



Cyberscope

Audit Report

\$IR Token

October 2022

Type ERC20

Network ETH

Address 0x483993e969b9A00c4aEDdAE647913530Dc35BA70

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Audit Updates	3
Source Files	4
Contract Analysis	6
ULTW - Transfers Liquidity to Team Wallet	7
Description	7
Recommendation	8
Contract Diagnostics	9
TSD - Total Supply Diversion	10
Description	10
Recommendation	11
STC - Succeeded Transfer Check	12
Description	12
Recommendation	12
BLC - Business Logic Concern	13
Description	13
Recommendation	14
CO - Code Optimization	15
Description	15
Recommendation	15
L04 - Conformance to Solidity Naming Conventions	16
Description	16
Recommendation	16
L13 - Divide before Multiply Operation	17
Description	17

Recommendation	17
L14 - Uninitialized Variables in Local Scope	18
Description	18
Recommendation	18
Contract Functions	19
Contract Flow	27
Domain Info	28
Summary	29
Disclaimer	30
About Cyberscope	31

Contract Review

Contract Name	SirToken
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	300 runs
Explorer	https://etherscan.io/token/0x483993e969b9A00c4aEDdAE647913530Dc35BA70
Symbol	\$IR
Decimals	18
Total Supply	1,000,000,000
Domain	https://sirtoken.com

Audit Updates

Initial Audit	26th October 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol	3e7aa0e0f69eec8f097ad664d525e7b3f0a3fda8dcdd97de5433ddb131db86ef
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol	fa36a21bd954262006d806b988e4495562e7b50420775e2aa0deecb596fd1902
@openzeppelin/contracts/utils/Address.sol	1e0922f6c0bf6b1b8b4d480dcabb691b1359195a297bde6dc5172e79f3a1f826
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/Constants.sol	2fdcb1343e5621385b62e57b5c7775607c272122b6f2dc77da8f84828aa40cd0
@openzeppelin/contracts/utils/math/Safemath.sol	0dc33698a1661b22981abad8e5c6f5ebca0dfe5ec14916369a2935d888ff257a
@openzeppelin/contracts/utils/strings	778d5305652c4eb562b12880cb6cf023d67df24844c15783a0b80fac2e715585

ucts/Enumerable Set.sol	
@uniswap/v2-core/contracts/interfaces/IUniswapV2Factory.sol	51d056199e3f5e41cb1a9f11ce581aa3e190cc982db5771ffef8d8d1f962a0d
@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol	29c75e69ce173ff8b498584700fef76bc81498c1d98120e2877a1439f0c31b5a
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol	0439ffe0fd4a5e1f4e22d71ddbda76d63d61679947d158cba4ee0a1da60cf663
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol	a2900701961cb0b6152fc073856b972564f7c798797a4a044e83d2ab8f0e8d38
sir-token.sol	93d0b55d6d20db31f035b7b269f10163635771c3ccba89205da6a58a665f4847

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Unresolved
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor / informative
Location	contract.sol#L1672,1838,1643
Status	Unresolved

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `withdrawETH`, `recoverETHFromProxy` and `recoverTokensFromProxy` methods.

```
function recoverETHFromProxy() external onlyOwner {
    proxy.sendEth();
}

/// @dev recovers Tokens stuck on proxy
function recoverTokensFromProxy() external onlyOwner {
    moveBalance(_address[0xA3], _address[0xA2]);
}

function withdrawETH(
    address address_
) external onlyOwner {
    uint256 sBalance = address(this).balance.sub(
        Tax[0xA0].ethBalance +
        Tax[0xA1].ethBalance +
        Tax[0xA2].ethBalance
    );
    if (sBalance > 1 ether) {
        payable(address_).transfer(sBalance);
        emit WithdrawnETH(address_, sBalance);
    }
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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 / Informative

Severity	Code	Description	Status
●	STC	Succeeded Transfer Check	Unresolved
●	BLC	Business Logic Concern	Unresolved
●	CO	Code Optimization	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

TSD - Total Supply Diversion

Criticality	minor
Location	contract.sol#L1547
Status	Unresolved

Description

Erase of ETH amount

The accumulated ETH amount counter is erased without handling the actual amount. As a result, the contract wrongly assumes that there is no remaining ETH amount for burn.

Diversion Between total supply and balances

According to the contract's burn mechanism, the burned amount is deducted from the corresponding balance and it is added to the DEAD's address balance. Additionally, the contract deducts the amount from the total supply as well. As a result, the burn operation divers the total supply from the sum of balances.

