



Smart Contract Analysis

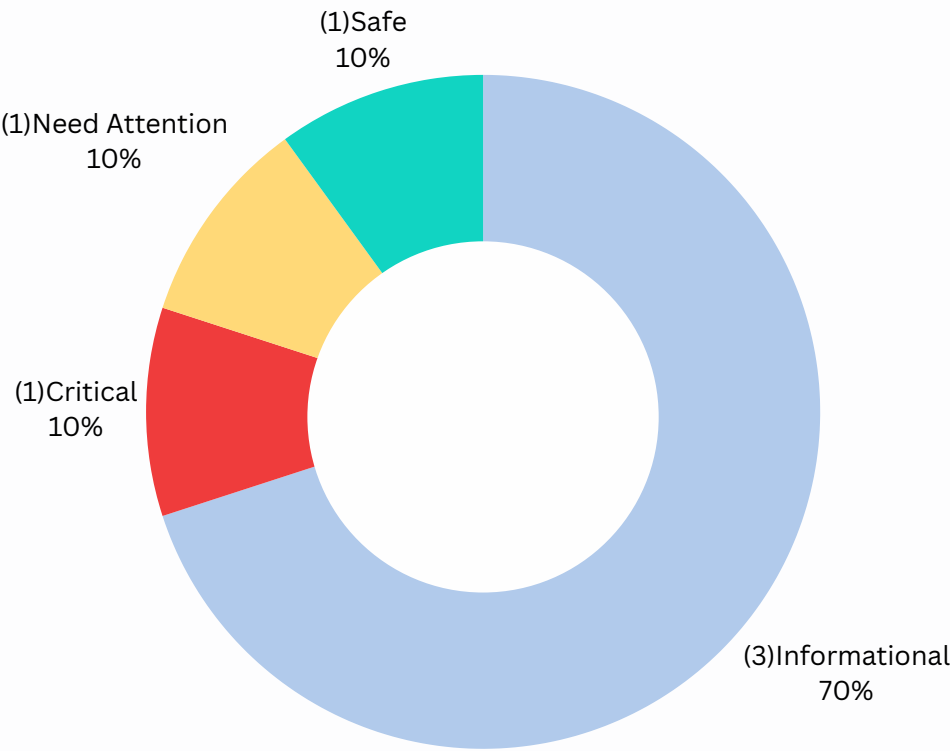
# Cunetwork Token Audit

Verified by Secureverse audit on Dec 27th, 2023

Verified by Secureverse audit on Dec 27th, 2023

# Cunetwork Token Audit

Report and security audit prepared by Secureverse



Contract score : 89 / 100

Liquidity score : 0 / 100

Checked vulnerabilities

Passed Failed Moderate

- Re-entrancy

Access control

Denial of service

Timestamp

Integer overflow

Liquidity

Dumping Ownership
- Wallet lock

Transfer lock

Mintable risk

Liquidity pairs

Rugpull

Honeypot

## Disclaimer | Cunetwork Token Audit

The team has examined this smart contract in accordance with industry best practices. We made every effort to secure the code and provide this report. audits done by smart contract auditors and automated algorithms; however, it is crucial to remember that you should not rely entirely on this report. The smart contract may have flaws that allow for hacking. As a result, the audit cannot ensure the explicit security of the audited smart contracts. The experts and its audit report do not encourage readers to consider them as providing any project-related financial or legal advice.

Audit Scope | Cunetwork Token Audit

Ticker	File name	Transaction hash
CUN	cunetwork.sol	0x8043e9044ddfd4b8ca3925e2e6245daad2d15f6f81ea1349ca2ac8108584e5a9

## Approach and method | Cunetwork Token Audit

The smart contract audit is performed by the experts and they make sure that audited project must comply with the industry security standards.

Process :

- In depth review of the white paper
- In depth analysis of project and code documentation.
- Checking the industry standards used in Code/Project.
- Checking and understanding Core Functionality of the Code.
- Comparing the code with documentation.
- Static analysis of the code.
- Manual analysis of the code.
- Gas Optimization and Function Testing.
- Verification of the overall audit.
- Report writing.

