Cyberscope

Audit Report

ISX INVESTING

August 2022

Type	BEP20
Network	BSC
Address	0x21b61a2fcc08d1a246360471787dab72c8fafd6b
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	ProMax
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x21b61a2fcc08d1a246360471787dab72c8fafd6b
Symbol	ISX
Decimals	7
Total Supply	600,000,000
Domain	https://islix.digital

Source Files

Filename	**SHA256**
contract.sol	0bf51ce8700e2f3297159879ac31f9003d0765d215e5ff5673677c31a71ab9c2

Audit Updates

Initial Audit	4th August 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	critical
Location	contract.sol#L1484

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 ‘setAllFeePercentfunction’ with the highest acceptable value for each argument. As a result, the fees could be set up to 60%.

```
function setAllFeePercent(
        uint8 taxFee,
        uint8 liquidityFee,
        uint8 burnFee,
        uint8 walletFee,
        uint8 buybackFee,
        uint8 walletCharityFee,
        uint8 rewardFee
    ) external onlyOwner {
        require(taxFee >= 0 && taxFee <= maxTaxFee, "TF err");
        require(liquidityFee >= 0 && liquidityFee <= maxLiqFee, "LF err");
        require(burnFee >= 0 && burnFee <= maxBurnFee, "BF err");
        require(walletFee >= 0 && walletFee <= maxWalletFee, "WF err");
        require(buybackFee >= 0 && buybackFee <= maxBuybackFee, "BBF err");
        require(
            walletCharityFee >= 0 && walletCharityFee <= maxWalletFee,
            "WFT err"
        );
        require(rewardFee >= 0 && rewardFee <= maxTaxFee, "RF err");
        //both tax fee and reward fee cannot be set
        require(rewardFee == 0 || taxFee == 0, "RT fee err");
```

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#L2536

Description

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

```
function blacklistAddress(address account, bool value) external onlyOwner {
        _isBlacklisted[account] = 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
	CO	Code Optimization
	CR	Code Repetition
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L07	Missing Events Arithmetic
	L08	Tautology or Contradiction
	L09	Dead Code Elimination
	L13	Divide before Multiply Operation
	L15	Local Scope Variable Shadowing

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1263,1493,2362

Description

The contract contains statements that checks if an unsigned integer is greater or equal to zero. An unsigned integer is always greater or equal to zero. As a result these expressions are redundant.

```
require(taxFee >= 0 && taxFee <= maxTaxFee, "TF err");
require(liquidityFee >= 0 && liquidityFee <= maxLiqFee, "LF err");
require(burnFee >= 0 && burnFee <= maxBurnFee, "BF err");
require(walletFee >= 0 && walletFee <= maxWalletFee, "WF err");
require(buybackFee >= 0 && buybackFee <= maxBuybackFee, "BBF err");
require(
    walletCharityFee >= 0 && walletCharityFee <= maxWalletFee,
    "WFT err"
);
require(rewardFee >= 0 && rewardFee <= maxTaxFee, "RF err");
//
if (index >= 0) {
```

Recommendation

The contract could remove the redundant expressions.

CR - Code Repetition

Criticality	minor
Location	contract.sol#L1873,1903

Description

The bidirectional logic of sending funds either in native currency or token is repetitive in the source code. Those segments increase the code size of the contract unnecessarily.

```
if (!walletCharityFeeInBNB) {
    uint256 currentBalance = balanceOf(feeWalletCharity);
    _tokenTransferNoFee(
        address(this),
        feeWalletCharity,
        spentAmount
    );
    setBalance(
        payable(feeWalletCharity),
        balanceOf(feeWalletCharity),
        currentBalance
    );
} else {
    uint256 initialBalance = address(this).balance;
    // swap tokens for ETH
    swapTokensForBNB(spentAmount);
    // how much ETH did we just swap into?
    uint256 newBalance = address(this).balance.sub(initialBalance);
    transferEth(feeWalletCharity, newBalance);
}
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L2228,2387,1386,1422,768,2508,1737,1292,2167,1410,1338,1447,1510,1535,2240,1373,1476,777,786,1356,2302,2191,1347,1296,1316,799,1472,1312,1402,1329,791,1406

Description

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

```
totalFees
lock
transfer
isExcludedFromReward
symbol
excludeFromFee
unlock
decimals
updatePcsV2Router
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L1062,1068,1126,1065,1064,1063,1060,1067,1066

Description

Constant state variables should be declared constant to save gas.

