Cyberscope

Audit Report

Peacock Mantis Shrimp

March 2022

Type	BEP20
Network	BSC
Address	0x173E1E0306b4cbF839cbA0596DC7EC7ff9A4beF7
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BurnableTaxHolderToken
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x173E1E0306b4cbF839cbA0596DC7EC7ff9A4beF7
Symbol	PMSP
Decimals	18
Total Supply	1,000,000,000,000,000
Source	contract.sol
Domain	mantisshrimptoken.com

Audit Updates

Initial Audit	11th 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

ELFM - Exceed Limit Fees Manipulation

Criticality	minor
Location	contract.sol#L706,713,721

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 changeReflectionFee, changeBurnFee and changeTaxFee function with the value 10.

```
function changeReflectionFee(uint256 newReflectionFee) public onlyOwner()
returns(bool) {
    require(newReflectionFee >= 0, "Reflection fee must be greater or equal to
zero");
    require(newReflectionFee <= 10, "Reflection fee must be lower or equal to
ten");
    _reflectionFee = newReflectionFee;
    return true;
}

function changeBurnFee(uint256 burnFee_) public onlyOwner() returns(bool) {
    require(burnFee_ >= 0, "Burn fee must be greater or equal to zero");
    require(burnFee_ <= 10, "Burn fee must be lower or equal to 10");
    _burnFee = burnFee_;
    return true;
}


function changeTaxFee(uint256 taxFee_) public onlyOwner() returns(bool) {
    require(taxFee_ >= 0, "Tax fee must be greater or equal to zero");
    require(taxFee_ <= 10, "Tax fee must be lower or equal to 10");
    _taxFee = taxFee_;
    return true;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value to be less than 25%.

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
	MTS	Manipulate Total Supply
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination
	L07	Missing Events Arithmetic
	L08	Tautology or Contradiction
	L13	Divide before Multiply Operation

MTS - Manipulate Total Supply

Criticality	minor
Location	contract.sol#L948

Description

The contract uses a fee called “burn”. This fee is removed from the transferred amount and the total supply. This change will have a direct impact on the token price and Market Cap.

```
function burnFeeTransfer(address sender, uint256 tAmount, uint256 currentRate)
private {
    uint256 tBurnFee = tAmount * _burnFee / 100;
    if(tBurnFee > 0){
        uint256 rBurnFee = tBurnFee * currentRate;
        _tTotal = _tTotal - tBurnFee;
        _rTotal = _rTotal - rBurnFee;
        emit Transfer(sender, address(0), tBurnFee);
    }
}
```

Recommendation

The contract owner should carefully manage the burn fee that adjust the circulating supply (increases or decreases), according to the token’s price fluctuations.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L173,181,252,260,648,652,303,322,340,368 and 20 more

Description

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

```
includeAccountinReward
excludeAccountFromReward
reflectionFromToken
reflect
changeTaxFee
changeBurnFee
changeReflectionFee
changeFeeAccount
includeInFee
...
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L226,230

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.

```
_totalSupply
_balances
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L465,442,411

Description

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

```
_transfer
_mint
_burn
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L706,713,721

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.

```
_taxFee = taxFee_
_burnFee = burnFee_
_reflectionFee = newReflectionFee
```

Recommendation

Emit an event for critical parameter changes.

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L706,713,721

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(taxFee_ >= 0,Tax fee must be greater or equal to zero)
require(bool,string)(burnFee_ >= 0,Burn fee must be greater or equal to zero)
require(bool,string)(newReflectionFee >= 0,Reflection fee must be greater or
equal to zero)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L915,948,958

Description

Performing divisions before multiplications may cause lose of prediction.

```
tTaxFee = tAmount * _taxFee / 100
tBurnFee = tAmount * _burnFee / 100
tFee = tAmount * _reflectionFee / 100
```

Recommendation

The multiplications should be prior to the divisions.