### Static Analysis

- Static analysis has been done by using the open source and state of the art
- Automatic smart contract vulnerability scanning tools.
- Manual Analysis
- Manual analysis is done by our smart contract auditors' team by performing in depth analysis of the smart contract and identify potential vulnerabilities.

Auditor also review and verify all the static analysis results to prevent the false positives identified by automated tools.

- Gas Consumption and Function Testing
- Function testing done by auditors by manually writing customized test cases for the smart contract to verify the intended behavior as per code and documentation. Gas Optimization done by reviews potential gas consumption by contract in production.

## Centralization Risk | Cunetwork Token Audit

Owner of the contract has authority of changing the admin1, admin2 and admin3 addresses and the ownership.

- setAdminaddresses -

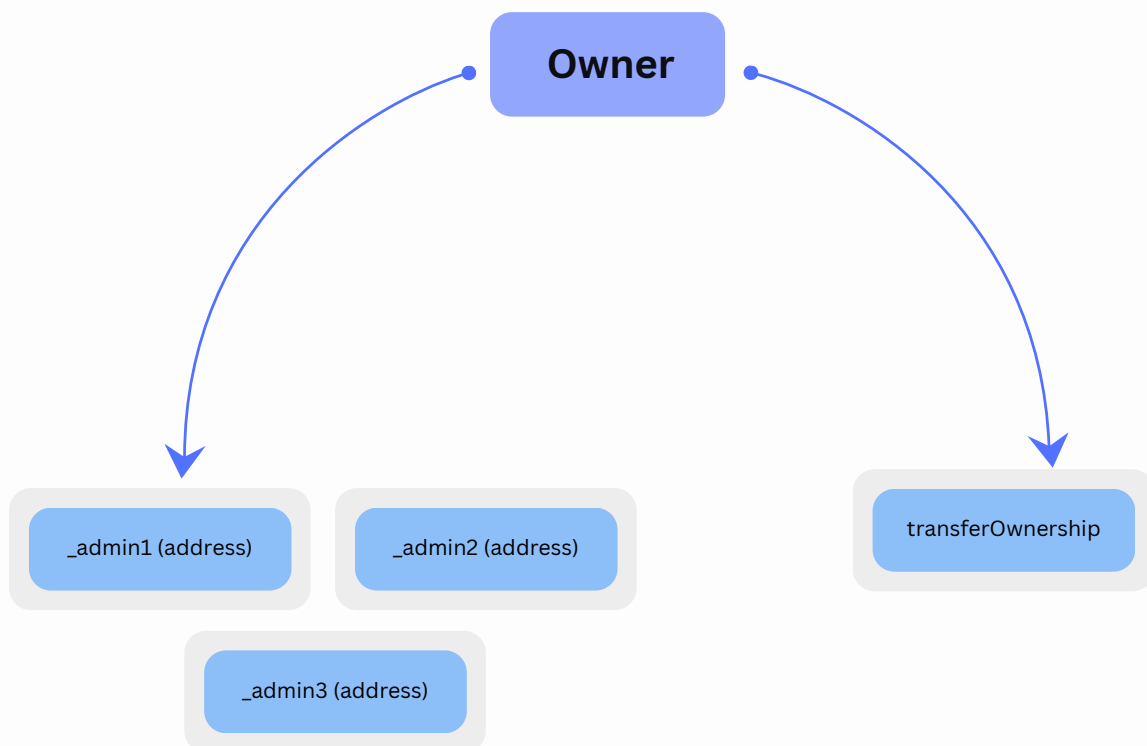
1.\_admin1 (address)

2.\_admin2 (address)

3.\_admin3 (address)

- transferOwnership -

Ownership can be transferred without any re-entrance access to the old owner



- No centralization of Liquidity & holdings found in contract.
- No authority of owner over liquidity removal or withdrawal.
- Admin1, admin2, admin3 can sell the tokens from their wallet

Severity : No severity found

## Buy Tokens | Cunetwork Token Audit

Buy Tokens Can be called by users.

