



Audit Report

PULI INU

January 2022

Type	BEP20
Network	BSC
Address	0xaef0a177c8c329cbc8508292bb7e06c00786bbfc
Audited by	© coinscope

Table of Contents

Table of Contents	1
Contract Review	3
Audit Updates	3
Contract Analysis	4
ST - Stop Transactions	5
Description	5
Recommendation	5
ELFM - Exceed Limit Fees Manipulation	6
Description	6
Recommendation	6
BC - Blacklisted Contracts	7
Description	7
Recommendation	7
Contract Diagnostics	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L09 - Dead Code Elimination	12
Description	12
Recommendation	12

L07 - Missing Events Arithmetic	13
Description	13
Recommendation	13
Contract Functions	14
Contract Flow	19
Domain Info	20
Summary	21
Disclaimer	22
About Coinscope	23

Contract Review

Contract Name	PULIINU
Compiler Version	v0.8.10+commit.fc410830
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xAEf0a177c8c329CBC8508292Bb7E06c00786BBfC
Symbol	PULI
Decimals	9
Total Supply	100,000,000
Source	contract.sol
Domain	puli-inu.com

Audit Updates

Initial Audit	26th January 2022
Corrected	17th May 2022

Contract Analysis

● Critical
 ● Medium
 ● Minor
 ● Pass

Severity	Code	Description	Status
●	ST	Contract Owner is not able to stop or pause transactions	Multi-Sign
●	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%)	Multi-Sign
●	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	Multi-Sign

ST - Stop Transactions

Criticality	critical
Location	https://bscscan.com/address/0xaef0a177c8c329cbc8508292bb7e06c00786bbfc#L1588
Status	Multi-Sign

Description

The contract owner has the authority to stop transactions for all users trying to sell their tokens and it will have a honeypot behaviour. The owner may take advantage of it by setting the `maxSellAmount` to zero.

```
if (recipient == uniswapV2Pair) {  
    require(amount <= maxSellAmount, "Sell exceeds limit");  
    setSell();  
}
```

The owner can also stop transactions for everyone by setting `maxWalletAmount` to the zero value

```
if(recipient != uniswapV2Pair){  
    require(  
        balanceOf(recipient) + amount <= maxWalletAmount,  
        "Balance exceeds limit"  
    );  
}
```

Recommendation

The contract could embody a check for not allowing setting the `maxSellAmount` less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

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.

Updated 17 May 2022

The team has acknowledged the threat and transferred the contract ownership to a multi-sign mechanism.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	https://bscscan.com/address/0xaef0a177c8c329cbc8508292bb7e06c00786bbfc#L1145
Status	Multi-Sign

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 `setSellFee` function with a high percentage value.

```
function setSellFee(uint16 liq, uint16 market, uint16 reward, uint16 tax)
external onlyOwner {
    sellFee.liquidityFee = liq;
    sellFee.marketingFee = market;
    sellFee.buybackFee = reward;
    sellFee.taxFee = 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.

BC - Blacklisted Contracts

Criticality	medium
Location	https://bscscan.com/address/0xaef0a177c8c329cbc8508292bb7e06c00786bbfc#L1182
Status	Multi-Sign

Description

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

```
function setBlackList(address addr, bool value) external onlyOwner {  
    _isBlackListed[addr] = 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.

Updated 17 May 2022

The team has acknowledged the threat and transferred the contract ownership to a multi-sign mechanism.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality

minor

Location

contract.sol#L1392,L1182,L1087 and 17 more

Description

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

```
isExcludedFromFee  
setSwapAndLiquifyEnabled  
excludeFromReward  
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L829,L828,L830

Description

Constant state variables should be declared constant to save gas.

```
_symbol  
_name  
_decimals
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L861,L820,L1357 and 6 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.

```
_marketingWallet  
_isBlackListed  
_amount  
...
```

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#L527,L539,L568 and 17 more

Description

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

```
trySub  
tryMul  
tryMod  
...
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1174,L1170,L1166 and 2 more

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.

```
maxWalletAmount = value * 10 ** _decimals
maxSellAmount = value * 10 ** _decimals
maxBuyAmount = value * 10 ** _decimals
...
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

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	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-

