Cyberscope

Audit Report

Bull financial

March 2022

Type	BEP20
Network	BSC
Sha256	5351271b39612ef3f6f70e5ee1cf0b8004cefb849288f12952cac4ce6349405e
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BullToken
Symbol	BULL
Decimals	18
Total Supply	1,000,000
Domain	bullfinancial.org

Source Files

Filename	**SHA256**
contract.sol	5351271b39612ef3f6f70e5ee1cf0b8004cefb849288f12952cac4ce6349405e

Audit Updates

Initial Audit	26th March 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#L1359

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `mint` function. As a result the contract tokens will be highly inflated.

```
    function mint(address to, uint256 amount) public onlyOwner {
        _mint(to, amount);
    }
```

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.

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L1370

Description

The contract owner has the authority to stop selling transactions for all users excluding the owner. The owner may take advantage of it by setting the `taxFee` to a high value percent. This will effectively convert the contract into a honeypot.

```
if(to == pair && pair != address(0) && BullManager != owner) {
          uint256 taxFee = _amount * tax / 100;
          _transfer(owner, OPERATING_COST, taxFee);
          _amount -= taxFee;
      }
```

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	critical
Location	contract.sol#L1334

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 `setSellTaxPercent` function with a high percentage value.

```
    function setSellTaxPercent(uint _tax) external onlyOwner {
        tax = _tax;
    }
```

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
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L12	Using Variables before Declaration
	L15	Local Scope Variable Shadowing

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L479,487,694,702,726,733,1366,768,1378,813 and 8 more

Description

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

```
mint
unpause
pause
burnFrom
burn
nonces
permit
decreaseAllowance
increaseAllowance
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L315,316,317,319,320,321,434,1067,1022,1334 and 2 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.

```
BullManager
OPERATING_COST
_tax
_PERMIT_TYPEHASH
DOMAIN_SEPARATOR
_TYPE_HASH
_HASHED_VERSION
_HASHED_NAME
_CACHED_THIS
...
```

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#L1334

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.

```
tax = _tax
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L23,31,182,210,294,280,137,195,66,82 and 1 more

Description

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

```
toString
toHexString
tryRecover
toEthSignedMessageHash
recover
reset
decrement
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L142

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
r
```

Recommendation

The variables should be declared before any usage of them.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L1030

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
name
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