```
maxBuybackFee
minMxTxPercentage
dead
maxTaxFee
maxBurnFee
maxWalletFee
mintedByMoonDeploy
minMxWalletPercentage
maxLiqFee
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L2244,1128,1150,1144,1122,1075,1169,1562,2240,1135,1165,1681,1677,1164,1076,1147,1132,2228,2338,1138,1129,847,1535,1141,2232

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_owner
_burnFee
_enabled
WETH
_symbol
_liquidityFee
_account
_taxFee
_walletCharityFee
...
```

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#L272

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#L1480,1514,1518,1526,2508,1561

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.

```
minimumTokenBalanceForDividends = _minimumTokenBalanceForDividends
gasForProcessing = newValue
_maxWalletAmount = _tTotal.mul(maxWalletPercent).div(10 ** 4)
_maxTxAmount = _tTotal.mul(maxTxPercent).div(10 ** 4)
buyBackUpperLimit = buyBackLimit * 10 ** uint256(_decimals)
_taxFee = taxFee
```

Recommendation

Emit an event for critical parameter changes.

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L1201,1480

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)(fee.setRewardFee >= 0 && fee.setRewardFee <= maxTaxFee,RF err)
require(bool,string)(fee.setLiqFee >= 0 && fee.setLiqFee <= maxLiqFee,LF err)
require(bool,string)(liquidityFee >= 0 && liquidityFee <= maxLiqFee,LF err)
require(bool,string)(walletCharityFee >= 0 && walletCharityFee <= maxWalletFee,WFT err)
require(bool,string)(fee.setBuybackFee >= 0 && fee.setBuybackFee <= maxBuybackFee,BBF err)
require(bool,string)(taxFee >= 0 && taxFee <= maxTaxFee,TF err)
require(bool,string)(fee.setWalletCharityFee >= 0 && fee.setWalletCharityFee <=
maxWalletFee,WFT err)
require(bool,string)(walletFee >= 0 && walletFee <= maxWalletFee,WF err)
require(bool,string)(buybackFee >= 0 && buybackFee <= maxBuybackFee,BBF err)
...
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L462,526,318,695,603,652,2257,350,633,614,546,559,670,433,507,494

Description

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

```
functionCall
isContract
safeDecreaseAllowance
_functionCallWithValue
functionCallWithValue
safeTransferFrom
safeApprove
get
_dtransfer
...
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1850

Description

Performing divisions before multiplications may cause lose of prediction.

```
spentAmount = contractTokenBalance.div(totFee).mul(_walletFee)
spentAmount = contractTokenBalance.div(totFee).mul(_burnFee)
spentAmount = contractTokenBalance.div(totFee).mul(_buybackFee)
spentAmount = contractTokenBalance.div(totFee).mul(_rewardFee)
spentAmount = contractTokenBalance.div(totFee).mul(_walletCharityFee)
```

Recommendation

The multiplications should be prior to the divisions.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L2228,2244,2232,2240

Description

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

```
_owner
```

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	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		

SafeMathUint	Library			
	toInt256Safe	Internal		
IterableMapping	Library			
	get	Internal		
	getIndexOfKey	Internal		
	getKeyAtIndex	Internal		
	size	Internal		
	set	Internal	✓	
	remove	Internal	✓	
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	_functionCallWithValue	Private	✓	
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