```
function __burn(
    address address_,
    uint256 amount_
) private {
    unchecked {
        _balances[address_] -= amount_;
    }
    _balances[DEAD] += amount_;
    Total.burned += amount_;
    Total.supply -= amount_;
    Tax[0xA0].ethBalance = 0;
    IUniswapV2Pair(_dexPair).sync();
    Max.txAmount = _getMaxTransactionAmount();
    emit Burned(address_, amount_);
    emit Transfer(address_, DEAD, amount_);
}
```

Recommendation

The contract should handle the ETH amount before erasing it.

Regarding the burn mechanism, the contract should either deduct the amount from the total supply or add it to the DEAD address, but not both of them.

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L1664
Status	Unresolved

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.

```
IERC20(token_).transfer(_owner, amount_);
```

Recommendation

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

BLC - Business Logic Concern

Criticality	medium
Location	contract.sol#L852,1241
Status	Unresolved

Description

The contract might misuse the threshold checker. The function accumulates tokens from the 0xA2 address, but the threshold checks for ETH.

```
function _processBurnFees() private {
    uint256 b = _balances[_address[0xA0]];
    moveBalance(_address[0xA0], address(this));
    bool t;
    if (Tax[0xA2].enabled && isOverThresholdE(0xA2)) {
        b += _balances[_address[0xA2]];
    }
}
```

The contract is using the balance of the 0xA0 amount, but is erasing the 0xA2 amount.

```
function _buyBackAndBurn() private lockTheSwap {
    uint256 eAmount = Tax[0xA0].ethBalance; //Burn
    Tax[0xA2].ethBalance = 0; //TREASURY
}
```

The contract set the new address in the index and then erase the previous locker. The setLocker does not allow to:

1. Change an address that is not owned by the 0xA6, 0xA8, 0xA9, 0xAB indexes.
2. The timestamp is not allowed to be zero.

```
_address[index_] = address_;
```

```
if (isAccountLocked(previousAddress)) {  
    setLock(_address[index_], _locked[previousAddress]);  
    setLock(previousAddress, 0);  
}
```

Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

The contract could use the corresponding threshold checker.

The contract should probably erase the BURN address.

The setAddress() approach regarding the functionality of the lockers should be reconsidered.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L1311
Status	Unresolved

Description

The contract is using a state variable (txAmount) to store the maximum transaction amount threshold. This variable depends on the total supply and a contrant number. The contract has to keep this variable updated when the total supply is getting changed.

```
function _getMaxTransactionAmount() private view returns(uint256) {  
    return Total.supply.mul(Max.txPercent).div(1e4);  
}
```

```
Max.txAmount = _getMaxTransactionAmount();
```

The contract is ussing an address key mapping mechanism that is making the code hard to read.

BURN	0xA0		LIQUIDITY	0xA1		TREASURY	0xA2		PROXY	0xA3
DEV	0xA4		ADVISORS	0xA5		TEAM	0xA6		MARKETING	0xA7
RESERVE	0xA8		SEED	0xA9		AIRDROPS	0xAA		LP PROV.	0xAB
PAIR	0xB3		FACTORY	0xB4		ROUTER	0xB2		WETH	0xB1
LP ADDR.	0xB0									

Recommendation

The contract could remove the txAmount as a property and use the _getMaxTransactionAmount() every time it is required to use the txAmount value.

Keys could be encoded to be more readable. For instance, `0xA0` could be `bytes32 public constant Burn = keccak256(abi.encodePacked("BURN"))` ;`. The extra size overhead is insignificant.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	sir-token.sol#L276,319,273,1545,275,271,272,262,303,261,291,331,1490,1509,260
Status	Unresolved

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.