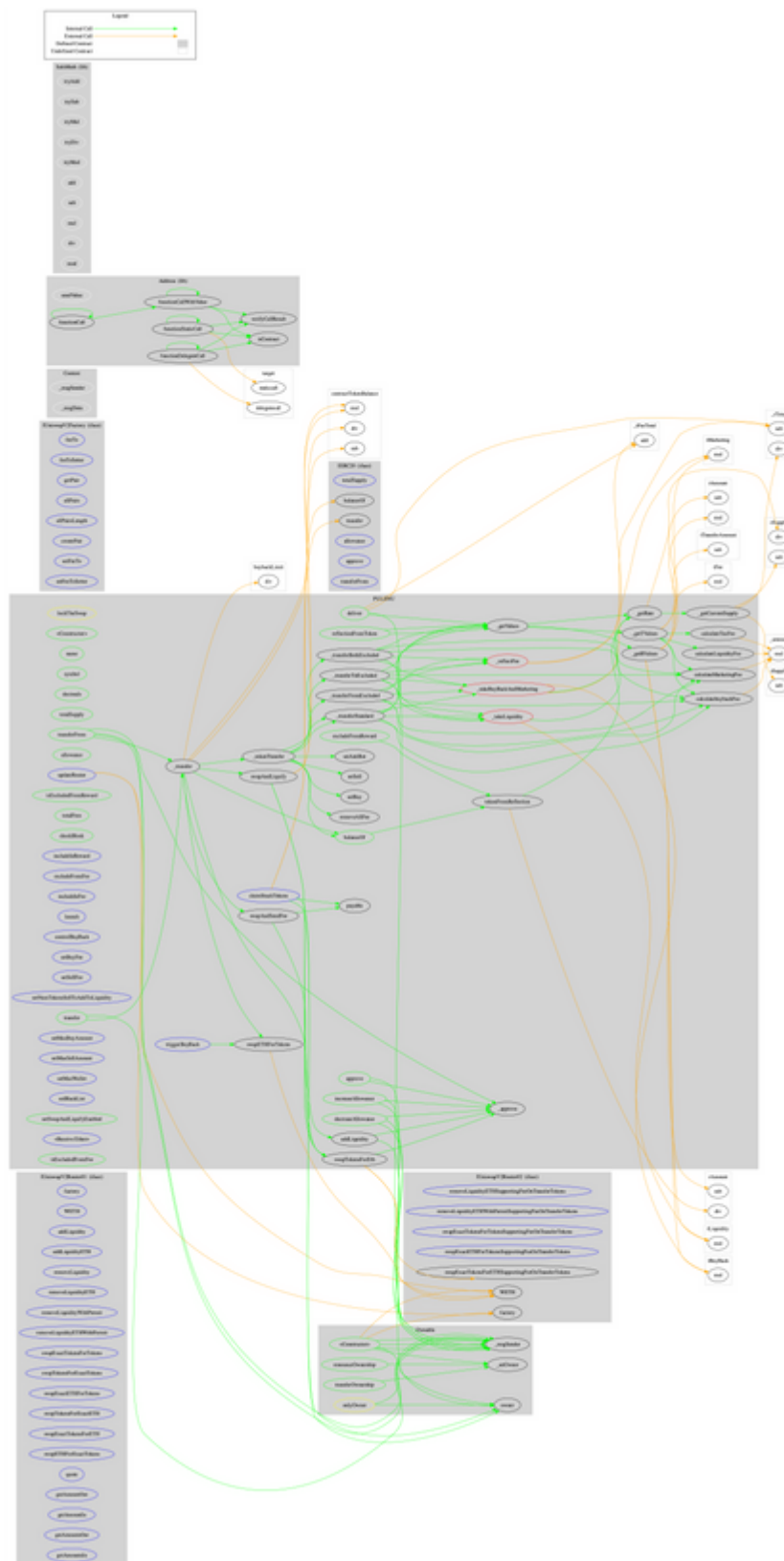
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		

	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
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		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
PULINU	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-

	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	checkBlock	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	excludeFromFee	External	✓	onlyOwner
	includeInFee	External	✓	onlyOwner
	launch	External	✓	onlyOwner
	controlBuyBack	External	✓	onlyOwner
	setBuyFee	External	✓	onlyOwner
	setSellFee	External	✓	onlyOwner
	setNumTokensSellToAddToLiquidity	External	✓	onlyOwner
	updateRouter	External	✓	onlyOwner
	setMaxBuyAmount	External	✓	onlyOwner
	setMaxSellAmount	External	✓	onlyOwner
	setMaxWallet	External	✓	onlyOwner
	setBlackList	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	triggerBuyBack	External	✓	onlyOwner
	claimStuckTokens	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		

	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	_takeBuyBackAndMarketing	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	calculatebuybackFee	Private		
	calculateMarketingFee	Private		
	removeAllFee	Private	✓	
	setBuy	Private	✓	
	setSell	Private	✓	
	setAntiBot	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Private	✓	
	swapAndSendFee	Private	✓	
	swapTokensForEth	Private	✓	lockTheSwap
	swapETHForTokens	Private	✓	lockTheSwap
	addLiquidity	Private	✓	lockTheSwap
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	

Contract Flow



Domain Info

Domain Name	puli-inu.com
Registry Domain ID	2665936265_DOMAIN_COM-VRSN
Creation Date	2022-01-04T16:23:05Z
Updated Date	2022-01-04T16:23:06Z
Registry Expiry Date	2023-01-04T16:23:05Z
Registrar WHOIS Server	whois.wix.com
Registrar URL	http://www.wix.com
Registrar	Wix.com Ltd.
Registrar IANA ID	3817

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

PULI INU is an interesting project with a friendly and growing community. The Smart Contract analysis reported no compiler error, but a few critical issues. There are some functions that can be abused by the owner, like manipulating fees up to 100%, blacklisting wallets from selling and stopping all transactions. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Updated 17 May 2022

The team has acknowledged the threats and transferred the contract ownership to a multi-sign mechanism.

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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.



