Cyberscope

Audit Report

THE PEOPLEs MONEY revamped

March 2024

Network	MATIC
Address	0x52763CBAeecbaaa4649B7894e7C0a81F945Da1A3
Audited by	© cyberscope

Analysis

Critical Medium Minor / Informative Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Unresolved
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Table of Contents

Review

Contract Name	TPMrp
Compiler Version	v0.8.24+commit.e11b9ed9
Optimization	200 runs
Explorer	https://polygonscan.com/address/0x52763cbaeecbaaa4649b7894e7c0a81f945da1a3
Address	0x52763cbaeecbaaa4649b7894e7c0a81f945da1a3
Network	MATIC
Symbol	TPMrp
Decimals	18
Total Supply	1,000,000,000
Badge Eligibility	Yes

Audit Updates

Initial Audit	13 Mar 2024

Source Files

Filename	SHA256
TPMrp.sol	055fcbdf5651c4b490881a4b567247deb5cf978af4d50ddf11464662aabbd719

Findings Breakdown

Critical	0
Medium	0
Minor / Informative	1

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	1	0	0	0

MT - Mints Tokens

Criticality	Minor / Informative
Location	TPMrp.sol#L856,864,905
Status	Unresolved

Description

The contract implements a minting functionality that allows users to mint and claim their rewards daily. While the `mintAll` and `mintBetween` functions are restricted to the contract owner, ensuring a controlled distribution to multiple holders, the `mintIndividual` function allows individual users to mint rewards based on their token holdings. This functionality, although intended to incentivize holding, introduces potential risks associated with unchecked minting, such as inflation or manipulation of the token supply.

```
    function mintAll() public onlyOwner {
        // Ensure there's at least one real holder
        require(holdersForReward.length > 1, "No holders for reward.");

        // Mint to all holders, skipping the dummy entry at index 0
        _mintTo(1, holdersForReward.length - 1);
    }

    function mintBetween(uint256 start, uint256 end) public onlyOwner {
        _mintTo(start, end);
    }

    function mintIndividual() public {
        // Check if the sender is in the holders mapping
        require(
            holderIndex[msg.sender] > 0,
            "Address is not eligible for rewards."
        );

        // Calculate minting reward based on tokens held
        uint256 reward = calculateMintingReward(
            balanceOf(msg.sender),
            msg.sender
        );
        require(reward > 0, "Already minted today");

        //Mint the reward to user
        totalRewardsMinted += reward;
        _performMint(msg.sender, reward);
    }
```

Recommendation

It is recommended to undertake a comprehensive review of the minting mechanism's objectives and its implications on the token's economy. Additionally, 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.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
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		-

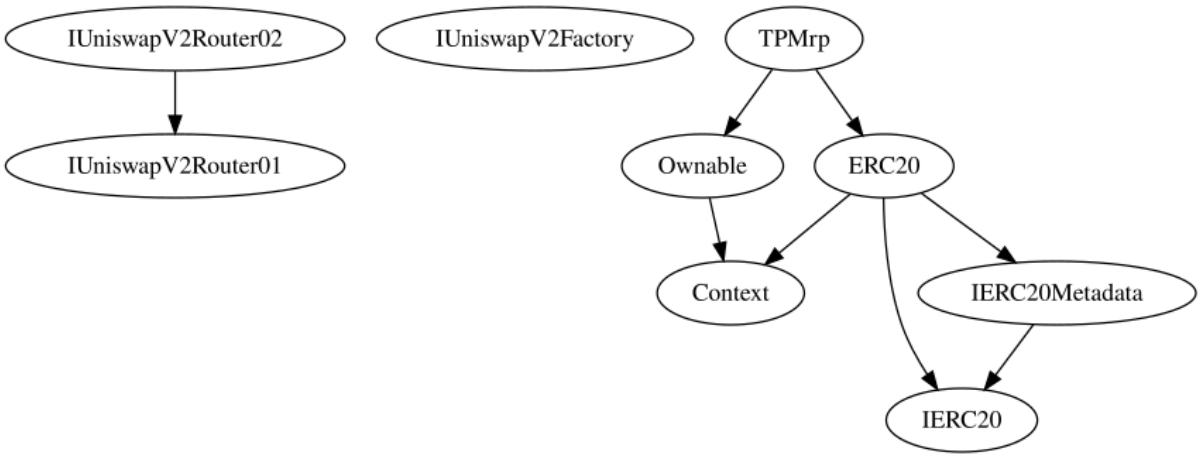
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2 Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		