```
_initialSupply  
Total  
_symbol  
__burn  
_multiplier  
_decimals  
_name  
Airdrops  
Tax  
...
```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	sir-token.sol#L850
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
p = e.div(Tax[0xA0].percent + Tax[0xA2].percent)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	sir-token.sol#L751,1400,853,985
Status	Unresolved

Description

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

```
_fee  
diff  
t  
previousTime
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	safePermit	Internal	✓	

	_callOptionalReturn	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		
Context	Implementation			
	_msgSender	Internal		
	_msgData	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		
EnumerableSet	Library			
	_add	Private	✓	
	_remove	Private	✓	
	_contains	Private		
	_length	Private		
	_at	Private		
	_values	Private		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	values	Internal		
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	✓	-

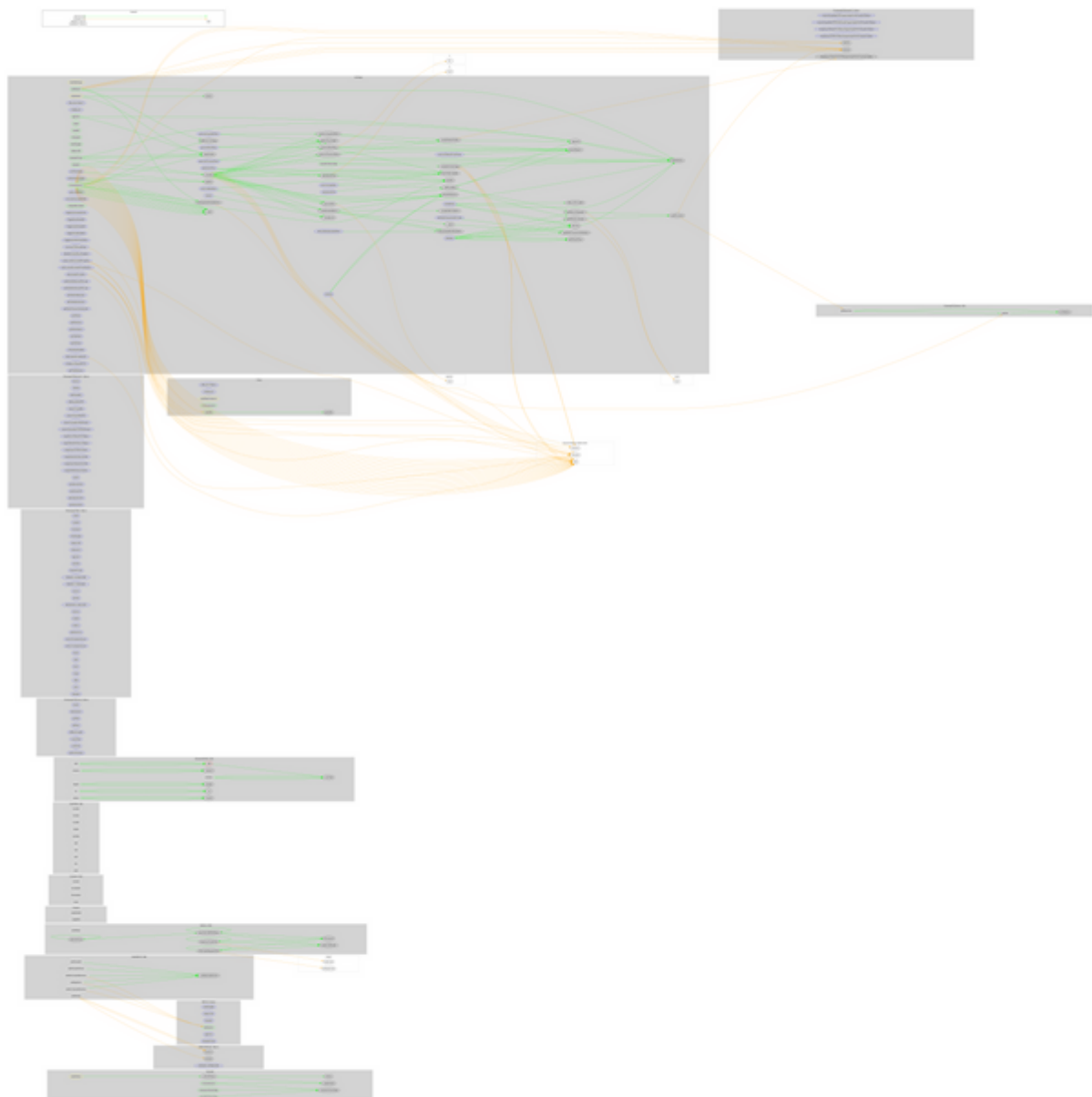
	initialize	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	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	✓	-

