Cyberscope

Audit Report

Karpine Supply Chain xCellence

March 2024

Network	BSC
Address	0x5706684bc4a6311b81c77239e0090f962dc811c1
Audited by	© cyberscope

Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Unresolved
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	RSW	Redundant Storage Writes	Unresolved
●	OCTD	Transfers Contract's Tokens	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved

Table of Contents

Review

Contract Name	KSCxToken
Compiler Version	v0.8.20+commit.a1b79de6
Optimization	200 runs
Explorer	https://bscscan.com/address/0x5706684bc4a6311b81c77239e0090f962dc811c1
Address	0x5706684bc4a6311b81c77239e0090f962dc811c1
Network	BSC
Symbol	KSCx
Decimals	18
Total Supply	100,000,000,000
Badge Eligibility	Yes

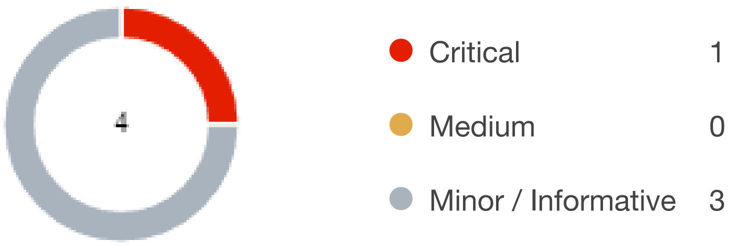
Audit Updates

Initial Audit	08 Mar 2024 https://github.com/cyberscope-io/audits/blob/main/kscx/v1/audit.pdf
Corrected Phase 2	11 Mar 2024

Source Files

Filename	SHA256
KSCxTok.sol	29b606299a72ef11e865dfd57a5a5a00c8455b4de5d59fbbce4dbffb9b86f31a

Findings Breakdown



Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	1	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	3	0	0	0

ST - Stops Transactions

Criticality	Critical
Location	KSCxTok.sol#L132
Status	Unresolved

Description

The contract owner has the authority to stop the sales for all users . The owner may take advantage of it by calling the pause function. As a result, the contract may operate as a honeypot.

```
function pause() external onlyOwner {
    _pause();
}

function _beforeTokenTransfer(address sender, address
recipient, uint256 amount) internal
override(ERC20,ERC20Pausable) whenNotPaused {
    super._beforeTokenTransfer(sender, recipient, amount);
}
```

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.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

RSW - Redundant Storage Writes

Criticality	Minor / Informative
Location	KSCxTok.sol#L64,106,113
Status	Unresolved

Description

The contract modifies the state of the following variables without checking if their current value is the same as the one given as an argument. As a result, the contract performs redundant storage writes, when the provided parameter matches the current state of the variables, leading to unnecessary gas consumption and inefficiencies in contract execution.

```
function toggleBurn(bool _enabled) external onlyOwner {
    burnForPublicEnabled = _enabled;
    emit BurnForPublicEnabledUpdated(_enabled);
}

function setTransactionFee(uint256 _transactionFee) external
onlyOwner {
    require(_transactionFee <= 5, "KSCxToken: Transaction fee
cannot exceed 5%");
    emit TransactionFeeUpdated(transactionFee,
_transactionFee);
    transactionFee = _transactionFee;
}

function setBurnRate(uint256 _burnRate) external  onlyOwner {
    require(_burnRate <= 5, "KSCxToken: Burn rate cannot exceed
5%");
    emit BurnRateUpdated(burnRate, _burnRate); // Emit an event
with the old and new burn rates
    burnRate = _burnRate;
}
```

Recommendation

The team is advised to implement additional checks within to prevent redundant storage writes when the provided argument matches the current state of the variables. By incorporating statements to compare the new values with the existing values before proceeding with any state modification, the contract can avoid unnecessary storage operations, thereby optimizing gas usage.

OCTD - Transfers Contract's Tokens

Criticality	Minor / Informative
Location	KSCxTok.sol#L100
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 `recoverERC20` function.

```
function recoverERC20(address tokenAddress, uint256
tokenAmount) external onlyOwner {
    IERC20(tokenAddress).safeTransfer(owner(), tokenAmount);
    emit TokensRecovered(tokenAddress, tokenAmount);
}
```