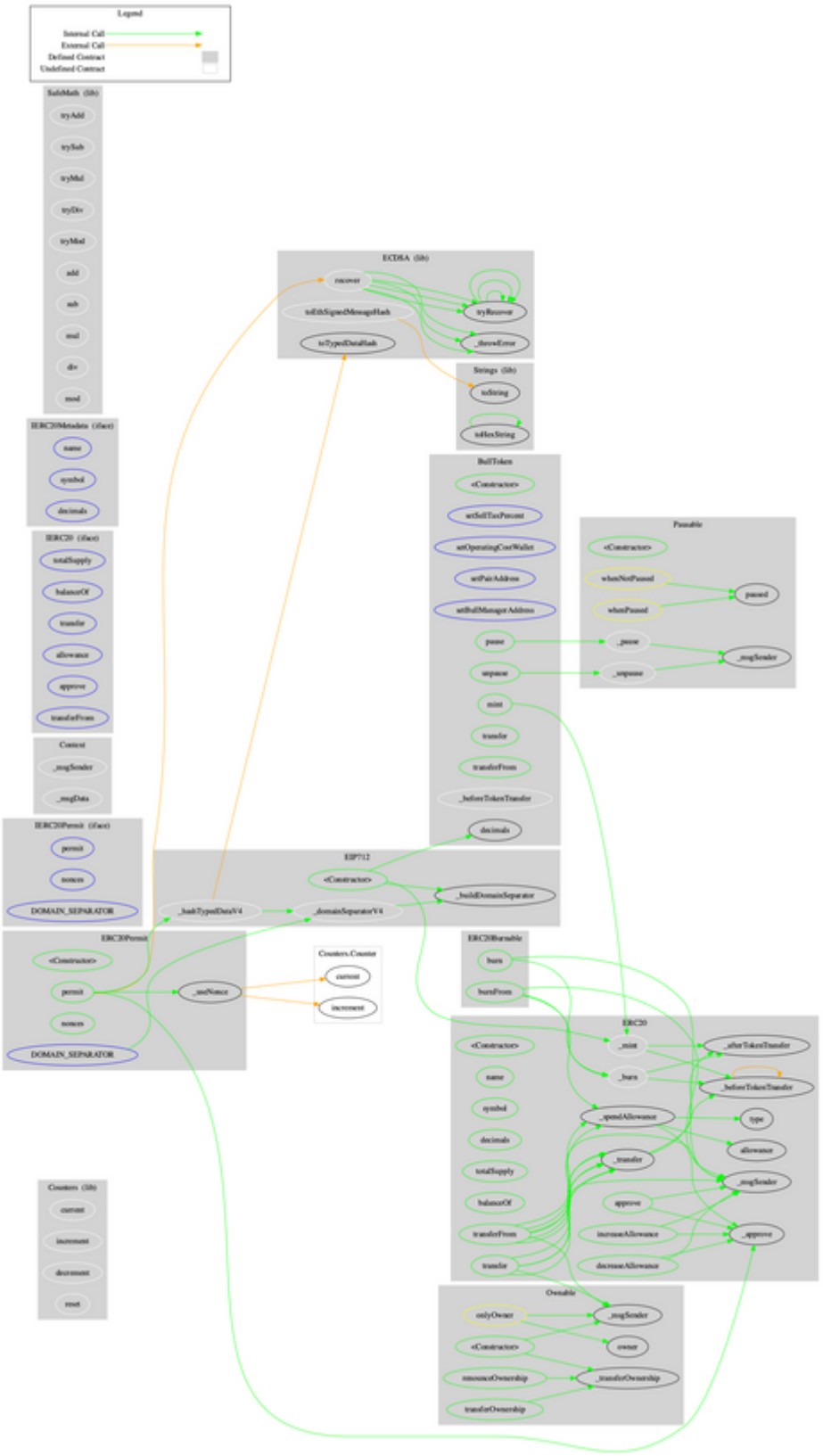
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Counters	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
	reset	Internal	✓	
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
ECDSA	Library			
	_throwError	Private		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	tryRecover	Internal		
	recover	Internal		
	toEthSignedMessageHash	Internal		
	toEthSignedMessageHash	Internal		
	toTypedDataHash	Internal		
EIP712	Implementation			
	<Constructor>	Public	✓	-
	_domainSeparatorV4	Internal		
	_buildDomainSeparator	Private		
	_hashTypedDataV4	Internal		

IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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
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	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
ERC20Permit	Implementation	ERC20, IERC20Permit, EIP712		
	<Constructor>	Public	✓	EIP712
	permit	Public	✓	-
	nonces	Public		-
	DOMAIN_SEPARATOR	External		-
	_useNonce	Internal	✓	
ERC20Burnable	Implementation	Context, ERC20		
	burn	Public	✓	-

	burnFrom	Public	✓	-
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		
BullToken	Implementation	ERC20, ERC20Burnable, Pausable, Ownable, ERC20Permit		
	<Constructor>	Public	✓	ERC20 ERC20Permit
	setSellTaxPercent	External	✓	onlyOwner
	setOperatingCostWallet	External	✓	onlyOwner
	setPairAddress	External	✓	onlyOwner
	setBullManagerAddress	External	✓	onlyOwner
	pause	Public	✓	onlyOwner
	unpause	Public	✓	onlyOwner
	mint	Public	✓	onlyOwner
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	_beforeTokenTransfer	Internal	✓	whenNotPaused

Contract Flow



Domain Info

Domain Name	bullfinancial.org
Registry Domain ID	D402200000018942331-LROR
Creation Date	2022-01-28T10:37:06Z
Updated Date	2022-01-28T10:37:07Z
Registry Expiry Date	2023-01-28T10:37:06Z
Registrar WHOIS Server	whois.networksolutions.com
Registrar URL	http://www.networksolutions.com
Registrar	Network Solutions, LLC
Registrar IANA ID	2

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

Bull financial is an interesting project that has a friendly and growing community. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. The owner can also mint tokens after initial deployment. 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