Cyberscope

# Audit Report

# EverChain

October 2022

| | |
|---|---|
| Type | BEP20 |
| Network | BSC |
| Address | 0xBEC511e6f74989002aA2BE18a651bFB290cD911D |
| Audited by | © cyberscope |

# Table of Contents

# Contract Review

| | |
|---|---|
| **Contract Name** | Everchain |
| **Compiler Version** | v0.8.6+commit.11564f7e |
| **Optimization** | 200 runs |
| **Licence** | MIT |
| **Explorer** | https://bscscan.com/token/0xbec511e6f74989002aa2be18a651bfb290cd911d |
| **Symbol** | EC |
| **Decimals** | 9 |
| **Total Supply** | 100,000,000 |
| **Domain** | ever-chain.io |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 1st October 2022 |
| **Corrected** | |

# Source Files

| Filename | SHA256 |
|---|---|
| **Context.sol** | f24761813202348c0b68d767e1bbc488191d654b526f5e817af9af54525abc99 |
| **DividendPayingToken.sol** | e48cb557a615dc346a725ca248fe8610145eaca7ca9f4d3d66a4ea28c297d703 |
| **DividendPayingTokenInterface.sol** | d7a558ad379fbd0b0b1822aebd104e2fa99ff7c3eac13b4ee45fdd34eb29a23f |
| **ERC20.sol** | f2afd1560d4b5cfa574c4ae7189da11f7d31d052e92a9d7755b8b4fbb47fadab |
| **Everchain.sol** | b4e5b39b72556765f68a9058af10eddf26b98699d8760b10b627492e9077864e |
| **IDex.sol** | f42c60eab527a2a8aab45407c03773c249237056e0f712c7949fd81099814e60 |
| **IERC20.sol** | 7c69e0bf19c4248ee1982923d6a421abcad9c5741ab499f344c64b23fd3e1001 |
| **IterableMapping.sol** | cc86bca02e7cca2407c00a505d463e89ae274d5cfe3664f152ec30f9ca5b1bc6 |
| **Ownable.sol** | a88be4357fa62460235dfd732182e02f1ecbf645abe05fc67f1ebcd1c23d0672 |
| **SafeMath.sol** | f39d9ee58c3ad0f46c8a1886875a5de9e833d8a97f6957075e18cf7ca6a3de27 |

# Contract 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 | Unresolved |
| ● | MT | Mints Tokens | Passed |
| ● | BT | Burns Tokens | Passed |
| ● | BC | Blacklists Addresses | Passed |

# ULTW - Transfers Liquidity to Team Wallet

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L157 |
| **Status** | Unresolved |

## Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `forceSend` method.

```
function forceSend() external {
    uint256 BNBbalance = address(this).balance;
    payable(owner()).sendValue(BNBbalance);
}
```

## 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 |
| | L02 | State Variables could be Declared Constant | Unresolved |
| | L04 | Conformance to Solidity Naming Conventions | Unresolved |
| | L05 | Unused State Variable | Unresolved |
| | L07 | Missing Events Arithmetic | Unresolved |
| | L11 | Unnecessary Boolean equality | Unresolved |
| | L12 | Using Variables before Declaration | Unresolved |
| | L13 | Divide before Multiply Operation | Unresolved |
| | L14 | Uninitialized Variables in Local Scope | Unresolved |

# L01 - Public Function could be Declared External

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L116,176,286,290,298,627,658,674 |
| **Status** | Unresolved |

## Description

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

```
updateDividendTracker
excludeMultipleAccountsFromFees
isExcludedFromFees
withdrawableDividendOf
dividendTokenBalanceOf
getAccountAtIndex
setBalance
process
```

## Recommendation

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

# L02 - State Variables could be Declared Constant

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L37,61 |
| **Status** | Unresolved |

## Description

Constant state variables should be declared constant to save gas.

```
currentRewardToken
launchtax
```

## Recommendation

Add the constant attribute to state variables that never change.

# L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L207,212,219,32,582 |
| **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.

```
_rewards
_marketing
_liquidity
_enabled
deadWallet
_account
```

## Recommendation

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

# L05 - Unused State Variable

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L37 |
| **Status** | Unresolved |

## Description

There are segments that contain unused state variables.

```
currentRewardToken
```

## Recommendation

Remove unused state variables.

