Cyberscope

Audit Report

IllumiShare SRG

November 2022

SHA256 af78b68d540627869bbd387681101ef560102980571f62007658e6f89d9e6b95

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	SrgToken
Compiler Version	v0.8.17+commit.8df45f5f
Testing Deploy	https://testnet.bscscan.com/token/0xe425e73C115cE82603fDb9B3aBb769e6B7404144
Symbol	SRG
Decimals	18

Audit Updates

Initial Audit	24th June 2022 https://github.com/cyberscope-io/audits/blob/main/illumi/v1/audit.pdf
Corrected Phase 1	11th October 2022 https://github.com/cyberscope-io/audits/blob/main/illumi/v2.pdf
Corrected Phase 2	2nd November 2022

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/token/ERC20/ERC20.sol	5031430cc2613c32736d598037d3075985a2a09e61592a013dbd09a5bc2041b8
@openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol	3e7aa0e0f69eec8f097ad664d525e7b3f0a3fda8dcdd97de5433ddb131db86ef
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@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/math/SafeMath.sol	0dc33698a1661b22981abad8e5c6f5ebca0dfe5ec14916369a2935d888ff257a
contracts/SrgToken-1.sol	af78b68d540627869bbd387681101ef560102980571f62007658e6f89d9e6b95

Contract Analysis

Critical Medium Minor Pass

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

ST - Stops Transactions

Criticality	medium
Location	SrgToken-1.sol#L179
Status	Unresolved

Description

The contract owner has the authority to stop the transactions. The owner may take advantage by setting the coldStakingAddress to null. As a result, the contract will revert. Additionally, the balance of coldStakingAddress might manipulate the transaction flow and stop transactions. This could happen by setting the balance of the sender to the maximum possible amount.

```
function _beforeTokenTransfer(
        address from,
        address,
        uint256 amount
    ) internal virtual override {
        if (from != address(0)) {
            require(
                balanceOf(from) >
                    IColdStaking(coldStakingAddress).balanceOf(from) + amount,
                "Not enough unlocked"
            );
        }
    }
```

Recommendation

The contract could embody a check for not allowing setting the coldStakingAddress to an invalid address. Additionally, the contract could embody the external call on a try-catch statement and sanitize the returned value.

Instead of using Untrusted Source to manipulate the transaction flow. It is recommended to add a max transaction amount to a fixed percentage of the total supply. For example, an acceptable max transaction amount is greater than 0.1% 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.

OCTD - Transfers Contract's Tokens

Criticality	minor / informative
Location	SrgToken-1.sol#L106
Status	Unresolved

Description

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

```
function withdrawTokens(IERC20 token) external onlyOwner {
        token.safeTransfer(owner(), token.balanceOf(address(this)));
}
```

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.

MT - Mints Tokens

Criticality	critical
Location	SrgToken-1.sol#L155
Status	Unresolved

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling `the deposit` function. As a result, the contract tokens will be highly inflated.

```
function deposit(address user, bytes calldata depositData) external {
        require(_msgSender() == depositAdmin, "sender != depositAdmin");
        uint256 amount = abi.decode(depositData, (uint256));
        _mint(user, amount);
}
```

Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

BT - Burns Tokens

Criticality	critical
Location	SrgToken-1.sol#L49
Status	Unresolved

Description

The contract owner has the authority to burn tokens from a specific address. The owner may take advantage of it by calling the `burn` function. As a result, the targeted contract address will lose the corresponding tokens.

```
function burn(address tokenHolder, uint256 amount)
        external
        onlyOwner
        returns (bool)
    {
        _burn(tokenHolder, amount);
        return (true);
    }
```

Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description	Status
●	US	Untrusted Source	Unresolved

US - Untrusted Source

Criticality	critical
Location	SrgToken-1.sol#L25
Status	Unresolved

Description

The contract uses an external contract in order to determine the transaction's flow. The external contract is untrusted. As a result, it may produce security issues and harm the transactions.

```
address public coldStakingAddress;
```

Recommendation

The contract should use a trusted external source. A trusted source could be either a commonly recognized or an audited contract. The pointing addresses should not be able to change after the initialization.

