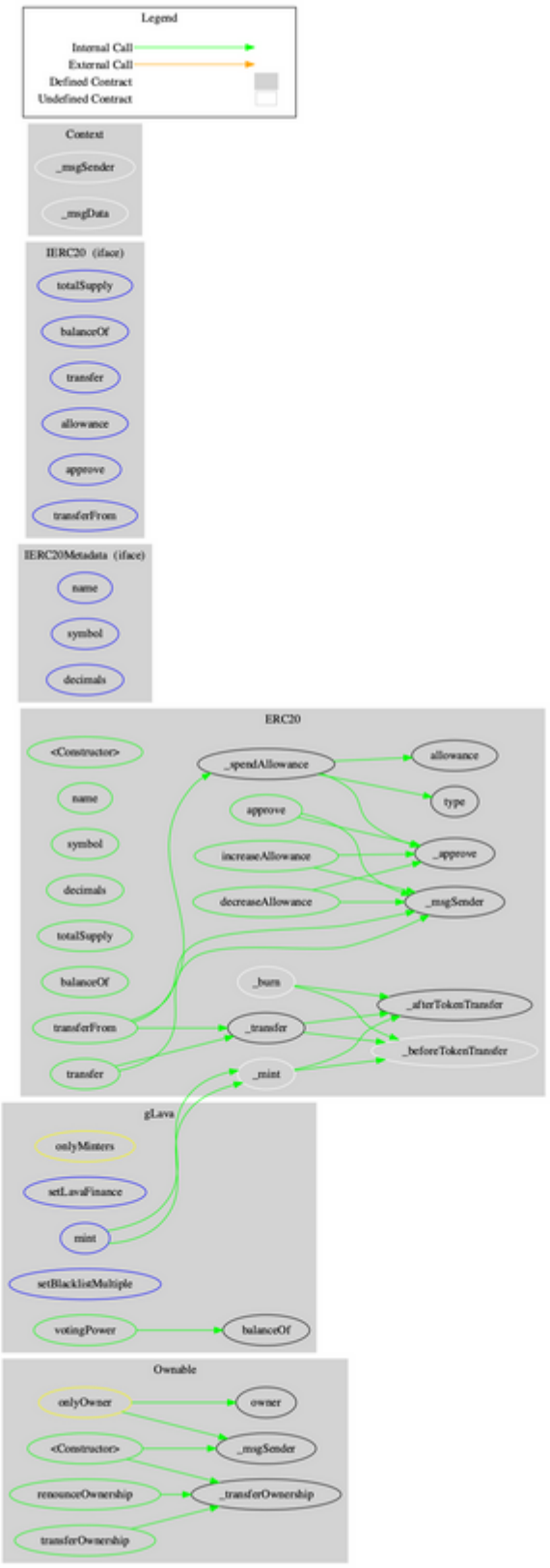
✓ Test blacklist voting power (113ms)

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	✓	
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	✓	
	_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	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
gLava	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	setLavaFinance	External	✓	onlyOwner
	mint	External	✓	onlyMinters
	setBlacklistMultiple	External	✓	onlyOwner
	votingPower	Public		-
	_beforeTokenTransfer	Internal	✓	

Contract Flow



Summary

There are some functions that can be abused by the owner, like minting tokens and blacklisting addresses. We state that the owner privileges are necessary and required for proper protocol operations of the Lava Finance ecosystem. Thus, we emphasise the contract owner to be extra careful with the credentials. 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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io