# L07 - Missing Events Arithmetic

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L202,229 |
| **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.

```
swapTokensAtAmount = amount * 10 ** 9
antiBotBlocks = numberOfBlocks
```

## Recommendation

Emit an event for critical parameter changes.

# L11 - Unnecessary Boolean equality

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L544 |
| **Status** | Unresolved |

## Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
value == true
```

## Recommendation

Remove the equality to the boolean constant.

# L12 - Using Variables before Declaration

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L425,424,423 |
| **Status** | Unresolved |

## Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
lastProcessedIndex
claims
iterations
```

## Recommendation

The variables should be declared before any usage of them.

# L13 - Divide before Multiply Operation

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L439 |
| **Status** | Unresolved |

## Description

Performing divisions before multiplications may cause lose of prediction.

```
unitBalance = deltaBalance / (denominator - sellTaxes.liquidity)
```

## Recommendation

The multiplications should be prior to the divisions.

# L14 - Uninitialized Variables in Local Scope

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | Everchain.sol#L425,423,397,424 |
| **Status** | Unresolved |

## Description

The are variables that are defined in the local scope and are not initialized.

```
lastProcessedIndex
iterations
swapAmt
claims
```

## Recommendation

All the local scoped variables should be initialized.

# Contract Functions

| Contract | Type | Bases | | |
|---|---|---|---|---|
| | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| | | | | |
| **Context** | Implementation | | | |
| | _msgSender | Internal | | |
| | _msgData | Internal | | |
| | | | | |
| **DividendPayingToken** | Implementation | ERC20, DividendPayingTokenInterface, Ownable | | |
| | <Constructor> | Public | ✓ | ERC20 |
| | <Receive Ether> | External | Payable | - |
| | distributeDividends | Public | Payable | - |
| | _withdrawDividendOfUser | Internal | ✓ | |
| | setRewardToken | External | ✓ | onlyOwner |
| | swapBnbForCustomToken | Internal | ✓ | |
| | dividendOf | Public | | - |
| | withdrawableDividendOf | Public | | - |
| | withdrawnDividendOf | Public | | - |
| | accumulativeDividendOf | Public | | - |
| | _transfer | Internal | ✓ | |
| | _tokengeneration | Internal | ✓ | |
| | _burn | Internal | ✓ | |
| | _setBalance | Internal | ✓ | |
| | | | | |
| **DividendPayingTokenInterface** | Interface | | | |
| | dividendOf | External | | - |
| | distributeDividends | External | Payable | - |
| | withdrawableDividendOf | External | | - |
| | withdrawnDividendOf | External | | - |

| | | | | |
|---|---|---|---|---|
| | accumulativeDividendOf | External | | - |
| | | | | |
| **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 | ✓ | |
| | _tokengeneration | Internal | ✓ | |
| | _burn | Internal | ✓ | |
| | _approve | Internal | ✓ | |
| | _beforeTokenTransfer | Internal | ✓ | |
| | | | | |
| **Address** | Library | | | |
| | sendValue | Internal | ✓ | |
| | | | | |
| **Everchain** | Implementation | ERC20, Ownable | | |
| | <Constructor> | Public | ✓ | ERC20 |
| | <Receive Ether> | External | Payable | - |
| | updateDividendTracker | Public | ✓ | onlyOwner |
| | processDividendTracker | External | ✓ | - |
| | claim | External | ✓ | - |
| | rescueBEP20Tokens | External | ✓ | onlyOwner |
| | forceSend | External | ✓ | - |
| | excludeFromFees | Public | ✓ | onlyOwner |