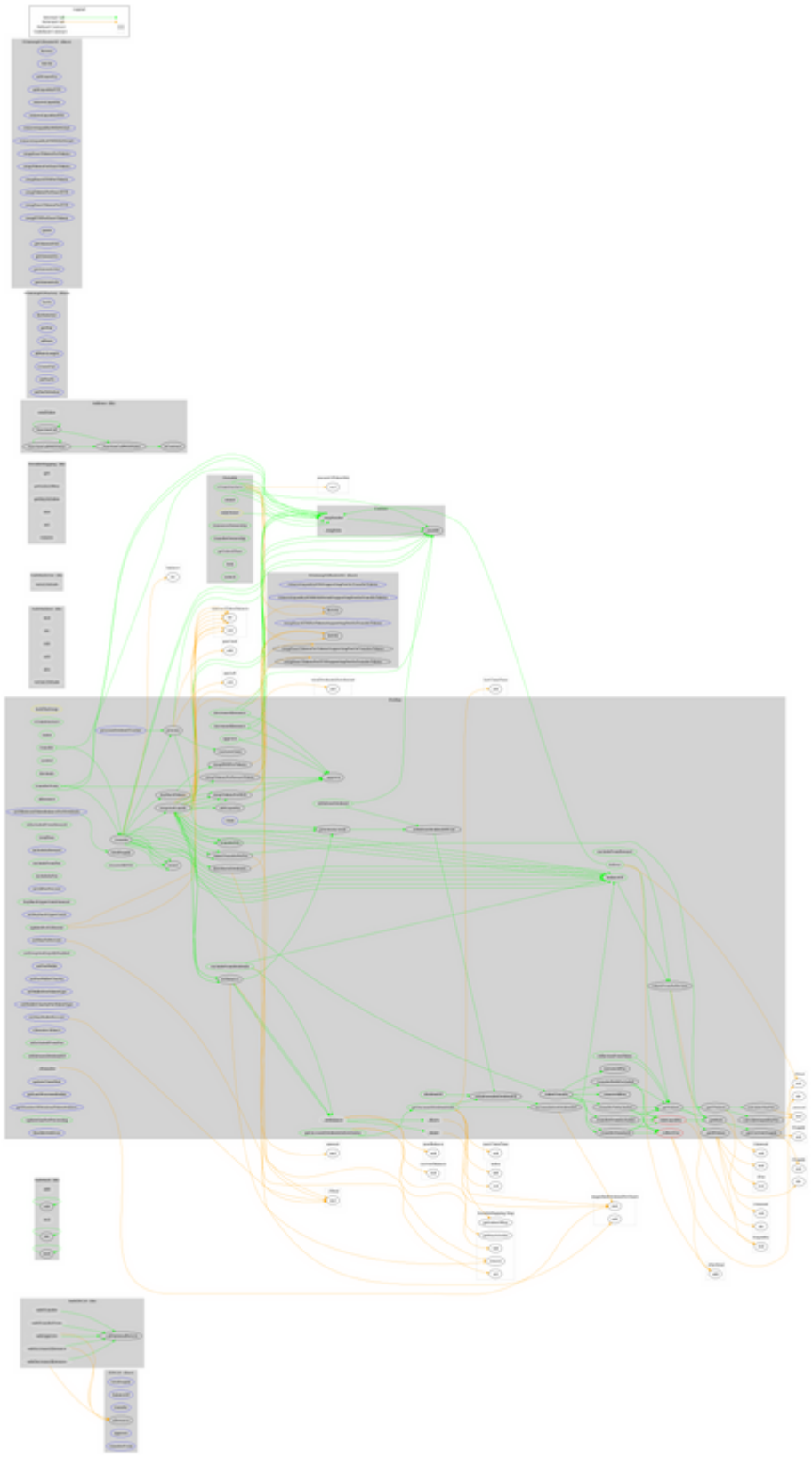
	geUnlockTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Router01	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	✓	-
ProMax	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	Payable	-
	name	Public		-
	updatePcsV2Router	Public	✓	onlyOwner
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner

	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setAllFeePercent	External	✓	onlyOwner
	buyBackUpperLimitAmount	Public		-
	setBuybackUpperLimit	External	✓	onlyOwner
	setMaxTxPercent	External	✓	onlyOwner
	setMaxWalletPercent	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	setFeeWallet	External	✓	onlyOwner
	setFeeWalletCharity	External	✓	onlyOwner
	setWalletFeeTokenType	External	✓	onlyOwner
	setWalletCharityFeeTokenType	External	✓	onlyOwner
	setMinimumTokenBalanceForDividends	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	buyBackTokens	Private	✓	lockTheSwap
	swapTokensForBNB	Private	✓	
	swapBNBForTokens	Private	✓	
	swapTokensForRewardToken	Private	✓	
	addLiquidity	Private	✓	

	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_tokenTransferNoFee	Private	✓	
	transferEth	Private	✓	
	recoverBEP20	Public	✓	onlyOwner
	distributeDividends	Internal	✓	
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_dtransfer	Internal	✓	
	_dmint	Internal	✓	
	_dburn	Internal	✓	
	_setBalance	Internal	✓	
	excludeFromDividends	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	getAccountDividendsInfo	Public		-
	getAccountDividendsInfoAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	Private	✓	
	process	Public	✓	-
	processAccount	Internal	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	processDividendTracker	External	✓	-
	blacklistAddress	External	✓	onlyOwner
	claim	External	✓	-

Contract Flow



Domain Info

Domain Name	islix.digital
Registry Domain ID	324aa301e7b74d9bb2a1b410c71596f2-DONUTS
Creation Date	2022-06-21T19:36:36Z
Updated Date	2022-06-27T18:33:09Z
Registry Expiry Date	2023-06-21T19:36:36Z
Registrar WHOIS Server	http://www.hostinger.com
Registrar URL	http://www.hostinger.com
Registrar	Hostinger, UAB
Registrar IANA ID	1636

The domain has been created in 11 months before the creation of the audit.

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

Summary

There are some functions that can be abused by the owner like manipulating fees and blacklisting addresses. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io