Contract Functions

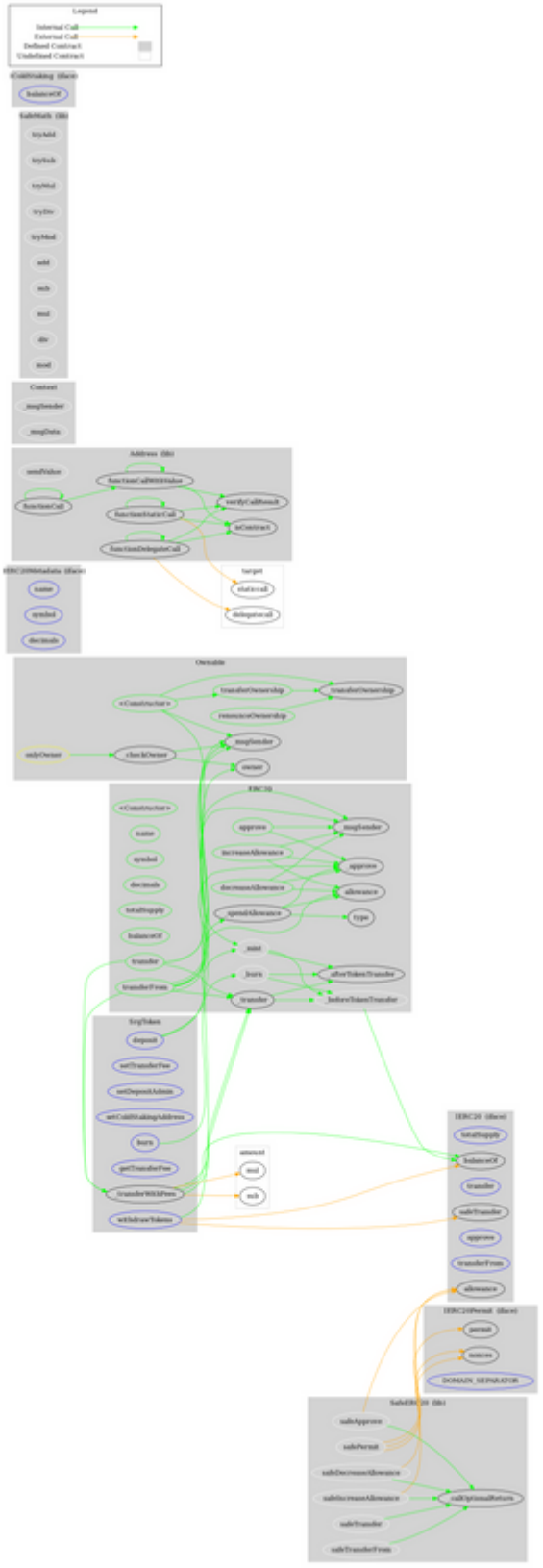
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_spendAllowance	Internal	✓	

	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
IERC20Permit	Interface			
	permit	External	✓	-
	nonces	External		-
	DOMAIN_SEPARATOR	External		-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	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		
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		
IColdStaking	Interface			
	balanceOf	External		-
SrgToken	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	burn	External	✓	onlyOwner

	setTransferFee	External	✓	onlyOwner
	setDepositAdmin	External	✓	onlyOwner
	setColdStakingAddress	External	✓	onlyOwner
	withdrawTokens	External	✓	onlyOwner
	getTransferFee	External		-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	deposit	External	✓	-
	_transferWithFees	Private	✓	
	_beforeTokenTransfer	Internal	✓	

Contract Flow



Summary

There are some functions that can be abused by the owner like minting tokens, and burning tokens. if the contract owner abuses the mint functionality, then the contract will be highly inflated. if the contract owner abuses the burning functionality, then the users could lose their tokens. 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