	_contextSuffixLength	Internal		
Ownable	Implementation	Context		
		Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-

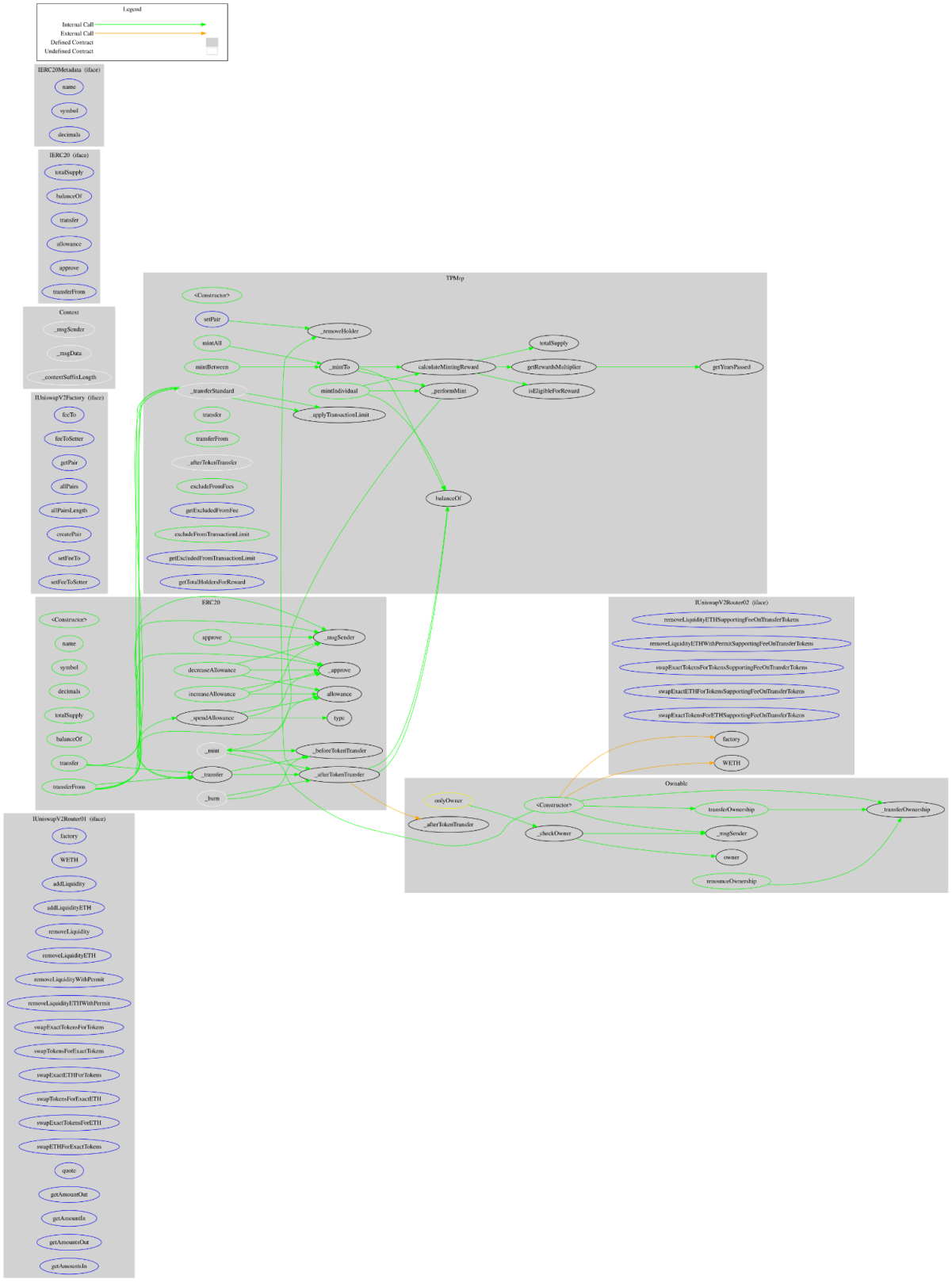
ERC20	Implementation	Context, IERC20, IERC20Metadata		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
TPMrp	Implementation	ERC20, Ownable		

