Cyberscope

Audit Report

Seismic

March 2023

SHA256 659166545dff63d5199913413d6ab718ed71238c11dd297368b1fd49573107a2

Audited by © cyberscope

Table of Contents

Review

Contract Name	SCBB
Testing Deploy	https://testnet.snowtrace.io/address/0x39d4190311C8434d208e8a88b4ce1580D34C834b
Symbol	SCB
Decimals	18
Total Supply	10,000,000

Audit Updates

Initial Audit	16 Mar 2023 https://github.com/cyberscope-io/audits/tree/main/seismic/v1/audit.pdf
Corrected Phase 2	20 Mar 2023

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/token/ERC20/ERC20.sol	5031430cc2613c32736d598037d3075985a2a09e61592a013dbd09a5bc2041b8
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/math/SafeMath.sol	0dc33698a1661b22981abad8e5c6f5ebca0dfe5ec14916369a2935d888ff257a
contracts/SCB.sol	659166545dff63d5199913413d6ab718ed71238c11dd297368b1fd49573107a2

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	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	PVC	Price Volatility Concern	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

PVC - Price Volatility Concern

Criticality	Minor / Informative
Location	contracts/SCB.sol#L242
Status	Unresolved

Description

The contract accumulates tokens from the taxes to swap them for ETH. The variable `swapTokensAtAmount` sets a threshold where the contract will trigger the swap functionality. If the variable is set to a big number, then the contract will swap a huge amount of tokens for ETH.

It is important to note that the price of the token representing it, can be highly volatile. This means that the value of a price volatility swap involving Ether could fluctuate significantly at the triggered point, potentially leading to significant price volatility for the parties involved.

```
function setSwapTokensAtAmount(uint256 _newAmount) external onlyOwner {
    require(
        _newAmount > 0,
        "SCB : Minimum swap amount must be greater than 0!"
    );
    swapTokensAtAmount = _newAmount;
}
```

Recommendation

The contract could ensure that it will not sell more than a reasonable amount of tokens in a single transaction. A suggested implementation could check that the maximum amount should be less than a fixed percentage of the total supply. Hence, the contract will guarantee that it cannot accumulate a huge amount of tokens in order to sell them.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	SCBB.sol#L38,67,102,138,147,156,165,174,184,185,198,199,211,219,229,230,242,262,263,269,402,413,427,446
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.

```
function WETH() external pure returns (address);
uint256 private constant _totalSupply = 1e7 * 1e18
address public TreasuryWallet = 0x74Adf47aD22a9C95EE58A6D956FA58924D697E0F
address _newTreasury
uint256 _mb
uint256 _ms
uint256 _mt
uint256 _mx
uint256 _lpTax
uint256 _TreasuryTax
uint256 _sc
uint256 _db
uint256 _newAmount
address _wallet

...
```

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.

L07 - Missing Events Arithmetic

Criticality	Minor / Informative
Location	SCBB.sol#L216,225
Status	Unresolved

Description

Events are a way to record and log information about changes or actions that occur within a contract. They are often used to notify external parties or clients about events that have occurred within the contract, such as the transfer of tokens or the completion of a task.

It's important to carefully design and implement the events in a contract, and to ensure that all required events are included. It's also a good idea to test the contract to ensure that all events are being properly triggered and logged.

```
sellCooldown = _sc
deadBlocks = _db
```

Recommendation

By including all required events in the contract and thoroughly testing the contract's functionality, the contract ensures that it performs as intended and does not have any missing events that could cause issues with its arithmetic.

L14 - Uninitialized Variables in Local Scope

Criticality	Minor / Informative
Location	SCBB.sol#L331
Status	Unresolved

Description

Using an uninitialized local variable can lead to unpredictable behavior and potentially cause errors in the contract. It's important to always initialize local variables with appropriate values before using them.

```
uint256 tax
```

Recommendation

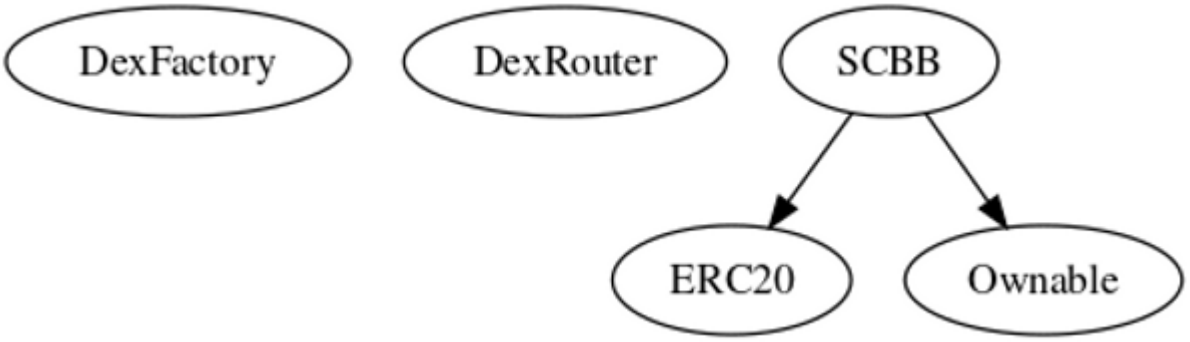
By initializing local variables before using them, the contract ensures that the functions behave as expected and avoid potential issues.

