Cyberscope

Audit Report

PoshCoin

September 2022

Type	BEP20
Network	BSC
Address	0x9219431d8dd5ad62FD1E95b62ba38eD50781d772
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	PoshCoin
Compiler Version	v0.8.16+commit.07a7930e
Optimization	200 runs
Explorer	https://bscscan.com/token/0x9219431d8dd5ad62FD1E95b62ba38eD50781d772
Symbol	PSCN
Decimals	18
Total Supply	1,000,000,000
Domain	https://poshcoin.io

Audit Updates

Initial Audit	9th September 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/AccessControl.sol	0b280a0fe505b5b8bcb700e0b1f6242acf73e0b509372ef3acc46db051512e32
@openzeppelin/contracts/access/IAccessControl.sol	d03c1257f2094da6c86efa7aa09c1c07ebd33dd31046480c5097bc2542140e45
@openzeppelin/contracts/access/Ownable.sol	75e3c97011e75627ffb36f4a2799a4e887e1a3e27ed427490e82d7b6f51cc5c9
@openzeppelin/contracts/security/Pausable.sol	5b6abc290190f46b9941c674594eee083a3fe6b92d1828d0cfefacc94d1cac9a
@openzeppelin/contracts/token/ERC20/ERC20.sol	f7831910f2ed6d32acff6431e5998baf50e4a00121303b27e974aab0ec637d79
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	c2b06bb4572bb4f84bfc5477dadc0fcc497cb66c3a1bd53480e68bedc2e154a6
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a

@openzeppelin/contracts/utils/introspection/ERC165.sol	8806a632d7b656cadb8133ff8f2acae4405b3a64d8709d93b0fa6a216a8a6154
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5
@openzeppelin/contracts/utils/math/SafeMath.sol	15941f3904992a62ed117e93d9e2d5c4c22bd09a7ff97fdd5f49273cf09703ac
@openzeppelin/contracts/utils/Strings.sol	8597c62818dcbc6cf85c21179b90b714fb4f70a4347ca2eed23e88c87b08b8a1
contracts/PoshCoin.sol	7e7fbf09f577c97c8effd7fbe40cf80cdd6fbdde44c8d15e6a7b174cf6900714

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

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

ST - Stops Transactions

Criticality	minor / informative
Location	contract.sol#L73
Status	Unresolved

Description

The contract owner has the authority to stop the transactions for all users excluding the owner. The owner may take advantage of it by setting the `maxAmountHold` to zero and by calling `pause` method.

```
function _transfer(
        address from,
        address to,
        uint256 amount
    ) internal override whenNotPaused {

        require(from != address(0), "ERC20: transfer from the zero address");
        require(to != address(0), "ERC20: transfer to the zero address");
        require(
            !_isBlacklisted[from] && !_isBlacklisted[to],
            "Blacklisted address"
        );
        uint256 newBalance = balanceOf(to) + amount;

        require(
            newBalance <= maxAmountHold,
            "Your balance is exceeded the limit to hold."
        );

        super._transfer(from, to, amount);
    }
```

Recommendation

The contract could embody a check for not allowing setting the maxAmountHold 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.

MT - Mints Tokens

Criticality	critical
Location	contract.sol#L95
Status	Unresolved

Description

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

```
function mint(address to, uint256 amount) public whenNotPaused onlyOwner {
        _mint(to, amount);
        totalMintedToken = totalMintedToken + 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	contract.sol#L100
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 from, uint256 amount) public whenNotPaused {
        // Check that the calling account has the Burner role
        require(hasRole(BURNER_ROLE, msg.sender), "Caller is not a Burner");
        _burn(from, amount);
        totalBurnedToken = totalBurnedToken + 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.

BC - Blacklists Addresses

Criticality	medium
Location	contract.sol#L44
Status	Unresolved

Description

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

```
function blacklistAddress(address account, bool value) external onlyOwner {
        require(account != owner(), "Shouldn't be owner address");
        _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 / Informative

Severity	Code	Description	Status
	L01	Public Function could be Declared External	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L07	Missing Events Arithmetic	Unresolved
	L15	Local Scope Variable Shadowing	Unresolved

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contracts/PoshCoin.sol#L57,61,65,111,95,100,107
Status	Unresolved

Description

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

```
decimals
grandBurnerRole
revokeBurnerAccess
unpause
mint
burn
pause
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/PoshCoin.sol#L29
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.