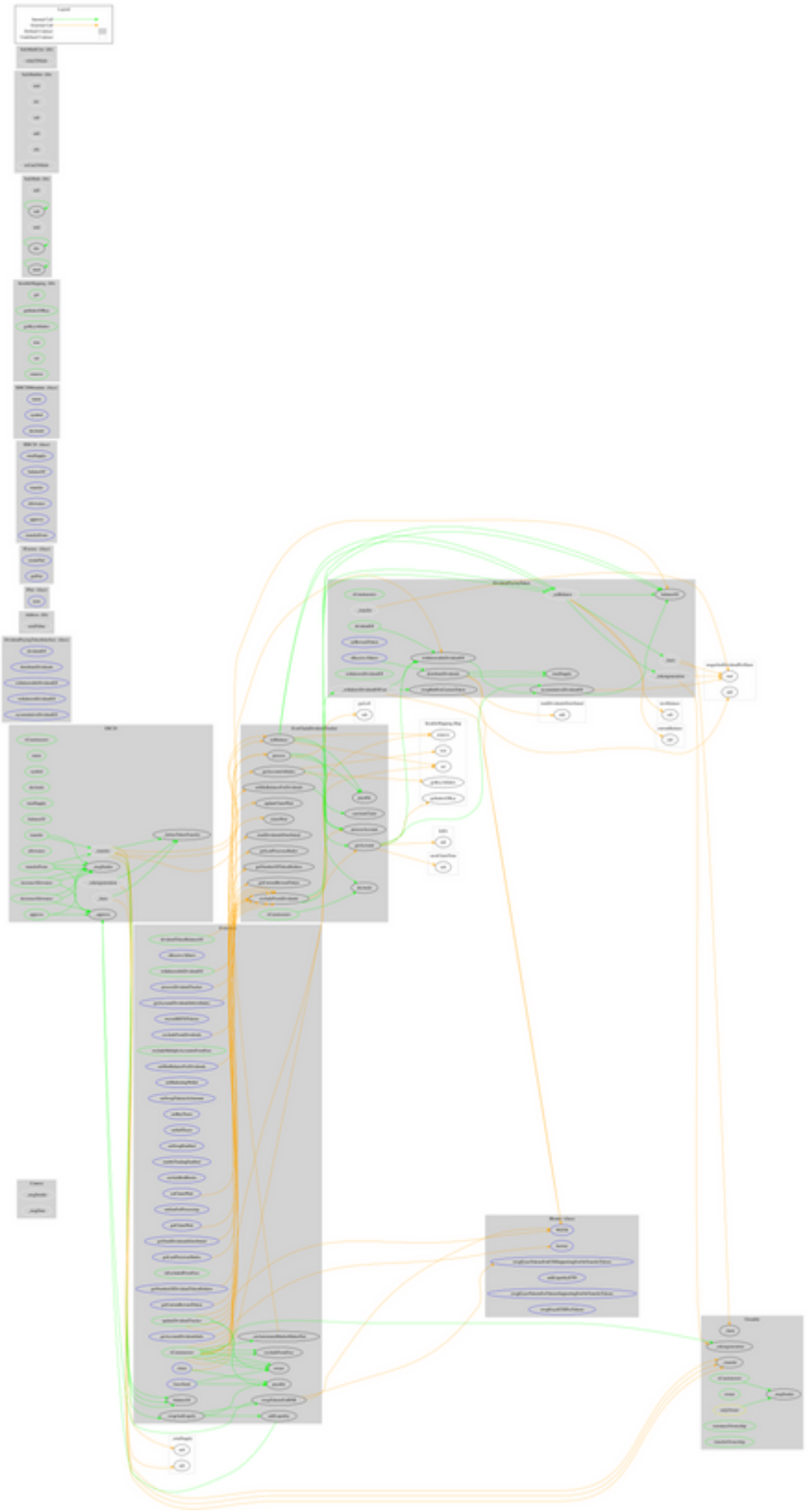
| | | | | |
|---|---|---|---|---|
| | excludeMultipleAccountsFromFees | Public | ✓ | onlyOwner |
| | excludeFromDividends | External | ✓ | onlyOwner |
| | setMarketingWallet | External | ✓ | onlyOwner |
| | setSwapTokensAtAmount | External | ✓ | onlyOwner |
| | setBuyTaxes | External | ✓ | onlyOwner |
| | setSellTaxes | External | ✓ | onlyOwner |
| | setSwapEnabled | External | ✓ | onlyOwner |
| | enableTradingEnabled | External | ✓ | onlyOwner |
| | setAntiBotBlocks | External | ✓ | onlyOwner |
| | setMinBalanceForDividends | External | ✓ | onlyOwner |
| | _setAutomatedMarketMakerPair | Private | ✓ | |
| | setGasForProcessing | External | ✓ | onlyOwner |
| | setClaimWait | External | ✓ | onlyOwner |
| | getClaimWait | External | | - |
| | getTotalDividendsDistributed | External | | - |
| | isExcludedFromFees | Public | | - |
| | withdrawableDividendOf | Public | | - |
| | getCurrentRewardToken | External | | - |
| | dividendTokenBalanceOf | Public | | - |
| | getAccountDividendsInfo | External | | - |
| | getAccountDividendsInfoAtIndex | External | | - |
| | getLastProcessedIndex | External | | - |
| | getNumberOfDividendTokenHolders | External | | - |
| | _transfer | Internal | ✓ | |
| | swapAndLiquify | Private | ✓ | |
| | swapTokensForBNB | Private | ✓ | |
| | addLiquidity | Private | ✓ | |
| | | | | |
| **EverChainDividendTracker** | Implementation | Ownable, DividendPayingToken | | |
| | <Constructor> | Public | ✓ | DividendPayingToken |
| | _transfer | Internal | | |
| | setMinBalanceForDividends | External | ✓ | onlyOwner |
| | excludeFromDividends | External | ✓ | onlyOwner |