The Coinscope.co team

<https://www.coinscope.co>



Cyberscope

Audit Report

Puli Payroll

April 2022

Type ERC20

Network RINKEBY

Address 0x4feB43230508384350e2b50914355E29Bc8c1098

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
Total Shares Concurrency	5
Description	5
Recommendation	5
Minimum Fund Distribution	6
Description	6
Recommendation	6
Redundant Checks	7
Description	7
Recommendation	7
Contract Diagnostics	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L04 - Conformance to Solidity Naming Conventions	10
Description	10
Recommendation	10
L09 - Dead Code Elimination	11
Description	11
Recommendation	11
L11 - Unnecessary Boolean equality	12
Description	12

Recommendation	12
L12 - Using Variables before Declaration	13
Description	13
Recommendation	13
Contract Functions	14
Contract Flow	16
Summary	17
Disclaimer	18
About Cyberscope	19

Contract Review

Contract Name	Payroll
Compiler Version	v0.8.1+commit.df193b15
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x4feB43230508384350e2b50914355E29Bc8c1098

Source Files

Filename	SHA256
contract.sol	2bc5bd3e58c24cd938888b21d38badf82217ee203487435048c3bcf66e99937b

Audit Updates

Initial Audit	11th April 2022
Corrected	

Contract Analysis

The contract implements a vesting-related feature where users are holding a node and get rewards proportionally to the node ratio. In Payroll terminology, the nodes are called departments, the vesting accounts are the head of the department, and the node ratio are the shares.

- A new department can be added only if all the current heads sign.
- One department head can only be owned by one address. So a single address can not hold more than one department.
- Anyone can distribute funds to the contract. Each department will be fueled with funds proportionally to the shares.
- The funds can either be the native currency or any other token.
- The head of each department can withdraw the proportional funds of the owned department any time.
- A department head can move the ownership to another address, only if the uniqueness rule is fulfilled.
- Each department role has the authority to increase the shares of the owned department only if it is signed by all the department heads.
- A department head can remove a department only if it is signed by all the department heads except one.
- The remove process will delete the department and add the share equally to the rest departments.

Total Shares Concurrency

Criticality	critical
Location	contract.sol#L716,560

Description

The contract is using the total shares as a divisor to distribute the shares for each department. Thus the totalShare property should always mirror the sum of the share of all the departments. There are two segments where this synchronisation is not happening:

1. The `updateShares()` method increases the shares of a specific department, but the `totalShares` is not updated.
2. The `removeDepartment()` method moves the shares of the department that is going to be removed to the rest departments. The calculation of the shared distribution is a result of division. That means that if the result is decimal in some deviations then the shared amount will not be precise. Hence, the sum of the share of all the departments will be different in comparison with the `totalShares`.

```
for (uint8 i = 0; i < departmentNames.length; i++) {  
    Department memory department = departments[departmentNames[i]];   
    uint256 addedAmount = (_amount * department.shares) / totalShares;  
    balances[departmentNames[i]][_token] += addedAmount;  
}
```

Recommendation

The contract should correctly update the `totalShares` variable when the `departments.share` property is updated.

Minimum Fund Distribution

Criticality	minor
Location	contract.sol#L752

Description

The contract deposit of native currency or pure token distributes the amount proportionally to the shares of each department. The calculation of the shared amount is a result of division. That means that if the provided amount is less than a threshold, there will be departments that will receive zero amount.

```
function _distributeTokens(address _token, uint256 _amount) internal {  
    for (uint8 i = 0; i < departmentNames.length; i++) {  
        Department memory department = departments[departmentNames[i]];   
        uint256 addedAmount = (_amount * department.shares) / totalShares;  
        balances[departmentNames[i]][_token] += addedAmount;  
    }  
}
```

Recommendation

The contract could embed a check for now allowing the transaction to proceed if there is at least one department that receives zero funds.

Redundant Checks

Criticality	minor
Location	contract.sol#L696

Description

The contract contains some tautology checks. That means that if one check is fulfilled then the other check will be fulfilled by definition.

```
require(  
    departments[_departmentName].head != address(0),  
    "Department does not exists"  
);  
require(  
    departments[_departmentName].head == _msgSender(),  
    "Sender is not head of the department"  
);
```

Recommendation

The contract could eliminate the tautology checks.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L11	Unnecessary Boolean equality
●	L12	Using Variables before Declaration

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L525,529,560,630,634,690,716

Description

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

```
updateShares  
transferOwnership  
releasedOf  
removeDepartment  
addDepartment  
getAllDepartmentNames
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L530,531,532,533,535,561,562,563,565,614,620,630,634,690,717,718,719,720,721,477

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.

```
maxDepartments
_signatures
_data
_reason
_changeInShares
_departmentName
_tokenAddresses
_shares
_head
...
```

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#L344,395,428,441,184,200,159

Description

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

```
toString  
toHexString  
toTypedDataHash  
toEthSignedMessageHash  
recover
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality

minor

Location

contract.sol#L760

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)(signatureRegister[signatures[i]] == false,Signature already registered)
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality

