Cyberscope

Audit Report

Phantom

March 2022

Type	ERC20
Network	FTM
Address	0x7853Ac81c1Cc9870B877ba6C05c614DA2a3a5548
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	PtmTOKEN
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	runs
Licence	MIT
Explorer	https://ftmscan.com/address/0x7853Ac81c1Cc9870B877ba6C05c614DA2a3a5548
Symbol	PtmTOKEN
Decimals	9
Total Supply	100,000,000
Domain	the-phantom-project.com

Source Files

Filename	**SHA256**
contract.sol	3ce3f608403ab8eaac3bb8ed969cf5936f6d14d3c09663cec1df2e15c296dbec

Audit Updates

Initial Audit	28th 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

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L1663,1694

Description

The contract owner has the authority to stop sales for all users excluding the owner. The owner may take advantage of it by setting the `maxSellTransactionAmount` or `_maxSellPercent` to zero. This will convert the contract into a **HONEYPOT**.

```
      if(automatedMarketMakerPairs[to]){
             require(amount <= maxSellTransactionAmount, "");
             amount = amount.mul(_maxSellPercent).div(100); // Maximum sell of 
99% per one single transaction, to ensure some loose change is left in the 
holders wallet .
        }
```

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `isTradingEnabled` to false.

```
    if(!isTradingEnabled) {
             require(_isAllowedDuringDisabled[to] || 
_isAllowedDuringDisabled[from], "");
        }
```

Recommendation

The contract could embody a check for not allowing setting the maxSellTransactionAmount and _maxSellPercent less 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.

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L1832

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `recoverContractFTM` function.

```
    function recoverContractFTM(uint256 recoverRate) public onlyOwner{
            uint256 ftmAmount = address(this).balance;
            if(ftmAmount > 0){
        sendToMarketingsWallet(ftmAmount.mul(recoverRate).div(100));
            }
    }
```

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#L1461

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 `setTaxFeePercent` function with a high percentage value.

```
    function updateFees(uint256 ftmRewardPerc, uint256 liquidityPerc, uint256
marketingPerc, uint256 buyBackPerc) external onlyOwner {
        // require (marketingPerc.add(buyBackPerc) <= liquidityPerc, "PtmTOKEN:
updateFees:: Liquidity Perc must be equal to or higher than marketings and
buyback combined.");
        emit FeesUpdated(ftmRewardPerc, liquidityPerc, marketingPerc,
buyBackPerc);
        FTMRewardsFee = ftmRewardPerc;
        liquidityFee = liquidityPerc;
        marketingFee = marketingPerc;
        buyBackFee= buyBackPerc;
        totalFees = FTMRewardsFee.add(liquidityFee);

    }
```

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1662

Description

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

```
require(!_isIgnoredAddress[to] || !_isIgnoredAddress[from], "");
```

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
●	CO	Code Optimization
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L11	Unnecessary Boolean equality
●	L12	Using Variables before Declaration
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope
●	L15	Local Scope Variable Shadowing

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1666

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

```
        if(!isTradingEnabled) {
            require(_isAllowedDuringDisabled[to] ||
_isAllowedDuringDisabled[from], "");
        }

        if(automatedMarketMakerPairs[to] && !isTradingEnabled &&
_isAllowedDuringDisabled[from]) {
            require((from == owner() || to == owner()) ||
_isAllowedDuringDisabled[from], "");
        }
```

Recommendation

The checks for isTradingEnabled on the second if statement are redundant as they have been checked previously. Same with the _isAllowedDuringDisabled[from] check.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L63,71,482,490,507,533,541,552,570,592 and 24 more

Description

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

```
process
getAccountAtIndex
size
getKeyAtIndex
getIndexOfKey
get
setMaxSellPercent
recoverContractFTM
activateContract
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L957,964,971,981,744,1036,1203,1204,1221,1246 and 5 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.

