Cyberscope

Audit Report

WEEDPLANT

July 2022

Type	BEP20
Network	BSC
Address	0xdcb23832e9536d8eaaa97102cdc344c0ef24290b
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	CoinToken
Compiler Version	v0.4.24+commit.e67f0147
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0xDcb23832e9536d8EaAa97102cdc344c0ef24290B
Symbol	WPT
Decimals	18
Total Supply	95,000,000
Domain	https://weedplantnft.com

Source Files

Filename	**SHA256**
contract.sol	bb1bf689dfc02fe92fd154749b0eb05b9a6e816cb81579b190d9f8d2ca71d9d4

Audit Updates

Initial Audit	20th July 2022
Corrected	25th July 2022

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
Minor	ST	Contract Owner is not able to stop or pause transactions
Pass	OCTD	Contract Owner is not able to transfer tokens from specific address
Pass	OTUT	Owner Transfer User's Tokens
Critical	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
Pass	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
Critical	MT	Contract Owner is not able to mint new tokens
Pass	BT	Contract Owner is not able to burn tokens from specific wallet
Medium	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	minor
Location	contract.sol#L237,L241

Description

The contract owner has the authority to stop transactions for all users including the owner. The owner may take advantage of it by calling the `pause` function.

```
function transfer(address _to, uint256 _value) public whenNotPaused returns
(bool) {
    return super.transfer(_to, _value);
}

function transferFrom(address _from, address _to, uint256 _value) public
whenNotPaused returns (bool) {
    return super.transferFrom(_from, _to, _value);
}
```

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#L289

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 `updateFee` function with a high percentage value.

```
function updateFee(uint256 _txFee,uint256 _burnFee,address _FeeAddress)
onlyOwner public{
    txFee = _txFee;
    burnFee = _burnFee;
    FeeAddress = _FeeAddress;
}
```

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.

MT - Mint Tokens

Criticality	critical
Location	contract.sol#L304

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 account, uint256 amount) onlyOwner public {

        totalSupply = totalSupply.add(amount);
        balances[account] = balances[account].add(amount);
        emit Mint(address(0), account, amount);
        emit Transfer(address(0), account, 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	medium
Location	contract.sol#L224

Description

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

```
function blackListAddress(address listAddress,  bool isBlackListed) public
whenNotPaused onlyOwner  returns (bool success) {
       return super._blackList(listAddress, isBlackListed);
}
```

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
	STC	Succeeded Transfer Check
	CR	Code Repetition
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L11	Unnecessary Boolean equality

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L281

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
service.transfer(msg.value);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

CR - Code Repetition

Criticality	minor
Location	contract.sol#L132,L165

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

This code segment on function transfer is almost identical to the function transferFrom.

```
    require(tokenBlacklist[msg.sender] == false);
    require(_to != address(0));
    require(_value <= balances[msg.sender]);
    balances[msg.sender] = balances[msg.sender].sub(_value);
    uint256 tempValue = _value;
    if(txFee > 0 && msg.sender != FeeAddress){
        uint256 DenverDeflaionaryDecay = tempValue.div(uint256(100 / txFee));
        balances[FeeAddress] =
balances[FeeAddress].add(DenverDeflaionaryDecay);
        emit Transfer(msg.sender, FeeAddress, DenverDeflaionaryDecay);
        _value =  _value.sub(DenverDeflaionaryDecay);
    }
    if(burnFee > 0 && msg.sender != FeeAddress){
        uint256 Burnvalue = tempValue.div(uint256(100 / burnFee));
        totalSupply = totalSupply.sub(Burnvalue);
        emit Transfer(msg.sender, address(0), Burnvalue);
        _value =  _value.sub(Burnvalue);
    }
    // SafeMath.sub will throw if there is not enough balance.
balances[_to] = balances[_to].add(_value);
    emit Transfer(msg.sender, _to, _value);
    return true;
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L85,101,300,51,285,93,107,281,253

Description

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

```
blackListAddress
burn
allowance
unpause
updateFee
transferOwnership
mint
balanceOf
pause
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L237,281,285,196,189,128,161,241,249,201,207,245,233,118,157

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.

```
_owner
_value
_to
FeeAddress
_spender
_subtractedValue
_addedValue
_FeeAddress
_from
...
```