Functions Analysis

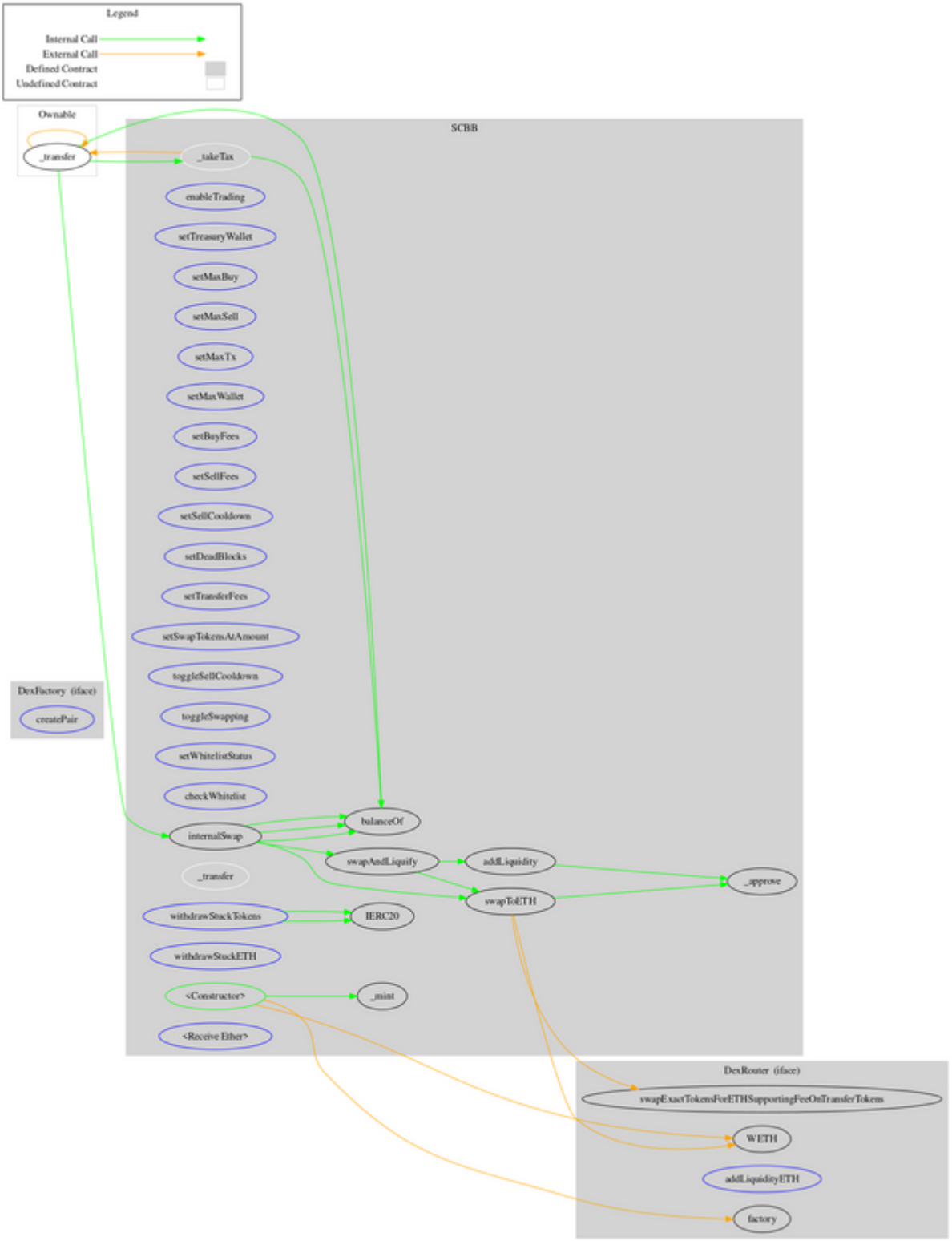
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
DexFactory	Interface			
	createPair	External	✓	-
DexRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupporting FeeOnTransferTokens	External	✓	-
SCBB	Implementation	ERC20, Ownable		
		Public	✓	ERC20
	enableTrading	External	✓	onlyOwner
	setTreasuryWallet	External	✓	onlyOwner
	setMaxBuy	External	✓	onlyOwner
	setMaxSell	External	✓	onlyOwner
	setMaxTx	External	✓	onlyOwner
	setMaxWallet	External	✓	onlyOwner
	setBuyFees	External	✓	onlyOwner
	setSellFees	External	✓	onlyOwner
	setSellCooldown	External	✓	onlyOwner
	setDeadBlocks	External	✓	onlyOwner
	setTransferFees	External	✓	onlyOwner
	setSwapTokensAtAmount	External	✓	onlyOwner

	toggleSellCooldown	External	✓	onlyOwner
	toggleSwapping	External	✓	onlyOwner
	setWhitelistStatus	External	✓	onlyOwner
	checkWhitelist	External		-
	_takeTax	Internal	✓	
	_transfer	Internal	✓	
	internalSwap	Internal	✓	
	swapAndLiquify	Internal	✓	
	swapToETH	Internal	✓	
	addLiquidity	Private	✓	
	withdrawStuckETH	External	✓	onlyOwner
	withdrawStuckTokens	External	✓	onlyOwner
		External	Payable	-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Ownable
_transfer
SCBB
_takeTax
enableTrading
setTreasuryWallet
setMaxBuy
setMaxSell
setMaxTx
setMaxWallet
setBuyFees
setSellFees
setSellCooldown
setDeadBlocks
setTransferFees
setSwapTokensAtAmount
toggleSellCooldown
toggleSwapping
setWhitelistStatus
checkWhitelist
internalSwap
_transfer
withdrawStuckTokens
withdrawStuckETH
<Constructor>
<Receive Ether>
balanceOf
swapAndLiquify
addLiquidity
swapToETH
_approve
IERC20
_mint
DexFactory (iface)
createPair
DexRouter (iface)
swapExactTokensForETHSupportingFeeOnTransferTokens
WETH
addLiquidityETH
factory

Summary

Seismic contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. Seismic is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors or critical issues. 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 22% buy/sell fees, and 11% transfer fees.

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