QuillAudits

Audit Report
June, 2021

Contents

Introduction

This Audit Report highlights the overall security of the CHAMPIONS LEAGUE Smart Contract. With this report, we have tried to ensure the reliability of their smart contract by a complete assessment of their system's architecture and the smart contract codebase.

Auditing Approach and Methodologies applied

The QuillAudits team has performed thorough testing of the project, starting with analysing the code design patterns in which we reviewed the smart contract architecture to ensure it is structured and safe use of third party smart contracts and libraries.

Our team then performed a formal line by line inspection of the Smart Contract to find any potential issue like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

In the Unit testing Phase, we coded/conducted Custom unit tests written for each function in the contract to verify that each function works as expected. In Automated Testing, We tested the Smart Contract with our in-house developed tools to identify vulnerabilities and security flaws.

The code was tested in collaboration with our multiple team members, and this included -

- Testing the functionality of the Smart Contract to determine proper logic has been followed throughout the process.
- Analysing the complexity of the code by thorough, manual review of the code, line-by-line.
- Deploying the code on testnet using multiple clients to run live tests
- Analysing failure preparations to check how the Smart Contract performs in case of bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.
- Analysing the security of the on-chain data.

Audit Details

Project Name: CHAMPIONS LEAGUE
BscScan: 0xDeEa8fdB85503531F6A5D2A036d3268bD7C06c09
Languages: Solidity (Smart contract), Javascript (Unit Testing)
Platforms and Tools: Remix IDE, Truffle, Truffle Team, Ganache, Slither, Surya

Summary of Smart Contract

QuillAudits conducted a security audit of a smart contract of CHAMPIONS LEAGUE. CHAMPIONS LEAGUE contracts are used to create smart wallets, registry contract to register wallet addresses and forwardProxy contract.

And some advanced features other than essential functions.

- Wallet
- Forwards signed transactions to the user's wallet for executing logic.
- Deploy wallets and track

Audit Goals

The focus of the audit was to verify that the smart contract system is secure, resilient and working according to its specifications. The audit activities can be grouped into the following three categories:

Security
Identifying security related issues within each contract and the system of contract.

Sound Architecture
Evaluation of the architecture of this system through the lens of established smart contract best practices and general software best practices.

Code Correctness and Quality

A full review of the contract source code. The primary areas of focus include:

- Accuracy
- Readability
- Sections of code with high complexity
- Quantity and quality of test coverage

Security Level references

Every issue in this report was assigned a severity level from the following:

High level severity issues

Issues on this level are critical to the smart contract's performance/ functionality and should be fixed before moving to a live environment.

Medium level severity issues

Issues on this level could potentially bring problems and should eventually be fixed.

Low level severity issues

Issues on this level are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

Number of issues per severity

	Low	Medium	High	Recommendations
Open	3	0	0	0
Closed	0	0	0	0

Manual Audit

High level severity issues

No high severity issues

Medium level severity issues

No medium severity issues

Low level severity issues

1. **Variable shadowing**

   Check: shadowing-state
   Severity: High
   Confidence: High

   CLEToken.allowance.owner (local variable @ CLE.sol#) shadows:
   - Ownable.owner (function @ CLE.sol#410-412)

   CLEToken._approve.owner (local variable @ CLE.sol#) shadows:
   - Ownable.owner (function @ CLE.sol#410-412)

   Remove the state variable shadowing.

   Status: Open

2. **Calls Inside a loop**

   Check: calls-loop
   Severity: Low
   Confidence: Medium

```
function includeInReward(address account) external onlyOwner() {
        require(_isExcluded[account], "Account is already included");
        for (uint256 i = 0; i < _excluded.length; i++) {
            if (_excluded[i] == account) {
                _excluded[i] = _excluded[_excluded.length - 1];
                _tOwned[account] = 0;
                _isExcluded[account] = false;
                _excluded.pop();
                break;
            }
        }
    }
 _excluded.pop();
```

Calls inside a loop might lead to a denial-of-service attack.

Recommendation
Favor pull over push strategy for external calls.