UniswapV2Library	Library			
	sortTokens	Internal		
	pairFor	Internal		
	getReserves	Internal		
Proxy	Implementation			
	<Receive Ether>	External	Payable	-
	<Fallback>	External	Payable	-
	<Constructor>	Public	Payable	-
	sendEth	Public	✓	onlyMainContract
SirToken	Implementation	IERC20, Context		
	<Constructor>	Public	Payable	-
	<Receive Ether>	External	Payable	-
	<Fallback>	External	Payable	-
	setRouter	Public	✓	onlyOwner
	initialize	External	✓	onlyOwner
	setDefaultVars	Private	✓	
	owner	Public		-
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	_approve	Private	✓	
	_transfer	Private	✓	
	_processFee	Private	✓	

	processBurnFees	External	✓	onlyOwner
	processLiquidityFees	External	✓	onlyOwner
	processTreasuryFees	External	✓	onlyOwner
	processBuyBack	External	✓	onlyOwner
	processLiquidity	External	✓	onlyOwner
	distributeFees	External	✓	onlyOwner
	_processBurnFees	Private	✓	
	_processLiquidityFees	Private	✓	
	_processTreasuryFees	Private	✓	
	_distributeFees	Private	✓	
	isOverThresholdE	Private		
	isOverThresholdT	Private		
	setAccountExcludedFromFee	External	✓	onlyOwner
	setAccountExcludedFromMaxTx	External	✓	onlyOwner
	setLock	Public	✓	onlyOwner
	getLockTimes	External		-
	isAccountLocked	Public		-
	accountIsLockedUntil	External		-
	isLiquidityLocked	Public		-
	liquidityIsLockedUntil	External		-
	toggleFeesEnabled	External	✓	onlyOwner
	toggleLimitsEnabled	External	✓	onlyOwner
	toggleSwapEnabled	External	✓	onlyOwner
	toggleAutoDistributeFees	External	✓	onlyOwner
	setTaxPercentage	External	✓	onlyOwner
	setAutoBuyBackAndBurn	External	✓	onlyOwner
	isBuyBackAndBurnEnabled	External		-
	setThreshold	External	✓	onlyOwner
	isExcludedFromFee	External		-
	isExcludedFromMaxTx	External		-
	setAddress	External	✓	onlyOwner
	setMaxSellAmountPercent	External	✓	onlyOwner
	setMaxBuyAmountPercent	External	✓	onlyOwner
	getMaxSellAmount	External		-

	_getMaxSellAmount	Private		
	getMaxBuyAmount	External		-
	_getMaxBuyAmount	Private		
	getMaxTransactionAmount	External		-
	_getMaxTransactionAmount	Private		
	getTotals	External		-
	getBooleans	External		-
	getMaxValues	External		-
	getTaxData	External		-
	setMaxTransactionPercent	External	✓	onlyOwner
	airdrop	External	✓	onlyOwner
	_buyBackAndBurn	Private	✓	lockTheSwap
	_swapTokensForEth	Private	✓	lockTheSwap
	_swapEthForTokens	Private	✓	lockTheSwap
	getAddress	External		-
	cleanUpAndEndTheBurn	Private	✓	
	isTradingEnabled	External		-
	_getReserves	Public		-
	getLiquidityPoolAddress	Public		-
	__mint	Private	✓	
	__burn	Private	✓	
	_addLiquidity	Private	✓	lockTheSwap
	moveBalance	Private	✓	
	transferBalance	Private	✓	
	recoverETHFromProxy	External	✓	onlyOwner
	recoverTokensFromProxy	External	✓	onlyOwner
	withdrawTokenERC20	External	✓	onlyOwner
	withdrawETH	External	✓	onlyOwner
	applies	Private		
	_checkLock	Private	✓	
	unlock	External	✓	onlyOwner
	getTaxBalances	External		-
	isRouterUniswap	Private		
	validAddress	Private		

Contract Flow



Domain Info

Domain Name	sirtoken.com
Registry Domain ID	2617638582_DOMAIN_COM-VRSN
Creation Date	2021-06-06T15:37:35.00Z
Updated Date	2022-06-07T10:17:04.56Z
Registry Expiry Date	2023-06-06T15:37:35.00Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	http://www.namecheap.com
Registrar	NAMECHEAP INC
Registrar IANA ID	1068

The domain was created over 1 year before the creation of the audit. It will expire in 8 months.

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

Summary

The Smart Contract analysis reported one minor severity issue. The contract owner has the authority to transfer funds to the team's wallet. Other than that, 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 limit of max 14% for buy and sell transactions. Additionally, there is a max of 8% transfer tax.

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>