```
_isBlacklisted
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contracts/PoshCoin.sol#L53
Status	Unresolved

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.

```
maxAmountHold = newAmount
```

Recommendation

Emit an event for critical parameter changes.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contracts/PoshCoin.sol#L33,34,32
Status	Unresolved

Description

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

```
symbol
totalSupply
name
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

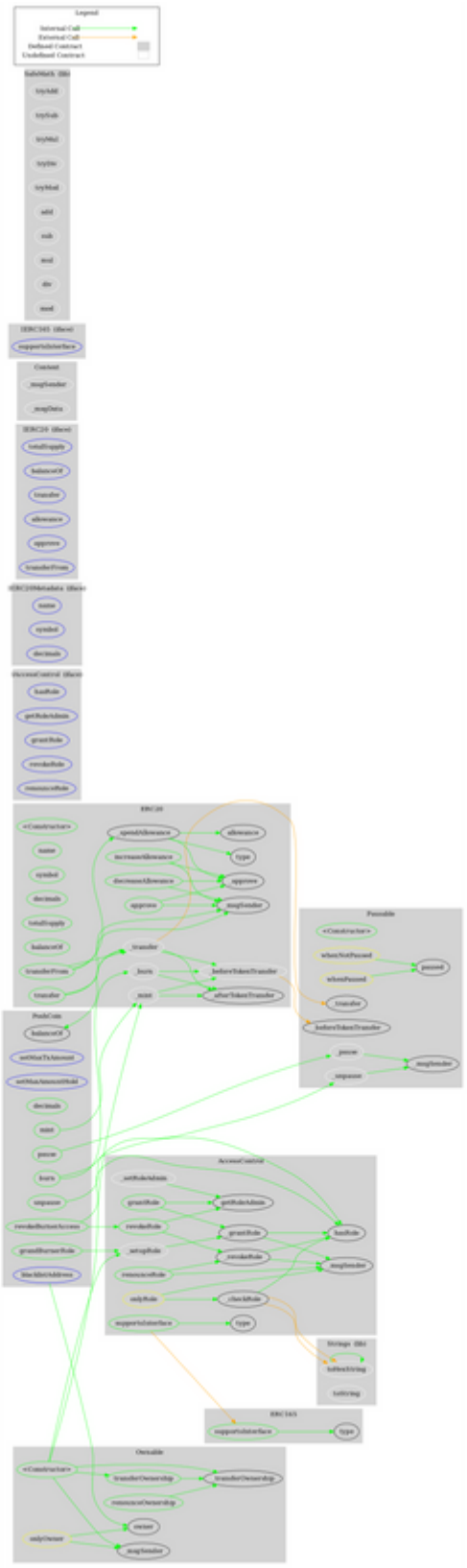
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	

Pausable	Implementation	Context		
	<Constructor>	Public	✓	-
	paused	Public		-
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
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	✓	
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-

	decimals	External		-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC165	Interface			
	supportsInterface	External		-
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		

Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
PoshCoin	Implementation	ERC20, Ownable, AccessControl, Pausable		
	<Constructor>	Public	✓	ERC20
	blacklistAddress	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	setMaxAmountHold	External	✓	onlyOwner
	decimals	Public		-
	grandBurnerRole	Public	✓	onlyOwner
	revokeBurnerAccess	Public	✓	onlyOwner
	_transfer	Internal	✓	whenNotPaused
	mint	Public	✓	whenNotPaused onlyOwner
	burn	Public	✓	whenNotPaused
	pause	Public	✓	onlyOwner
	unpause	Public	✓	onlyOwner
	_beforeTokenTransfer	Internal	✓	whenNotPaused

Contract Flow



Domain Info

Domain Name	poshcoin.io
Registry Domain ID	03ffff9fe6c54b03b1a0944fb8ac5f59-DONUTS
Creation Date	2022-08-04T15:47:01Z
Updated Date	2022-08-09T15:47:14Z
Registry Expiry Date	2023-08-04T15:47:01Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain was created about 1 month 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

There are some functions that can be abused by the owner like stopping transactions, minting tokens, burning tokens and blacklisting addresses. if the contract owner abuses the mint functionality, then the contract will be highly inflated. if the contract owner abuses the burn functionality, then the users could lost 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