Cyberscope

# Audit Report

# Stelnar

April 2024

| | |
|---|---|
| Network | BSC Testnet |
| Address | 0xcd224393ce8ccd7b034dbe9619ddfd11ade121ad |
| Audited by | © cyberscope |

# Analysis

● Critical ● Medium ● Minor / Informative ● Pass

| Severity | Code | Description | Status |
|---|---|---|---|
| ● | ST | Stops Transactions | Passed |
| ● | 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 |
|---|---|---|---|
| ⚪ | MVN | Misleading Variable Naming | Unresolved |
| ⚪ | RCS | Redundant Code Segment | Unresolved |
| ⚪ | OCTD | Transfers Contract's Tokens | Unresolved |
| ⚪ | L04 | Conformance to Solidity Naming Conventions | Unresolved |

# Table of Contents

# Review

| | |
|---|---|
| **Contract Name** | Stelnar |
| **Compiler Version** | v0.8.20+commit.a1b79de6 |
| **Optimization** | 200 runs |
| **Explorer** | https://testnet.bscscan.com/address/0xcd224393ce8ccd7b034dbe9619ddfd11ade121ad |
| **Address** | 0xcd224393ce8ccd7b034dbe9619ddfd11ade121ad |
| **Network** | BSC_TESTNET |
| **Symbol** | STL |
| **Decimals** | 18 |
| **Total Supply** | 10,000,000,000 |
| **Badge Eligibility** | Yes |

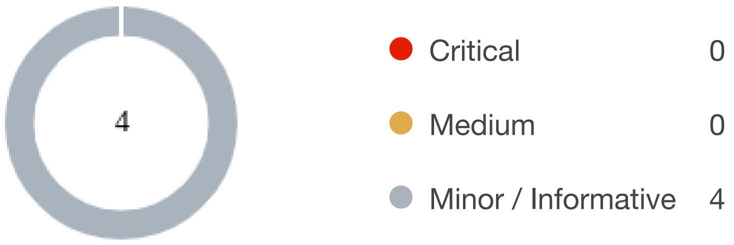
## Audit Updates

| | |
|---|---|
| **Initial Audit** | 16 Apr 2024 |
| **Corrected Phase 2** | 20 Apr 2024 |

## Source Files

| **Filename** | SHA256 |
|---|---|
| **Stelnar.sol** | c3fb3445b51111709f94057625955d2f98c0565af2e181b033b837b0e6fe47da |

# Findings Breakdown



| Severity | Unresolved | Acknowledged | Resolved | Other |
|---|---|---|---|---|
| ● Critical | 0 | 0 | 0 | 0 |
| ● Medium | 0 | 0 | 0 | 0 |
| ● Minor / Informative | 4 | 0 | 0 | 0 |

# MVN - Misleading Variable Naming

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | Stelnar.sol#L233,250 |
| **Status** | Unresolved |

## Description

The contract is employing the `_maxBuyAmount` variable to limit transaction amounts in both buy and sell scenarios, contrary to what the variable name suggests. This variable is checked against the transaction value in conditions that identify either a buying or a selling event. The use of `_maxBuyAmount` for both types of transactions can lead to confusion and misinterpretation of the code, as users might expect that such a specifically named variable would be exclusive to buy transactions only.

```
// when Buy
if (_isBTaxEnabled && _automatedMarketMakerPairs[from]) {
        require(
            value <= _maxBuyAmount,
            "Transfer amount exceeds the maxTxAmount."
        );
    }
...
// when sell
else if (_isSTaxEnabled && _automatedMarketMakerPairs[to]) {
        require(
            value <= _maxBuyAmount,
            "Transfer amount exceeds the maxTxAmount."
        );
    }
```

## Recommendation

It is recommended to rename the variable to reflect the actual usages across different transaction types. A more neutral and descriptive name like `_maxTransactionAmount` would eliminate any ambiguity concerning the variable's purpose and application. This change will enhance the clarity and readability of the contract, thereby reducing the potential for errors and misunderstandings in the management and auditing of the contract.

Ensuring precise and intuitive naming conventions is crucial for maintaining best practices in smart contract development.

## RCS - Redundant Code Segment

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | Stelnar.sol#L235 |
| **Status** | Unresolved |

### Description

The smart contract contains a section of code that has been commented out, indicating a deliberate decision to disable certain functionality. Having commented-out code segments in a contract presents several drawbacks. Firstly, it can confuse developers and auditors, leading to misunderstandings about the purpose of the code and whether it should be reactivated in the future. Maintenance becomes more challenging over time, as outdated or irrelevant code remains in the contract, detracting from readability and clarity. Furthermore, commented-out code can cause version control issues and make it harder to track meaningful changes. Therefore, it's best practice to remove commented-out code entirely, reducing complexity, improving readability.

```
// if (_transferDelay && uniswapV2Pair != address(0)) {
//     if (to != _uniswapV2Router && to != address(uniswapV2Pair)) {
//         require(
//             _holderLastTransferTimestamp[tx.origin] <
//                 block.number - 4 &&
//                 _holderLastTransferTimestamp[to] < block.number - 4,
//             "_transfer:: Transfer Delay enabled.  Try again later."
//         );

//         _holderLastTransferTimestamp[tx.origin] = block.number;
//         _holderLastTransferTimestamp[to] = block.number;
//         emit TransferDelay(to, block.number, block.number + 5);
//     }
// }
```

### Recommendation

We recommend removing the commented-out code segments related to the transfer delay mechanism, as they are currently inactive and add unnecessary complexity to the codebase. This action will streamline the contract, enhance code readability, and reduce the risk of confusion during the development process.