minor

Location

contract.sol#L276

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

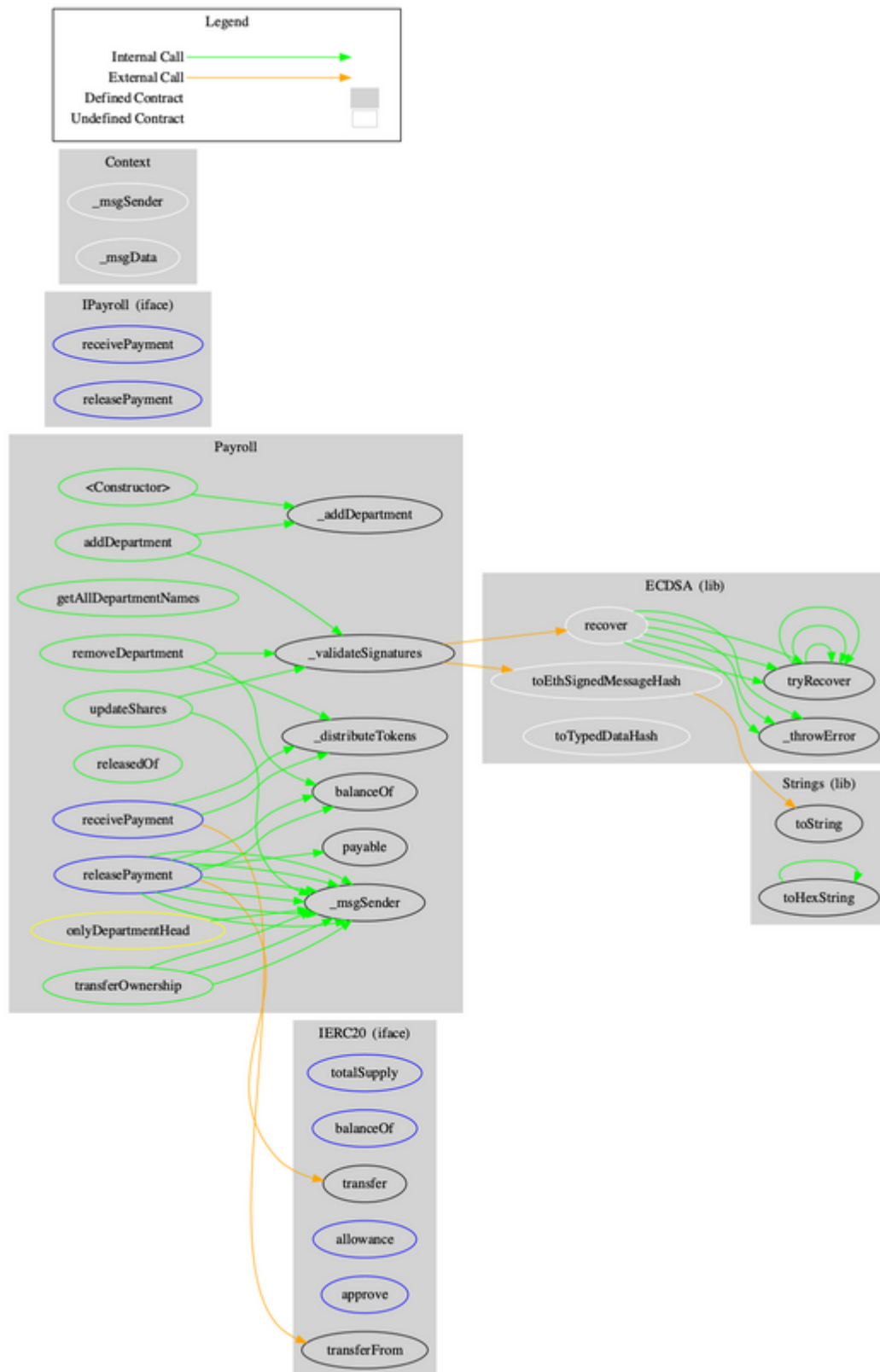
The variables should be declared before any usage of them.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IPayroll	Interface			
	receivePayment	External	Payable	-
	receivePayment	External	✓	-
	releasePayment	External	Payable	-
	releasePayment	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	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		
Payroll	Implementation	IPayroll, Context		
	<Constructor>	Public	✓	-
	getAllDepartmentNames	Public		-
	addDepartment	Public	✓	onlyDepartmentHead
	removeDepartment	Public	✓	onlyDepartmentHead
	balanceOf	Public		-
	balanceOf	Public		-
	releasedOf	Public		-
	releasedOf	Public		-
	receivePayment	External	Payable	-
	receivePayment	External	✓	-
	releasePayment	External	Payable	onlyDepartmentHead
	releasePayment	External	✓	onlyDepartmentHead
	transferOwnership	Public	✓	onlyDepartmentHead
	updateShares	Public	✓	onlyDepartmentHead
	_distributeTokens	Internal	✓	
	_validateSignatures	Internal	✓	
	_addDepartment	Internal	✓	

Contract Flow



Summary

Puli Payroll is a staking variation contract. The users own nodes and get rewards according to the node's power. In this audit we investigate the business logic, we mention some potential vulnerabilities and we suggest potential improvements.

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.

Cyberscope 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>



Cyberscope

Audit Report

Puli Runners

April 2022

Type	ERC20
Network	RINKEBY
Address	0x5341e6fccc30Cba7b9bf032C9164913009Bd8ED0
Audited by	© cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
Contract Owner Privileges	4
Whitelisted Maximum Quantity Check	5
Description	5
Recommendation	5
Total Supply Inflation	6
Description	6
Recommendation	6
Remove Whitelist Naming Convention	7
Description	7
Recommendation	7
Contract Diagnostics	8
CO - Code Optimization	9
Description	9
Recommendation	9
MC - Missing Check	10
Description	10
Recommendation	10
L01 - Public Function could be Declared External	11
Description	11
Recommendation	11
L04 - Conformance to Solidity Naming Conventions	12

Description	12
Recommendation	12
L09 - Dead Code Elimination	13
Description	13
Recommendation	13
L12 - Using Variables before Declaration	14
Description	14
Recommendation	14
L15 - Local Scope Variable Shadowing	15
Description	15
Recommendation	15
Contract Functions	16
Contract Flow	22
Summary	23
Disclaimer	24
About Cyberscope	25

Contract Review

Contract Name	PuliRunners
Compiler Version	v0.8.1+commit.df193b15
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x5341e6fccc30Cba7b9bf032C9164913009Bd8ED0
Symbol	PRNR
Decimals	0
Total Supply	2
Domain	pulitoken.net

Source Files

Filename	SHA256
contract.sol	7a7e84ded3a7542f7e53f9b97e15ad2030068a57a05d69405a17793d4265f6f8

Audit Updates

Initial Audit	11th April 2022
Corrected	

Contract Analysis

- The contract owner can whitelist addresses to buy nfts.
- A whitelist describes the quantity and the price that each whitelisted user can buy.
- The price can be either tokens or the native currency.
- The whitelisted users can buy NFTs by either paying in the native currency or in specific tokens that have been defined by the whitelist descriptor.
- The non-whitelisted users can buy NFTs by either paying in the native currency or in specific tokens that have been defined by the general price list.