Recommendation

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

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L128,161

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
require(bool)(tokenBlacklist[msg.sender] == false)
```

Recommendation

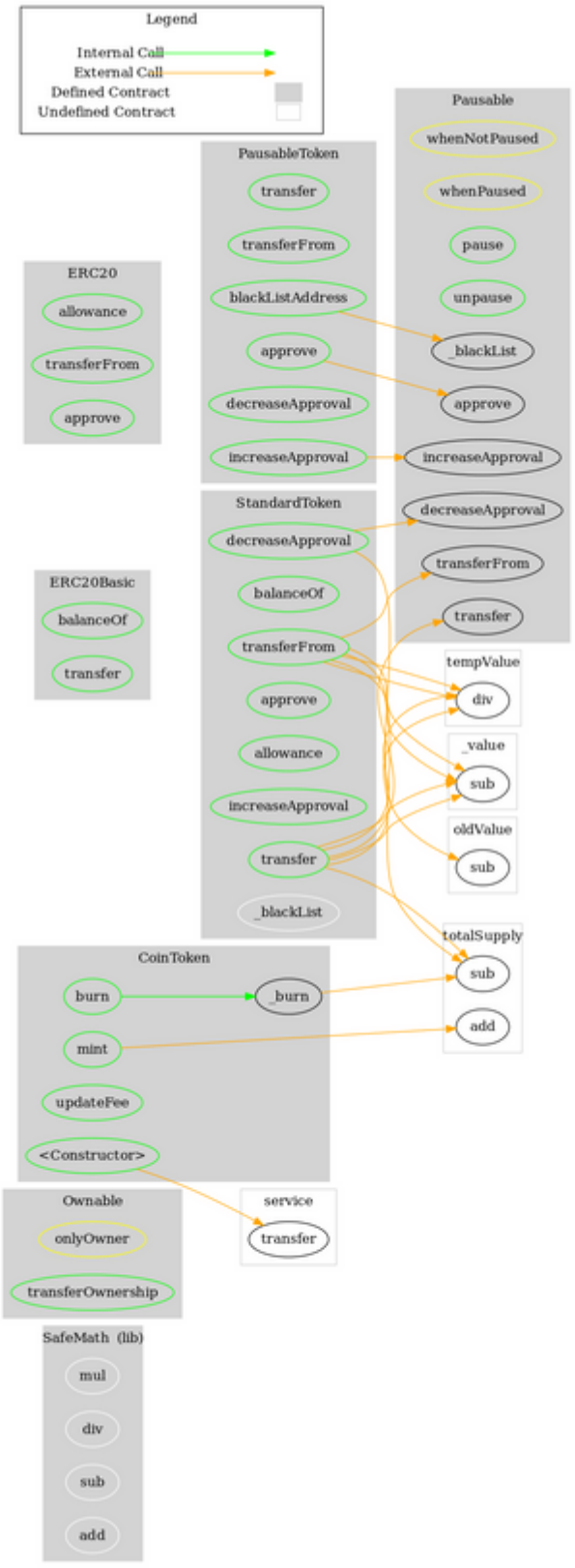
Remove the equality to the boolean constant.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
Ownable	Implementation			
	transferOwnership	Public	✓	onlyOwner
Pausable	Implementation	Ownable		
	pause	Public	✓	onlyOwner whenNotPaused
	unpause	Public	✓	onlyOwner whenPaused
ERC20Basic	Implementation			
	balanceOf	Public		-
	transfer	Public	✓	-
ERC20	Implementation	ERC20Basic		
	allowance	Public		-
	transferFrom	Public	✓	-
	approve	Public	✓	-
StandardToken	Implementation	ERC20		
	transfer	Public	✓	-
	balanceOf	Public		-

	transferFrom	Public	✓	-
	approve	Public	✓	-
	allowance	Public		-
	increaseApproval	Public	✓	-
	decreaseApproval	Public	✓	-
	_blackList	Internal	✓	
PausableToken	Implementation	StandardToken, Pausable		
	transfer	Public	✓	whenNotPaused
	transferFrom	Public	✓	whenNotPaused
	approve	Public	✓	whenNotPaused
	increaseApproval	Public	✓	whenNotPaused
	decreaseApproval	Public	✓	whenNotPaused
	blackListAddress	Public	✓	whenNotPaused onlyOwner
CoinToken	Implementation	PausableToken		
	<Constructor>	Public	Payable	-
	burn	Public	✓	-
	updateFee	Public	✓	onlyOwner
	_burn	Internal	✓	
	mint	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	weedplantnft.com
Registry Domain ID	2700223295_DOMAIN_COM-VRSN
Creation Date	2022-05-30T21:05:24Z
Updated Date	2022-05-30T21:05:25Z
Registry Expiry Date	2023-05-30T21:05:24Z
Registrar WHOIS Server	whois.publicdomainregistry.com
Registrar URL	www.publicdomainregistry.com
Registrar	PDR Ltd. d/b/a PublicDomainRegistry.com
Registrar IANA ID	303

The domain has been created in 10 months before the creation of the audit.

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, manipulating fees, minting tokens and blacklisting addresses. if the contract owner abuses the mint functionality, then the contract will be highly inflated. 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