## OCTD - Transfers Contract's Tokens

| | |
|---|---|
| **Criticality** | Minor / Informative |
| **Location** | Stelnar.sol#L584 |
| **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 `transferForeignToken` function.

```
    function transferForeignToken(
        address _token,
        address _to
    ) external onlyOwner {
        require(_token != address(0), "_token address cannot be 0");
        uint256 _contractBalance =
IERC20(_token).balanceOf(address(this));
        IERC20(_token).safeTransfer(_to, _contractBalance);
    }
```

### 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** | src/Stelnar.sol#L83,194,438,445,482,491,501,518,524,569,592,593 |
| **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.

```
nction WETH() external pure returns (address);


dress _owner,

nt256 _newTax)
...


dress _token,


dress _to
```

## Recommendation

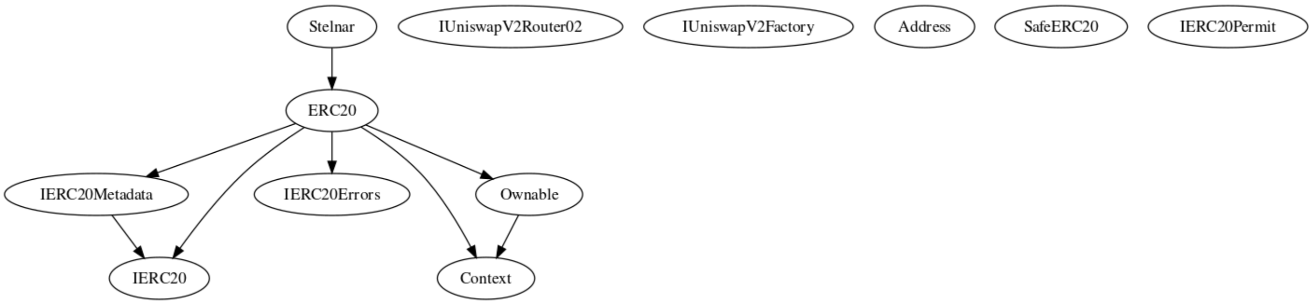
By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation
https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention.

# Functions Analysis

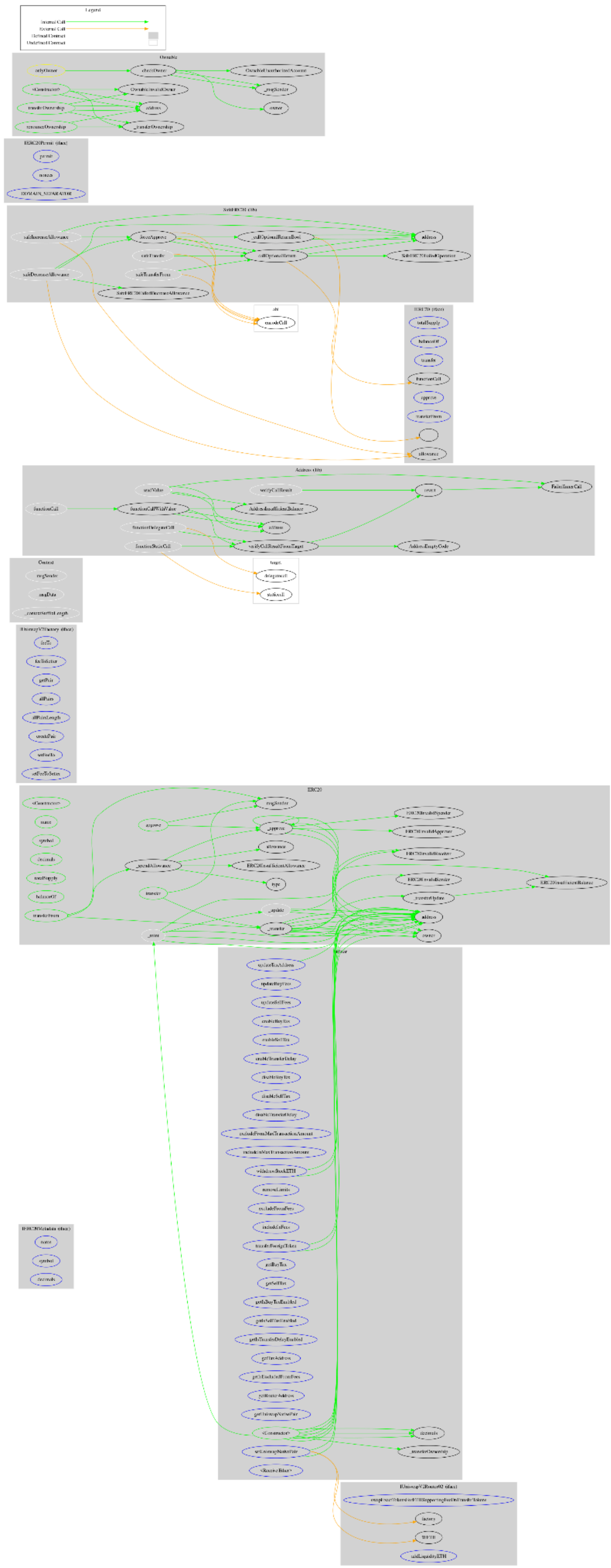
| Contract | Type | Bases | | |
|---|---|---|---|---|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| **Stelnar** | Implementation | ERC20 | | |
| | | Public | ✓ | ERC20 |
| | updateBuyFees | External | ✓ | onlyOwner |
| | updateSellFees | External | ✓ | onlyOwner |
| | enableBuyTax | External | ✓ | onlyOwner |
| | enableSellTax | External | ✓ | onlyOwner |
| | enableTransferDelay | External | ✓ | onlyOwner |
| | disableBuyTax | External | ✓ | onlyOwner |
| | disableSellTax | External | ✓ | onlyOwner |
| | disableTransferDelay | External | ✓ | onlyOwner |
| | excludeFromMaxTransactionAmount | External | ✓ | onlyOwner |
| | includeInMaxTransactionAmount | External | ✓ | onlyOwner |
| | updateTaxAddress | External | ✓ | onlyOwner |
| | removeLimits | External | ✓ | onlyOwner |
| | excludeFromFees | External | ✓ | onlyOwner |
| | includeInFees | External | ✓ | onlyOwner |
| | setUniswapNativePair | External | ✓ | onlyOwner |
| | getBuyTax | External | | - |
| | getSellTax | External | | - |