Required arguments to execute the function are as follows ;

- amountInUSDT (uint256)
- \_referrer (address)

### Description

Buy Tokens function is executed by users. Approval is needed from USDT contract to complete the transaction. 18 Decimal USDT contract pegged with binance smart chain is internally called upon the deployment. Buy tokens

### Actions triggered upon Buy Tokens

- 1.userBuys[msg.sender].push(BuyDetail)
- 2.userLevelUnlock
- 3.SetReferrer
4.  $\text{uint LpPoolAmount} = (\text{amountInUSDT} * 85) / 100;$
- 5.MintTokens
- 6.DistributeRewardsandFees

Severity found : Informational

- Public functions should be declared External
- State variable should be declared constant

**Status : Acknowledged**

## Liquidity Risk | Cunetwork Token Audit

No liquidity pairs found in the contract. Which makes the liquidity score of the contract to zero.

- Pancakeswap router pair : Not found
- Uniswap router pairs : Not found
- Liquidity stored : Mapping of the contract **UsdtPool**

Severity found : Critical & needs to be acknowledged

**Status : Acknowledged**



## Mint Tokens | Cunetwork Token Audit

Token has no fixed or maximum supply. Tokens are minted without any limitations of maximum total supply. This certain event is triggered upon BuyToken function. Minted supply is sent to the respective wallets set in the contract and to the user.

Constructor is defined as payable which may affect the gas fees while signing the transaction.

Severity found : Medium & needs to be acknowledged

**Status : Acknowledged**

## General Risk | Cunetwork Token Audit

- Division before multiplication

### Issue Location in Code

CUNetwork.mintTokens(uint256) (CUNetwork.sol#553-570)  
performs a multiplication on the result of a division:  
- tokenPerUSDT = 1e18 / TokenPrice (CUNetwork.sol#554)  
- amountForTokenUserConversion = (amountInUSDT \* 50) / 100  
(CUNetwork.sol#555)  
- tokensToMintForUser = amountForTokenUserConversion \*  
tokenPerUSDT (CUNetwork.sol#556)

The order of operations used may result in a loss of precision

- Missing Arithmetic Events

### Issue Location in Code

CUNetwork.BuyTokens(uint256,address) (CUNetwork.sol#453-530)  
should **emit** an **event** for:  
- UsdtPool += LpPoolAmount (CUNetwork.sol#517)

This contract is missing useful arithmetic events.

- Unchecked Transfer

### Issue Location in Code

CUNetwork.withdraw(uint256) (CUNetwork.sol#573-611) ignores  
**return** value by USDT.transfer(admin1\_address,amountForAdmin1)  
(CUNetwork.sol#598)

A transfer call made in this contract may be unstable and cause tokens to become stuck in contract in some cases.

Severity found : Informational

**Status : Acknowledged**

## Other Findings | Cunetwork Token Audit

- No compiler version inconsistencies found
- No unchecked call responses found
- No vulnerable self-destruct functions found
- No assertion vulnerabilities found
- No old solidity code found
- No external delegated calls found
- No external call dependency found
- No vulnerable authentication calls found
- No dead code found
- No risky data allocation found
- No uninitialized storage variables found
- No risky data handling found

Summary | Cunetwork Token Audit

Title	Category	Status
Centralization	Safe	Acknowledged
Buy Tokens	Informational	Acknowledged
Liquidity	Critical	Acknowledged & Mitigated
Note : Issue is mitigated as the respective token is not going to be listed on any exchange.		
Mint Tokens	Informational	Acknowledged
General Risks	Informational	Acknowledged
Other Findings	Positive	Acknowledged

## Conclusion | Cunetwork Token Audit

In this audit, we examined the CUN 's smart contract with the framework, and we discovered several medium and informational flaws in the smart contract. We have included solutions and recommendations in the audit report to improve the quality and security posture of the code. All of the findings and solutions have been acknowledged by the project team.