Contract Owner Privileges

- The contract owner has the authority to mint NFTs to any address.
- The contract owner has the authority to mint the remaining whitelisted NFTs to the owner's address.

Whitelisted Maximum Quantity Check

Criticality	minor
Location	contract.sol#L1941

Description

The contract does not check if the quantity of NFTs that is going to be minted is less than the maximum allowed. As a result, the whitelisted users counter will underflow.

```
whitelistMap[batchDescription].users[_msgSender()]--;
```

Recommendation

The contract should check if the quantity that is going to be minted by the whitelisted users is less than the maximum allowed.

Total Supply Inflation

Criticality	minor
Location	contract.sol#L1971,1987

Description

The contract contains a maximum amount of the total NFTs that can be minted. This number is fixed to 2010. The regular buys that are not whitelisted do not check if the total supply has reached the maximum allowed supply.

```
function mint(uint256 quantity) public payable {
    require(price[NATIVE] > 0, "Price is not set for minting");
    require(
        msg.value >= price[NATIVE] * quantity,
        "Insufficient amount, cannot mint"
    );

    for (uint256 index = 0; index < quantity; index++) {
        _safeMint(_msgSender(), tokenIdCounter);
        totalSupply++;
        emit Mint(msg.sender, tokenIdCounter++);
    }
}
```

Recommendation

The contract should check if the maximum allowed supply has been reached.

Remove Whitelist Naming Convention

Criticality

minor

Location

contract.sol#L2024

Description

The `removeWhitelist` method mints all the remaining NFTs of a whitelisted user to the contract owner. These NFTs have not minted yet, it is registered to the user's whitelist as a potential buy. Hence, the method does not solely remove the whitelist from the users, but it does mint the NFTs to the contract owner as well.

```
_safeMint(_msgSender(), tokenId); //Remove whitelist and send to owner
```

Recommendation

The method could have a more representative name according to the business logic or the implementation could mirror the expected functionality.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	CO	Code Optimization
●	MC	Missing Check
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L12	Using Variables before Declaration
●	L15	Local Scope Variable Shadowing

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1958,1980,1994,2047

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.

The contract contains loops that increase the variables on every iteration. This process produces $O(n)$ total computation at the end of the loop.

```
for (uint256 index = 0; index < quantity; index++) {
    uint256 tokenId = whiteListMap[batchDescription].start +
        whiteListMap[batchDescription].counter;

    _safeMint(_msgSender(), tokenId);

    whiteListMap[batchDescription].users[_msgSender()]--; // Decrement counter
    for specific user
    whiteListMap[batchDescription].counter++; // Increment batch counter
    totalSupply++;
    emit WhiteListMint(msg.sender, tokenId);
}
```

Recommendation

The total increased value could be added once outside the loop scope. As a result, complexity will be decreased to $O(1)$.

In this example the improvement could be applied to the `whiteListMap[batchDescription].users[_msgSender()]`, `whiteListMap[batchDescription].counter` and `totalSupply` variables.

MC - Missing Check

Criticality

minor

Location

contract.sol#L1937,1978,2176

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.

```
token.transferFrom(msgSender(), address(this), quantity * _price);
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L737,754,761,768,787,811,825,839,1179,1187,1637,1689,1785,1787,1828,1857,1881,1902,1922,1971,1987,2001,2007,2024,2130,2144,2150

Description

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

```
withdraw
setWithdrawalAddress
setBaseTokenURI
removeWhitelist
getWhitelistPrice
getPrice
mint
mintByToken
mintWhitelistToken
...
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L854,1787,1816,2130,2144

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.

```
_withdrawalAddress  
_baseToken  
_tokenId  
_to  
_data
```

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#L406,416,495,505,468,478,1226,1236,1230,1244,780,979,613,629

Description

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

```
toHexString  
_burn  
_baseURI  
reset  
increment  
decrement  
current  
functionStaticCall  
functionDelegateCall  
...
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality

minor

Location

contract.sol#L1070,1072

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.