| | | | | |
|---|---|---|---|---|
| | getIsBuyTaxEnabled | External | | - |
| | getIsSellTaxEnabled | External | | - |
| | getIsTransferDelayEnabled | External | | - |
| | getTaxAddress | External | | - |
| | getIsExcludedFromFees | External | | - |
| | getRouterAddress | External | | - |
| | getUniswapNativePair | External | | - |
| | withdrawStuckETH | External | ✓ | onlyOwner |
| | transferForeignToken | External | ✓ | onlyOwner |
| | | External | Payable | - |

# Inheritance Graph



# Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Ownable
onlyOwner
_checkOwner
OwnableUnauthorizedAccount
<Constructor>
OwnableInvalidOwner
_msgSender
transferOwnership
address
owner
renounceOwnership
_transferOwnership
IERC20Permit (iface)
permit
nonces
DOMAIN_SEPARATOR
SafeERC20 (lib)
safeIncreaseAllowance
forceApprove
_callOptionalReturnBool
address
safeTransfer
_callOptionalReturn
SafeERC20FailedOperation
safeDecreaseAllowance
safeTransferFrom
SafeERC20FailedDecreaseAllowance
abi
encodeCall
IERC20 (iface)
totalSupply
balanceOf
transfer
functionCall
approve
transferFrom
allowance
Address (lib)
sendValue
verifyCallResult
revert
FailedInnerCall
functionCall
functionCallWithValue
AddressInsufficientBalance
functionDelegateCall
address
functionStaticCall
verifyCallResultFromTarget
AddressEmptyCode
Context
_msgSender
_msgData
_contextSuffixLength
target
delegatecall
staticcall
IUniswapV2Factory (iface)
feeTo
feeToSetter
getPair
allPairs
allPairsLength
createPair
setFeeTo
setFeeToSetter
ERC20
<Constructor>
_msgSender
name
approve
ERC20InvalidSpender
symbol
_approve
ERC20InvalidApprover
decimals
allowance
ERC20InvalidReceiver
totalSupply
_spendAllowance
ERC20InsufficientAllowance
balanceOf
type
ERC20InvalidSender
ERC20InsufficientBalance
transfer
_transferUpdate
transferFrom
_update
address
_transfer
owner
_mint
Stelnar
updateTaxAddress
updateBuyFees
updateSellFees
enableBuyTax
enableSellTax
enableTransferDelay
disableBuyTax
disableSellTax
disableTransferDelay
excludeFromMaxTransactionAmount
includeInMaxTransactionAmount
withdrawStuckETH
removeLimits
excludeFromFees
includeInFees
transferForeignToken
getBuyTax
getSellTax
getIsBuyTaxEnabled
getIsSellTaxEnabled
getIsTransferDelayEnabled
getTaxAddress
getIsExcludedFromFees
getRouterAddress
getUniswapNativePair
<Constructor>
decimals
setUniswapNativePair
_transferOwnership
<Receive Ether>
IERC20Metadata (iface)
name
symbol
decimals
IUniswapV2Router02 (iface)
swapExactTokensForETHSupportingFeeOnTransferTokens
factory
WETH
addLiquidityETH

# Summary

Stelnar 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. There is also a limit of max 20% 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