		Public	✓	ERC20
	setPair	External	✓	onlyOwner
	mintAll	Public	✓	onlyOwner
	mintBetween	Public	✓	onlyOwner
	_mintTo	Internal	✓	
	mintIndividual	Public	✓	-
	_performMint	Internal	✓	
	calculateMintingReward	Internal	✓	
	isEligibleForReward	Public		-
	getRewardsMultiplier	Public		-
	getYearsPassed	Public		-
	_transferStandard	Internal	✓	
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	_applyTransactionLimit	Internal	✓	
	_afterTokenTransfer	Internal	✓	
	_removeHolder	Internal	✓	
	excludeFromFees	Public	✓	onlyOwner
	getExcludedFromFee	External		-
	excludeFromTransactionLimit	Public	✓	onlyOwner
	getExcludedFromTransactionLimit	External		-
	getTotalHoldersForReward	External		-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
IERC20Metadata (iface)
name
symbol
decimals
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
Context
_msgSender
_msgData
_contextSuffixLength
IUniswapV2Factory (iface)
feeTo
feeToSetter
getPair
allPairs
allPairsLength
createPair
setFeeTo
setFeeToSetter
TPMrp
<Constructor>
setPair
mintAll
mintBetween
_transferStandard
transfer
transferFrom
afterTokenTransfer
excludeFromFees
getExcludedFromFee
excludeFromTransactionLimit
getExcludedFromTransactionLimit
getTotalHoldersForReward
_removeHolder
_mintTo
mintIndividual
_applyTransactionLimit
calculateMintingReward
_performMint
totalSupply
getRewardsMultiplier
isEligibleForReward
getYearsPassed
balanceOf
ERC20
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
transfer
transferFrom
approve
decreaseAllowance
increaseAllowance
_spendAllowance
_mint
_transfer
_burn
_msgSender
_approve
allowance
type
_beforeTokenTransfer
_afterTokenTransfer
IUniswapV2Router02 (iface)
removeLiquidityETHSupportingFeeOnTransferTokens
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens
swapExactTokensForTokensSupportingFeeOnTransferTokens
swapExactETHForTokensSupportingFeeOnTransferTokens
swapExactTokensForETHSupportingFeeOnTransferTokens
factory
WETH
Ownable
onlyOwner
_afterTokenTransfer
<Constructor>
_checkOwner
transferOwnership
_msgSender
owner
renounceOwnership
_transferOwnership
IUniswapV2Router01 (iface)
factory
WETH
addLiquidity
addLiquidityETH
removeLiquidity
removeLiquidityETH
removeLiquidityWithPermit
removeLiquidityETHWithPermit
swapExactTokensForTokens
swapTokensForExactTokens
swapExactETHForTokens
swapTokensForExactETH
swapExactTokensForETH
swapETHForExactTokens
quote
getAmountOut
getAmountIn
getAmountsOut
getAmountsIn

Summary

THE PEOPLEs MONEY revamped contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. THE PEOPLEs MONEY revamped is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a 0.0001% fee on buy and sell transactions.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io