```
reason
retval
```

Recommendation

The variables should be declared before any usage of them.

L15 - Local Scope Variable Shadowing

Criticality

minor

Location

contract.sol#L1772,1773

Description

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

```
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
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC165	Interface			
	supportsInterface	External		-
IERC721	Interface	IERC165		
	balanceOf	External		-
	ownerOf	External		-
	safeTransferFrom	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	getApproved	External		-
	setApprovalForAll	External	✓	-
	isApprovedForAll	External		-
	safeTransferFrom	External	✓	-
IERC721Receiver	Interface			
	onERC721Received	External	✓	-
IERC721Metadata	Interface	IERC721		
	name	External		-

	symbol	External		-
	tokenURI	External		-
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
ERC721	Implementation	Context, ERC165, IERC721, IERC721Me tadata		
	<Constructor>	Public	✓	-
	supportsInterface	Public		-
	balanceOf	Public		-
	ownerOf	Public		-

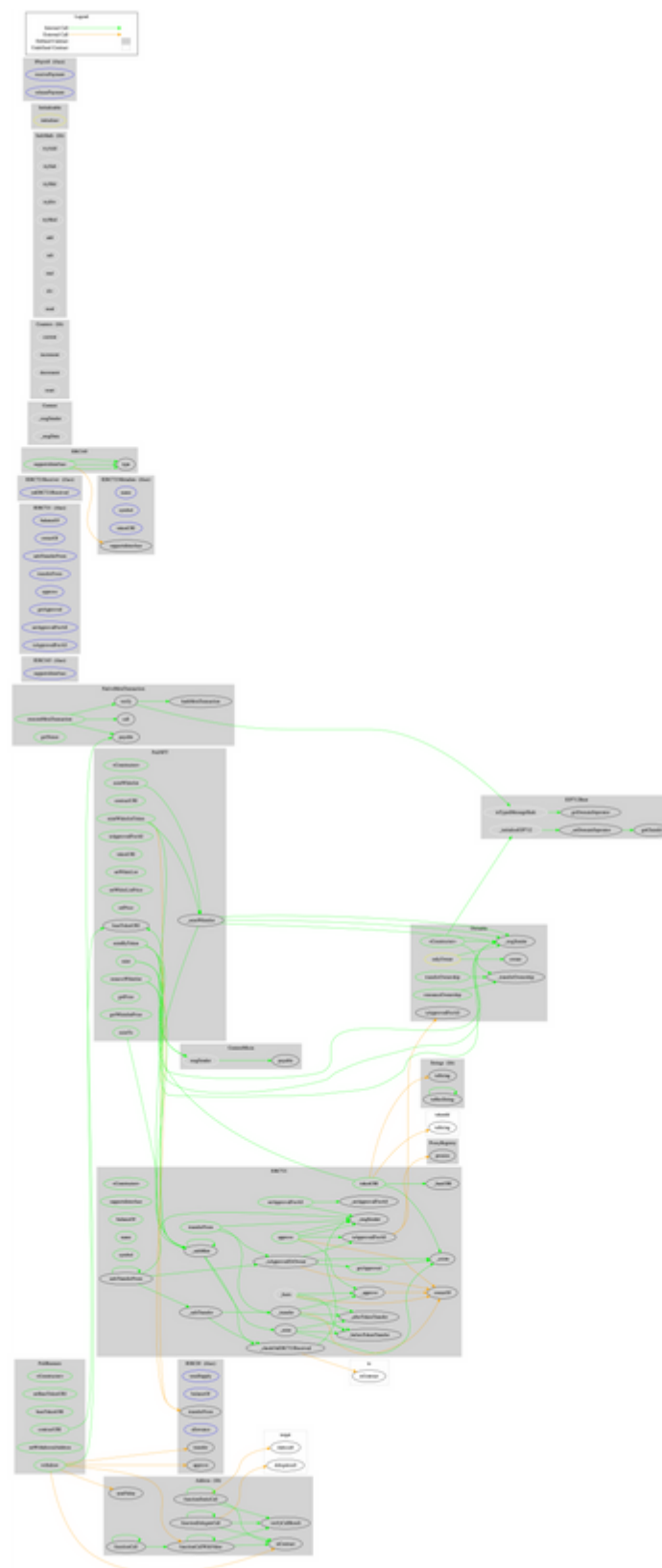
	name	Public		-
	symbol	Public		-
	tokenURI	Public		-
	_baseURI	Internal		
	approve	Public	✓	-
	getApproved	Public		-
	setApprovalForAll	Public	✓	-
	isApprovedForAll	Public		-
	transferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	safeTransferFrom	Public	✓	-
	_safeTransfer	Internal	✓	
	_exists	Internal		
	_isApprovedOrOwner	Internal		
	_safeMint	Internal	✓	
	_safeMint	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_transfer	Internal	✓	
	_approve	Internal	✓	
	_setApprovalForAll	Internal	✓	
	_checkOnERC721Received	Private	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
Counters	Library			
	current	Internal		
	increment	Internal	✓	

	decrement	Internal	✓	
	reset	Internal	✓	
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		
ContextMixin	Implementation			
	msgSender	Internal		
Initializable	Implementation			
EIP712Base	Implementation	Initializable		
	_initializeEIP712	Internal	✓	initializer
	_setDomainSeperator	Internal	✓	
	getDomainSeperator	Public		-
	getChainId	Public		-
	toTypedMessageHash	Internal		
NativeMetaTransaction	Implementation	EIP712Base		
	executeMetaTransaction	Public	Payable	-
	hashMetaTransaction	Internal		
	getNonce	Public		-
	verify	Internal		