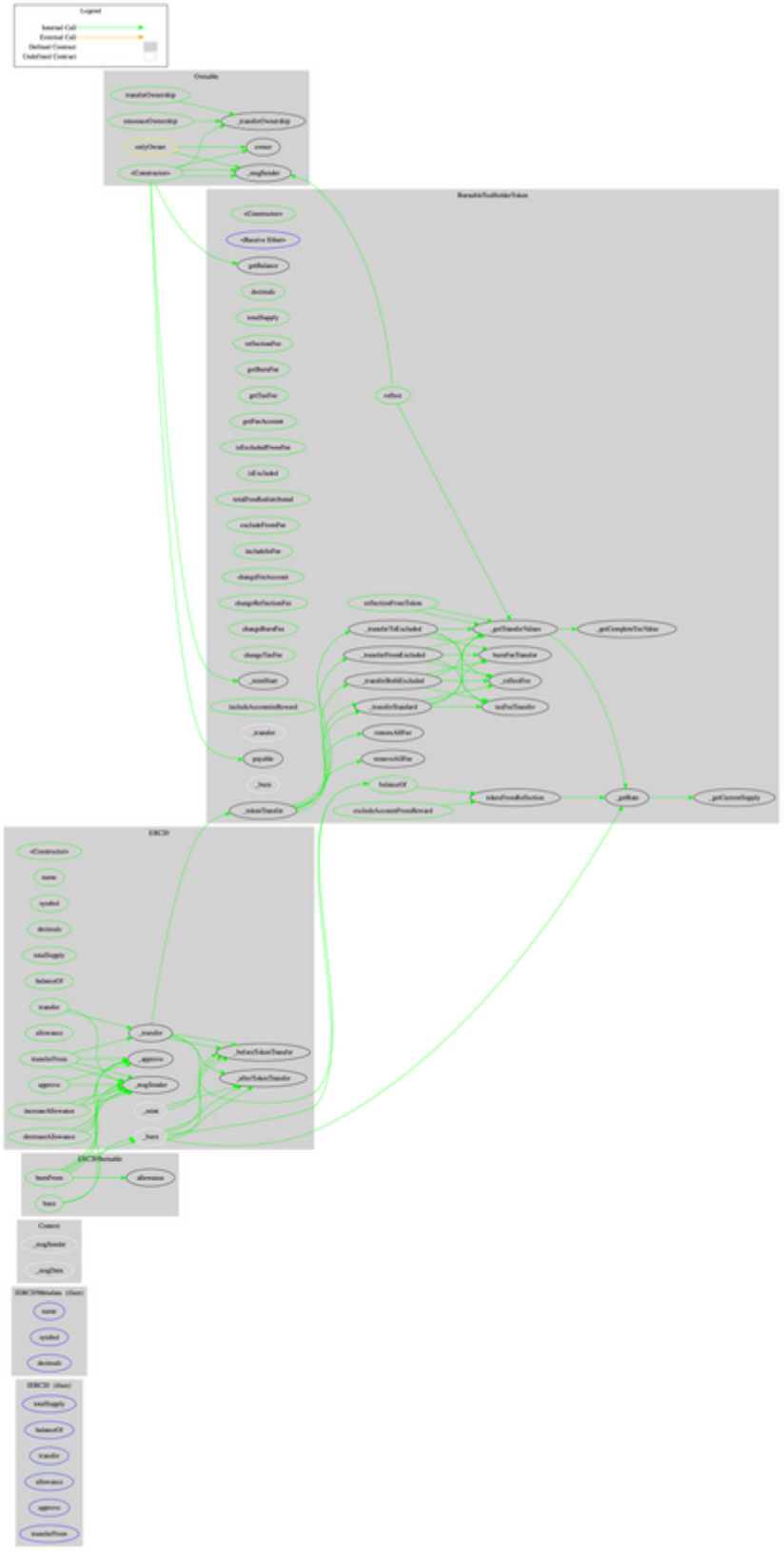
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
ERC20Burnable	Implementation	Context, ERC20		
	burn	Public	✓	-
	burnFrom	Public	✓	-
BurnableTaxHolderToken	Implementation	ERC20Burnable, Ownable		
	<Constructor>	Public	Payable	ERC20
	<Receive Ether>	External	Payable	-
	getBalance	Private		
	decimals	Public		-
	totalSupply	Public		-
	reflectionFee	Public		-
	getBurnFee	Public		-
	getTaxFee	Public		-
	getFeeAccount	Public		-
	isExcludedFromFee	Public		-

	balanceOf	Public		-
	isExcluded	Public		-
	totalFeesRedistributed	Public		-
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	changeFeeAccount	Public	✓	onlyOwner
	changeReflectionFee	Public	✓	onlyOwner
	changeBurnFee	Public	✓	onlyOwner
	changeTaxFee	Public	✓	onlyOwner
	_mintStart	Private	✓	
	reflect	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Private		
	excludeAccountFromReward	Public	✓	onlyOwner
	includeAccountinReward	Public	✓	onlyOwner
	_transfer	Internal	✓	
	_tokenTransfer	Private	✓	
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_getCompleteTaxValue	Private		
	_getTransferValues	Private		
	_reflectFee	Private	✓	
	_getRate	Private		
	_getCurrentSupply	Private		
	burnFeeTransfer	Private	✓	
	taxFeeTransfer	Private	✓	
	_burn	Internal	✓	

Contract Flow



Domain Info

Domain Name	MantisShrimpToken.com
Registry Domain ID	2655074560_DOMAIN_COM-VRSN
Creation Date	2021-11-15T05:28:32+00:00
Updated Date	2021-11-15T05:28:32+00:00
Registry Expiry Date	2022-11-15T05:28:32+00:00
Registrar WHOIS Server	whois.godaddy.com
Registrar URL	https://www.godaddy.com
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

There is no public billing information, the creator is protected by the privacy settings.

Summary

The Smart Contract analysis reported no critical issues. The contract owner can increase the fees to 30%. Apart from that, the contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. Additionally the contract is decreasing the total supply on every transaction if the "burn" fee is enabled.

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