Cyberscope

Audit Report

Stelia

June 2022

Type	BEP20
Network	BSC
Address	0x0019450b0fb021ad2e9f7928101b171272cd537c
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Stelia
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x0019450b0fb021ad2e9f7928101b171272cd537c
Symbol	STELIA
Decimals	18
Total Supply	60,000,000
Domain	stelia.finance

Source Files

Filename	**SHA256**
contract.sol	78ac950621338cfa1a5c14c96781d61991e7049aa0c4c73f172865022a0794eb

Audit Updates

Initial Audit	2nd June 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	critical
Location	contract.sol#L810, 851, 694

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by enabling/disabling the isAntibotModeEnabled boolean.

```
if (!isAntibotModeEnabled) return;
if (from == owner() || from == airdropContract) return;
require(antibotModeWhitelist[from] && antibotModeWhitelist[to], "Address not
in antibot mode whitelist");
```

The contract owner can also convert the contract into a honeypot and prevent users from selling by increasing the selling taxes.

```
}else if(isSell){
_taxFee = _sellTaxFee;
_advestisementFee = _sellAdvestisementFee;
}
```

```
uint256 tFee = calculateTaxFee(tAmount);
uint256 tAdvertisement = calculateAdvestisementFee(tAmount);
uint256 tBurn = calculateBurnFee(tAmount);

uint256 tTransferAmount = tAmount.sub(tFee).sub(tAdvertisement).sub(tBurn);
```

Recommendation

The contract could embody a check for not allowing setting the total fees more than a reasonable amount.

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#L677

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 setAdvestisementFeePercent function with a high percentage value.

```
function setAdvestisementFeePercent(uint256 buyAdvestisementFee, uint256
sellAdvestisementFee) external onlyOwner() {
_sellAdvestisementFee = sellAdvestisementFee;
_buyAdvestisementFee = buyAdvestisementFee;
}
```

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
	BLC	Business Logic Concern
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination

BLC - Business Logic Concern

Criticality	minor
Location	contract.sol#L804

Description

The business logic seems peculiar. The implementation may not follow the expected behavior. In both iterations the owner is able to enable the antibotModeWhitelist addresses.

```
function setAntibotModeWhitelist(address[] memory toAddAddesses, address[]
memory toRemoveAddesses) public onlyOwner {
        for (uint256 i = 0; i < toAddAddesses.length; i++)
antibotModeWhitelist[toAddAddesses[i]] = true;
        for (uint256 i = 0; i < toRemoveAddesses.length; i++)
antibotModeWhitelist[toRemoveAddesses[i]] = true;
    }
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L429,438,444,449,457,547,551,555,559,563,567,572,577,581,586,592,597,602,606,610,661,665,669,780,792,796,800,804

Description

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

```
setAntibotModeWhitelist
setAirdropContract
turnOnAntibotMode
turnOffAntibotMode
isExcludedFromFee
includeInFee
manageAmmPairs
excludeFromFee
reflectionFromToken
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L502,500,523,498,505,499,494

Description

Constant state variables should be declared constant to save gas.

```
_tTotal
_symbol
_sellTaxFee
_name
_maxTxAmount
_decimals
_buyTaxFee
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L475,745,751,757,800,502,503,505,506,518,523

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.

```
_maxTxAmount
_advestisementFee
_sellAdvestisementFee
_sellTaxFee
_buyAdvestisementFee
_buyTaxFee
_airdropContract
_amount
WETH
...
```

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#L673,677

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.

```
_sellAdvestisementFee = sellAdvestisementFee
_burnFee = fee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L355,315,325,340,350,262,289

Description

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

```
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
```

Recommendation

Remove unused functions.