```
_account
_maxSellPercent
FTMRewardsFee
_holderLastSellDate
_isIgnoredAddress
_isAllowedDuringDisabled
MINIMUM_LIQUIDITY
PERMIT_TYPEHASH
DOMAIN_SEPARATOR
...
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L238

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1361,1383,1837

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.

```
_maxSellPercent = maxSellPercent
maxSellTransactionAmount = maxTxnAmount
swapTokensAtAmount = newAmount
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L991,284

Description

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

```
abs
_transfer
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L857,1589

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,string)(isAMMWhitelisted(ammContractAddress) == true,)
userHasCustomRewardToken[holder] == true
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1741

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.

```
iterations
lastProcessedIndex
claims
```

Recommendation

The variables should be declared before any usage of them.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1514,1651

Description

Performing divisions before multiplications may cause lose of prediction.

```
otherSellFee =
amount.mul(liquidityFee).div(100).mul(getHolderSellFactor(from)).div(100)
rewardSellFee =
amount.mul(FTMRewardsFee).div(100).mul(rewardFeeSellFactor).div(100)
amount = amount.mul(_maxSellPercent).div(100)
timeSinceLastSale =
(block.timestamp.sub(_holderLastSellDate[holder])).div(1209600)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1741,800

Description

The are variables that are defined in the local scope and are not initialized.

```
claims
iterations
swapSuccess
lastProcessedIndex
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L780,957,964,971,981

Description

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

```
_owner
_decimals
_symbol
_name
```

Recommendation

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

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-

DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-
	withdrawDividend	External	✓	-
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	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	✓	
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface, Ownable		
	updateDividendUniswapV2Router	External	✓	onlyOwner
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	swapETHForTokens	Private	✓	
	setIgnoreToken	External	✓	onlyOwner
	isIgnoredToken	Public		-
	getRawFTMDividends	External		-
	setWhiteListAMM	External	✓	onlyOwner
	setRewardToken	External	✓	onlyOwner
	unsetRewardToken	External	✓	onlyOwner
	distributeDividends	Public	Payable	-
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-

	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-

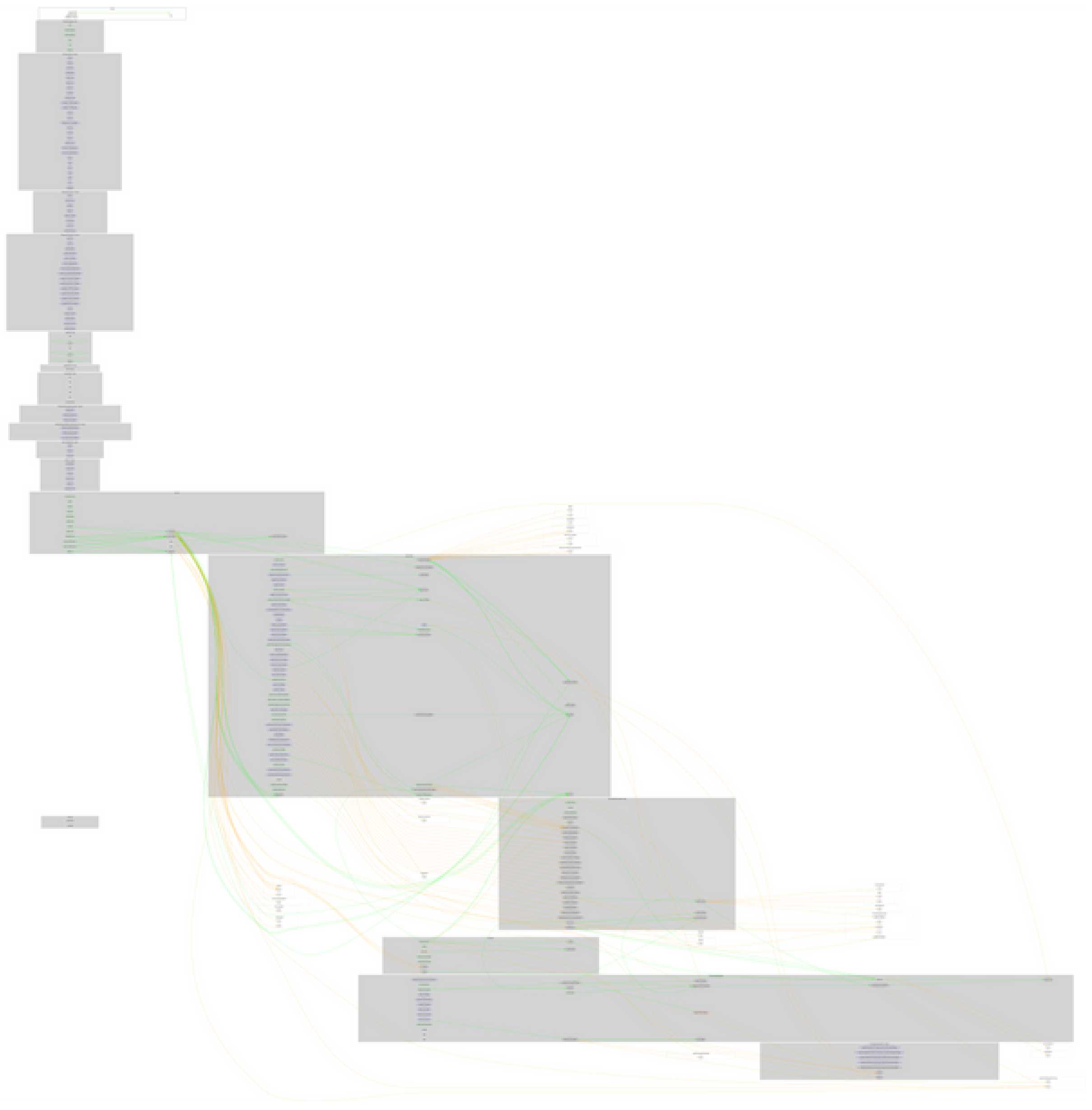
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-