| | | | | |
|---|---|---|---|---|
| | updateClaimWait | External | ✓ | onlyOwner |
| | getLastProcessedIndex | External | | - |
| | getNumberOfTokenHolders | External | | - |
| | getCurrentRewardToken | External | | - |
| | getAccount | Public | | - |
| | getAccountAtIndex | Public | | - |
| | canAutoClaim | Private | | |
| | setBalance | Public | ✓ | onlyOwner |
| | process | Public | ✓ | - |
| | processAccount | Public | ✓ | onlyOwner |
| | | | | |
| **IPair** | Interface | | | |
| | sync | External | ✓ | - |
| | | | | |
| **IFactory** | Interface | | | |
| | createPair | External | ✓ | - |
| | getPair | External | | - |
| | | | | |
| **IRouter** | Interface | | | |
| | factory | External | | - |
| | WETH | External | | - |
| | addLiquidityETH | External | Payable | - |
| | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ✓ | - |
| | swapExactETHForTokens | External | Payable | - |
| | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ✓ | - |
| | | | | |
| **IERC20** | Interface | | | |
| | totalSupply | External | | - |
| | balanceOf | External | | - |
| | transfer | External | ✓ | - |
| | allowance | External | | - |
| | approve | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | | | | |

| | | | | |
|---|---|---|---|---|
| **IERC20Metadata** | Interface | IERC20 | | |
| | name | External | | - |
| | symbol | External | | - |
| | decimals | External | | - |
| | | | | |
| **IterableMapping** | Library | | | |
| | get | Public | | - |
| | getIndexOfKey | Public | | - |
| | getKeyAtIndex | Public | | - |
| | size | Public | | - |
| | set | Public | ✓ | - |
| | remove | Public | ✓ | - |
| | | | | |
| **Ownable** | Implementation | Context | | |
| | <Constructor> | Public | ✓ | - |
| | owner | Public | | - |
| | renounceOwnership | Public | ✓ | onlyOwner |
| | transferOwnership | Public | ✓ | onlyOwner |
| | | | | |
| **SafeMath** | Library | | | |
| | add | Internal | | |
| | sub | Internal | | |
| | sub | Internal | | |
| | mul | Internal | | |
| | div | Internal | | |
| | div | Internal | | |
| | mod | Internal | | |
| | mod | Internal | | |
| | | | | |
| **SafeMathInt** | Library | | | |
| | mul | Internal | | |
| | div | Internal | | |
| | sub | Internal | | |
| | add | Internal | | |

| | | | | |
|---|---|---|---|---|
| | abs | Internal | | |
| | toUint256Safe | Internal | | |
| | | | | |
| **SafeMathUint** | Library | | | |
| | toInt256Safe | Internal | | |

# Contract Flow



# Domain Info

| | |
|---|---|
| **Domain Name** | ever-chain.io |
| **Registry Domain ID** | f7bd5e9036ac46faa120e97f0ca75c8d-DONUTS |
| **Creation Date** | 2022-09-28T11:54:58Z |
| **Updated Date** | 2022-09-28T11:55:21Z |
| **Registry Expiry Date** | 2023-09-28T11:54:58Z |
| **Registrar WHOIS Server** | whois.namecheap.com |
| **Registrar URL** | https://www.namecheap.com/ |
| **Registrar** | NameCheap, Inc. |
| **Registrar IANA ID** | 1068 |

The domain was created 3 days before the creation of the audit. It will expire in 12 months.

There is no public billing information, the creator is protected by the privacy settings.

# Summary

The Smart Contract analysis reported one minor severity issue. The contract owner has the authority to transfer funds to the team's wallet. Other than that, the contract owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions.

# 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