Contract Functions

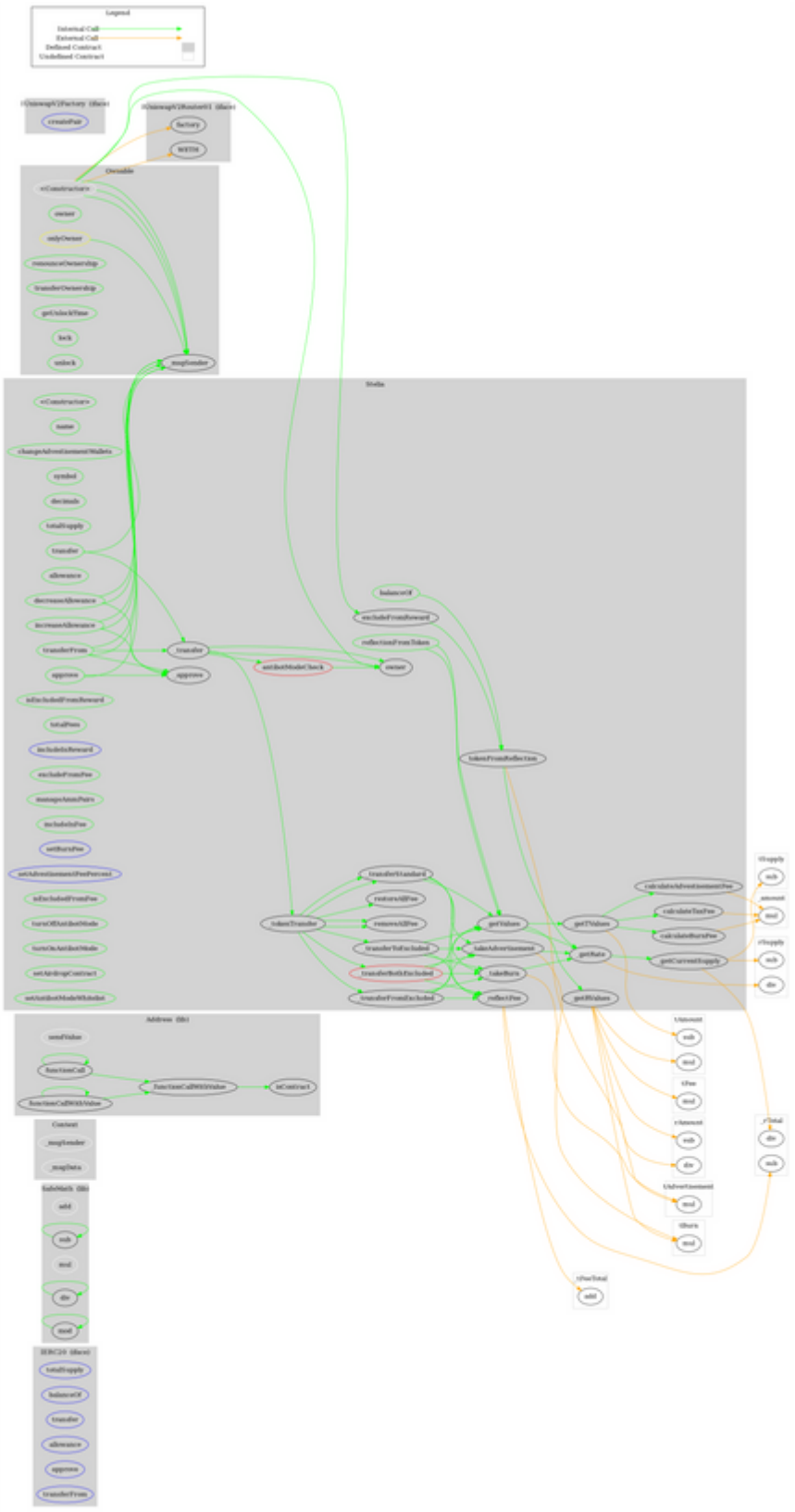
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	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	✓	
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	geUnlockTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
Stelia	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	changeAdvestisementWallets	Public	✓	onlyOwner
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-

	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	_transferBothExcluded	Private	✓	
	excludeFromFee	Public	✓	onlyOwner
	manageAmmPairs	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setBurnFee	External	✓	onlyOwner
	setAdvestisementFeePercent	External	✓	onlyOwner
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeAdvertisement	Private	✓	
	_takeBurn	Private	✓	
	calculateTaxFee	Private		
	calculateAdvestisementFee	Private		
	calculateBurnFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	turnOffAntibotMode	Public	✓	onlyOwner
	turnOnAntibotMode	Public	✓	onlyOwner
	setAirdropContract	Public	✓	onlyOwner
	setAntibotModeWhitelist	Public	✓	onlyOwner
	antibotModeCheck	Private		
	_transfer	Private	✓	
	_tokenTransfer	Private	✓	

	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	

Contract Flow



Domain Info

Domain Name	stelia.finance
Registry Domain ID	bc99d451d24549fd8dd512298313cdef-DONUTS
Creation Date	2022-04-08T04:45:54Z
Updated Date	2022-05-27T23:20:19Z
Registry Expiry Date	2023-04-08T04:45:54Z
Registrar WHOIS Server	whois.porkbun.com
Registrar URL	http://porkbun.com
Registrar	Porkbun LLC
Registrar IANA ID	1861

The domain has been created about 2 months before the creation of the audit. It will expire in 10 months.

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 stopping transactions and manipulating fees. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. 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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io