OwnableDelegateProxy	Implementation			
ProxyRegistry	Implementation			
PuliNFT	Implementation	ERC721, ContextMixin, NativeMeta Transaction, Ownable		
	<Constructor>	Public	✓	ERC721
	baseTokenURI	Public		-
	contractURI	Public		-
	mintTo	Public	✓	onlyOwner
	isApprovedForAll	Public		-
	tokenURI	Public		-
	setWhiteList	Public	✓	onlyOwner
	setWhiteListPrice	Public	✓	onlyOwner
	setPrice	Public	✓	onlyOwner
	mintWhitelist	Public	Payable	-
	mintWhitelistToken	Public	✓	-
	_mintWhitelist	Internal	✓	
	mintByToken	Public	✓	-
	mint	Public	Payable	-
	getPrice	Public		-
	getWhitelistPrice	Public		-
	removeWhitelist	Public	✓	onlyOwner
IPayroll	Interface			
	receivePayment	External	Payable	-
	receivePayment	External	✓	-
	releasePayment	External	Payable	-
	releasePayment	External	✓	-
PuliRunners	Implementation	PuliNFT		

	<Constructor>	Public	✓	PuliNFT
	setBaseTokenURI	Public	✓	onlyOwner
	baseTokenURI	Public		-
	contractURI	Public		-
	setWithdrawalAddress	Public	✓	onlyOwner
	withdraw	Public	Payable	onlyOwner

Contract Flow



Summary

Puli Runners is an NFT contract that gives the ability to the users to buy NFT under a specific price. There are whitelisted users that can buy a specific amount of NFTs at a different price. The price could be either in the native currency or in tokens. This audit focuses on the business logic flow, security concerns and potential improvements.

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.

Cyberscope 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>



Audit Report

PULI INU

January 2022

Type	BEP20
Network	BSC
Address	0xaef0a177c8c329cbc8508292bb7e06c00786bbfc
Audited by	© coinscope

Table of Contents

Table of Contents	1
Contract Review	3
Audit Updates	3
Contract Analysis	4
ST - Stop Transactions	5
Description	5
Recommendation	5
ELFM - Exceed Limit Fees Manipulation	6
Description	6
Recommendation	6
BC - Blacklisted Contracts	7
Description	7
Recommendation	7
Contract Diagnostics	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L09 - Dead Code Elimination	12
Description	12
Recommendation	12

L07 - Missing Events Arithmetic	13
Description	13
Recommendation	13
Contract Functions	14
Contract Flow	19
Domain Info	20
Summary	21
Disclaimer	22
About Coinscope	23

Contract Review

Contract Name	PULIINU
Compiler Version	v0.8.10+commit.fc410830
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xAEf0a177c8c329CBC8508292Bb7E06c00786BBfC
Symbol	PULI
Decimals	9
Total Supply	100,000,000
Source	contract.sol
Domain	puli-inu.com

Audit Updates

Initial Audit	26th January 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	https://bscscan.com/address/0xaef0a177c8c329cbc8508292bb7e06c00786bbfc#L1588

Description

The contract owner has the authority to stop transactions for all users trying to sell their tokens and it will have a honeypot behaviour. The owner may take advantage of it by setting the `maxSellAmount` to zero.

```
if (recipient == uniswapV2Pair) {  
    require(amount <= maxSellAmount, "Sell exceeds limit");  
    setSell();  
}
```

The owner can also stop transactions for everyone by setting `maxWalletAmount` to the zero value

```
if(recipient != uniswapV2Pair){  
    require(  
        balanceOf(recipient) + amount <= maxWalletAmount,  
        "Balance exceeds limit"  
    );  
}
```

Recommendation

The contract could embody a check for not allowing setting the `maxSellAmount` less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

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	https://bscscan.com/address/0xaef0a177c8c329cbc8508292bb7e06c00786bbfc#L1145

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 `setSellFee` function with a high percentage value.

```
function setSellFee(uint16 liq, uint16 market, uint16 reward, uint16 tax)
external onlyOwner {
    sellFee.liquidityFee = liq;
    sellFee.marketingFee = market;
    sellFee.buybackFee = reward;
    sellFee.taxFee = 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.

BC - Blacklisted Contracts

Criticality	medium
Location	https://bscscan.com/address/0xaef0a177c8c329cbc8508292bb7e06c00786bbfc#L1182

Description

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

```
function setBlackList(address addr, bool value) external onlyOwner {  
    _isBlackListed[addr] = 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.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L07	Missing Events Arithmetic

L01 - Public Function could be Declared External

Criticality

minor

Location

contract.sol#L1392,L1182,L1087 and 17 more

Description

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

```
isExcludedFromFee  
setSwapAndLiquifyEnabled  
excludeFromReward  
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality

minor

Location

contract.sol#L829,L828,L830

Description

Constant state variables should be declared constant to save gas.

```
_symbol  
_name  
_decimals
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L861,L820,L1357 and 6 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.

```
_marketingWallet  
_isBlackListed  
_amount  
...
```

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#L527,L539,L568 and 17 more

Description

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

```
trySub  
tryMul  
tryMod  
...
```

Recommendation

Remove unused functions.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1174,L1170,L1166 and 2 more

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.

```
maxWalletAmount = value * 10 ** _decimals
maxSellAmount = value * 10 ** _decimals
maxBuyAmount = value * 10 ** _decimals
...
```

Recommendation

Emit an event for critical parameter changes.