	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
PtmTOKEN	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	setWhiteListAMM	External	✓	onlyOwner
	updateSwapTokensAtAmount	External	✓	onlyOwner
	disableTransferDelay	External	✓	onlyOwner
	updateDividendTracker	Public	✓	onlyOwner
	updateMaxTxn	External	✓	onlyOwner
	updateDividendTokensMinimum	External	✓	onlyOwner
	updateUniswapV2Router	External	✓	onlyOwner
	updateDividendUniswapV2Router	External	✓	onlyOwner
	updateTradingStatus	External	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	External	✓	onlyOwner
	addToWhitelist	External	✓	onlyOwner
	setIsBot	External	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	setAutomatedMarketMakerPair	External	✓	onlyOwner
	updateLiquidityWallet	External	✓	onlyOwner
	updateMarketingWallet	External	✓	onlyOwner
	updateBuyBackWallet	External	✓	onlyOwner
	updateFees	External	✓	onlyOwner
	updateGasForProcessing	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	setIgnoreToken	External	✓	onlyOwner

	isAMMWhitelisted	Public		-
	isContract	Internal		
	getUserCurrentRewardToken	Public		-
	getUserHasCustomRewardToken	Public		-
	getRewardTokenSelectionCount	Public		-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	getHolderSellFactor	Public		-
	getDividendTokensMinimum	External		-
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	getRawFTMDividends	Public		-
	getFTMAvailableForHolderBuyBack	Public		-
	isIgnoredToken	Public		-
	setRewardToken	Public	✓	-
	setRewardTokenWithCustomAMM	Public	✓	-
	unsetRewardToken	Public	✓	-
	activateContract	Public	✓	onlyOwner
	buyBackTokensWithNoFees	External	Payable	-
	claim	External	✓	-
	processDividendTracker	External	✓	-
	_setAutomatedMarketMakerPair	Private	✓	
	_transfer	Internal	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
	recoverContractFTM	Public	✓	onlyOwner
	sendToMarketingsWallet	Private	✓	
	setMaxSellPercent	Public	✓	onlyOwner

IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
PtmTOKENDividendTracker	Implementation	DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal		
	withdrawDividend	Public		-
	excludeFromDividends	External	✓	onlyOwner
	includeInDividends	External	✓	onlyOwner
	updateDividendMinimum	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	the-phantom-project.com
Registry Domain ID	2679648250_DOMAIN_COM-VRSN
Creation Date	2022-03-06T10:46:02Z
Updated Date	2022-03-06T10:47:27Z
Registry Expiry Date	2023-03-06T10:46:02Z
Registrar WHOIS Server	whois.webnic.cc
Registrar URL	webnic.cc
Registrar	WEBCC
Registrar IANA ID	460

The domain has been created 22 days before the creation of the audit. It will expire in 11 months.

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

Summary

Phantom Token is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees up to 100%, blacklisting users and transferring funds to the team's wallet. The owner can also convert the contract into a honeypot and prevent users from selling by abusing 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

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