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.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	KSCxTok.sol#L64,82,91,106,113,120
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
bool _enabled
uint256 _maxTransferAmountRate
uint256 _maxWalletBalanceRate
uint256 _transactionFee
uint256 _burnRate
address _feeDestination
```

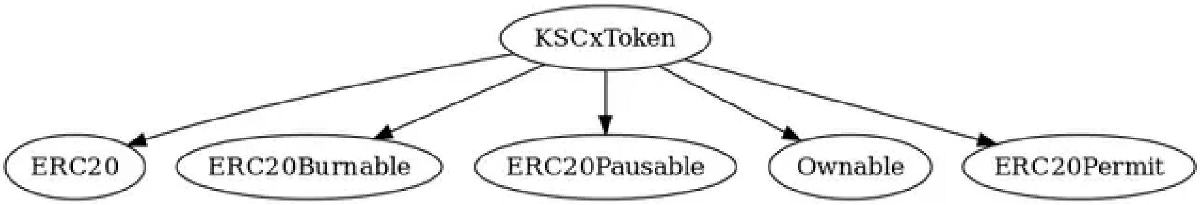
Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation
https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

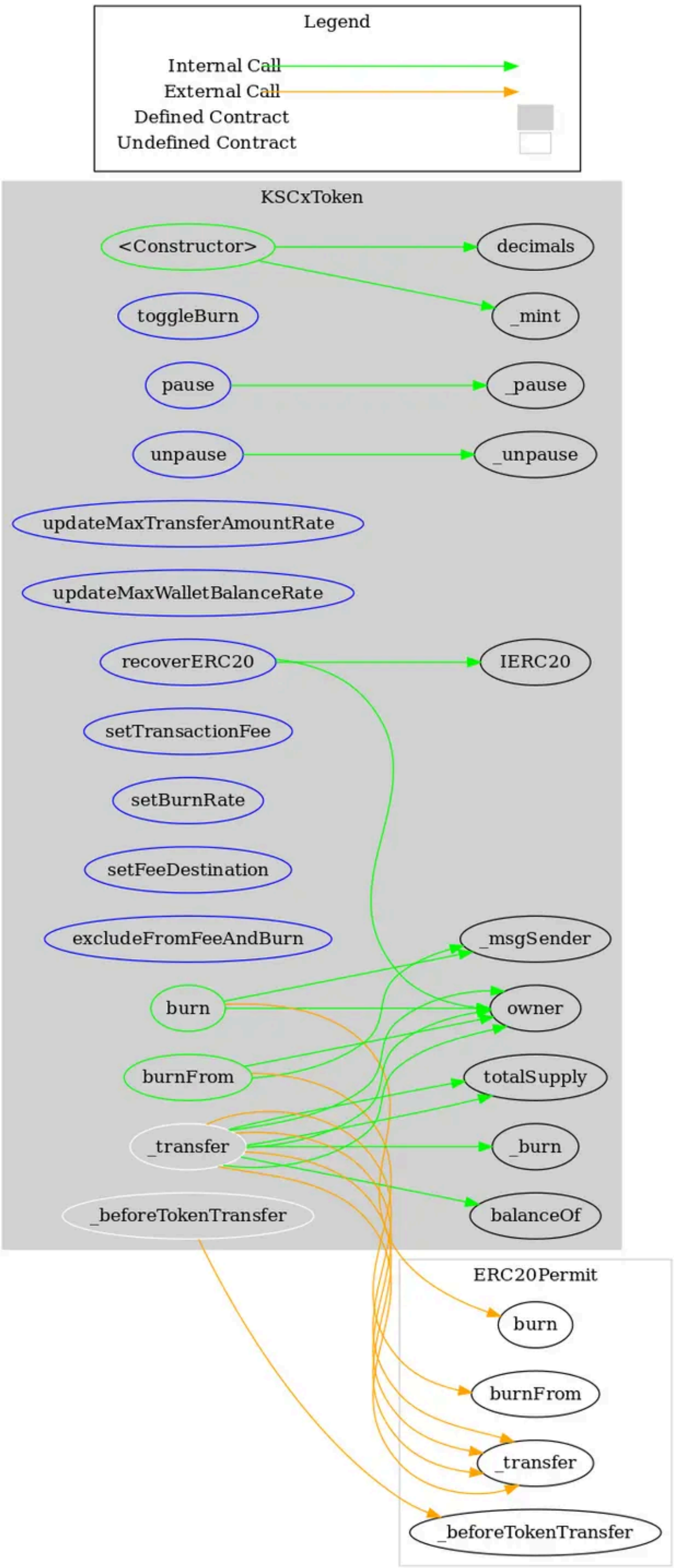
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
KSCxToken	Implementation	ERC20, ERC20Burnable, ERC20Pausable, Ownable, ERC20Permit		
		Public	✓	ERC20 Ownable ERC20Permit
	toggleBurn	External	✓	onlyOwner
	burn	Public	✓	-
	burnFrom	Public	✓	-
	updateMaxTransferAmountRate	External	✓	onlyOwner
	updateMaxWalletBalanceRate	External	✓	onlyOwner
	recoverERC20	External	✓	onlyOwner
	setTransactionFee	External	✓	onlyOwner
	setBurnRate	External	✓	onlyOwner
	setFeeDestination	External	✓	onlyOwner
	excludeFromFeeAndBurn	External	✓	onlyOwner
	pause	External	✓	onlyOwner
	unpause	External	✓	onlyOwner
	_beforeTokenTransfer	Internal	✓	whenNotPaused
	_transfer	Internal	✓	

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
KSCxToken
<Constructor>
decimals
toggleBurn
_mint
pause
_pause
unpause
_unpause
updateMaxTransferAmountRate
updateMaxWalletBalanceRate
recoverERC20
IERC20
setTransactionFee
setBurnRate
setFeeDestination
excludeFromFeeAndBurn
_msgSender
burn
owner
burnFrom
totalSupply
_transfer
_burn
_beforeTokenTransfer
balanceOf
ERC20Permit
burn
burnFrom
_transfer
_beforeTokenTransfer

Summary

Karpine Supply Chain xCellence contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. There are some functions that can be abused by the owner like stop 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

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io