Contract Functions

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	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-

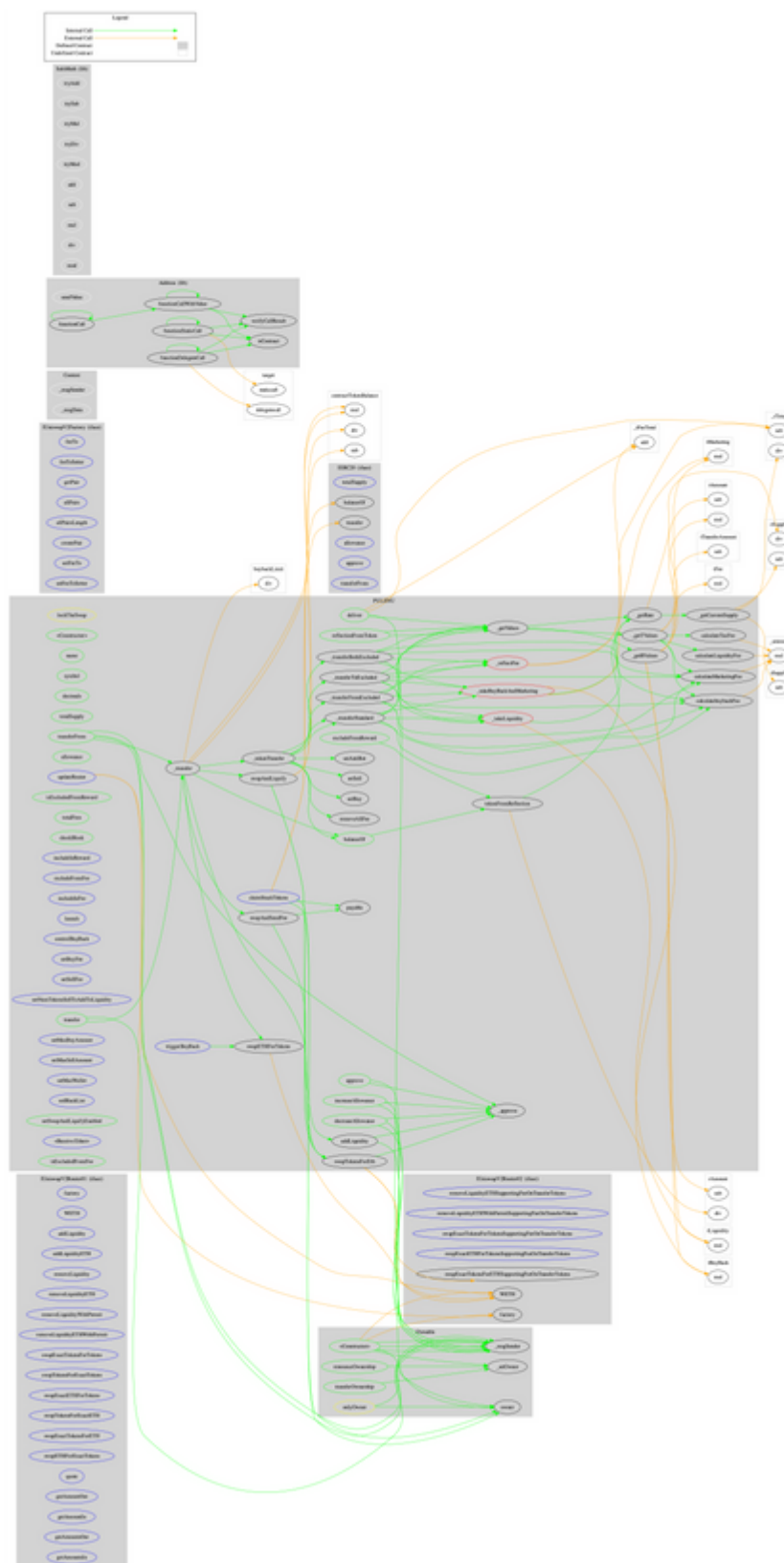
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		

	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
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		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
PULINU	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-

	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	checkBlock	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	excludeFromFee	External	✓	onlyOwner
	includeInFee	External	✓	onlyOwner
	launch	External	✓	onlyOwner
	controlBuyBack	External	✓	onlyOwner
	setBuyFee	External	✓	onlyOwner
	setSellFee	External	✓	onlyOwner
	setNumTokensSellToAddToLiquidity	External	✓	onlyOwner
	updateRouter	External	✓	onlyOwner
	setMaxBuyAmount	External	✓	onlyOwner
	setMaxSellAmount	External	✓	onlyOwner
	setMaxWallet	External	✓	onlyOwner
	setBlackList	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	triggerBuyBack	External	✓	onlyOwner
	claimStuckTokens	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		

	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	_takeBuyBackAndMarketing	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	calculatebuybackFee	Private		
	calculateMarketingFee	Private		
	removeAllFee	Private	✓	
	setBuy	Private	✓	
	setSell	Private	✓	
	setAntiBot	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Private	✓	
	swapAndSendFee	Private	✓	
	swapTokensForEth	Private	✓	lockTheSwap
	swapETHForTokens	Private	✓	lockTheSwap
	addLiquidity	Private	✓	lockTheSwap
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	

Contract Flow



Domain Info

Domain Name	puli-inu.com
Registry Domain ID	2665936265_DOMAIN_COM-VRSN
Creation Date	2022-01-04T16:23:05Z
Updated Date	2022-01-04T16:23:06Z
Registry Expiry Date	2023-01-04T16:23:05Z
Registrar WHOIS Server	whois.wix.com
Registrar URL	http://www.wix.com
Registrar	Wix.com Ltd.
Registrar IANA ID	3817

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

PULI INU is an interesting project with a friendly and growing community. The Smart Contract analysis reported no compiler error, but a few critical issues. There are some functions that can be abused by the owner, like manipulating fees up to 100%, blacklisting wallets from selling and stopping all transactions. 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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.



The Coinscope.co team

<https://www.coinscope.co>