Status: Open

3. **Functions that can be declared external**

Ownable.renounceOwnership (CLE.sol#429-432) should be declared external
Ownable.transferOwnership (CLE.sol#438-442) should be declared external
Ownable.geUnlockTime (CLE.sol#444-446) should be declared external
Ownable.lock (CLE.sol#449-454) should be declared external
Ownable.unlock (CLE.sol#457-462) should be declared external
CLEToken.totalSupply (CLE.sol#754-756) should be declared external
IERC20.totalSupply (CLE.sol#48) should be declared external
IERC20.transfer (CLE.sol#62) should be declared external
CLEToken.transfer (CLE.sol#763-766) should be declared external
CLEToken.allowance (CLE.sol#) should be declared external
IERC20.allowance (CLE.sol) should be declared external
CLEToken.approve (CLE.sol#) should be declared external
IERC20.approve (CLE.sol#) should be declared external

CLEToken.transferFrom (CLE.sol#) should be declared external
IERC20.transferFrom (CLE.sol#98) should be declared external
CLEToken.name (CLE.sol#742-744) should be declared external
CLEToken.symbol (CLE.sol#746-748) should be declared external
CLEToken.decimals (CLE.sol#750-752) should be declared external
CLEToken.increaseAllowance (CLE.sol#783-786) should be declared external
CLEToken.decreaseAllowance (CLE.sol#788-791) should be declared external
CLEToken.isExcludedFromReward (CLE.sol#) should be declared external
CLEToken.totalFees (CLE.sol#) should be declared external
CLEToken.deliver (CLE.sol#) should be declared external
CLEToken.reflectionFromToken (CLE.sol#) should be declared external
CLEToken.excludeFromReward (CLE.sol#) should be declared external
CLEToken.excludeFromFee (CLE.sol#) should be declared external
CLEToken.includeInFee (CLE.sol) should be declared external
CLEToken.setSwapAndLiquifyEnabled (CLE.sol) should be declared external
CLEToken.isExcludedFromFee (CLE.sol#) should be declared external

Status: Open

Automated Testing

Slither Tool Result

```
INFO:Detectors:
Function 'Context._msgSender' (CLE.sol#275-277) is not in mixedCase
Function 'Context._msgData' (CLE.sol#279-282) is not in mixedCase
Parameter '' of IUniswapV2Factory.allPairs (CLE.sol#474) is not in mixedCase
Parameter '' of IUniswapV2Factory.setFeeTo (CLE.sol#479) is not in mixedCase
Parameter '' of IUniswapV2Factory.setFeeToSetter (CLE.sol#480) is not in mixedCase
Function 'IUniswapV2Pair.DOMAIN_SEPARATOR' (CLE.sol#500) is not in mixedCase
Function 'IUniswapV2Pair.PERMIT_TYPEHASH' (CLE.sol#501) is not in mixedCase
Function 'IUniswapV2Pair.MINIMUM_LIQUIDITY' (CLE.sol#518) is not in mixedCase
Parameter '' of IUniswapV2Pair.initialize (CLE.sol#533) is not in mixedCase
Parameter '_scope_0' of IUniswapV2Pair.initialize (CLE.sol#533) is not in mixedCase
Function 'IUniswapV2Router01.WETH' (CLE.sol#539) is not in mixedCase
Function 'CLEToken._transferBothExcluded' (CLE.sol#850-860) is not in mixedCase
Parameter '_enabled' of CLEToken.setSwapAndLiquifyEnabled (CLE.sol#898) is not in mixedCase
Function 'CLEToken._reflectFee' (CLE.sol#906-909) is not in mixedCase
Function 'CLEToken._getValues' (CLE.sol#911-915) is not in mixedCase
Function 'CLEToken._getTValues' (CLE.sol#917-923) is not in mixedCase
Function 'CLEToken._getRValues' (CLE.sol#925-932) is not in mixedCase
Function 'CLEToken._getRate' (CLE.sol#934-937) is not in mixedCase
Function 'CLEToken._getCurrentSupply' (CLE.sol#939-949) is not in mixedCase
Function 'CLEToken._takeLiquidity' (CLE.sol#951-957) is not in mixedCase
Function 'CLEToken._takeCommunity' (CLE.sol#959-965) is not in mixedCase
Parameter '_amount' of CLEToken.calculateTaxFee (CLE.sol#967) is not in mixedCase
Parameter '_amount' of CLEToken.calculateCommunityFee (CLE.sol#973) is not in mixedCase
Parameter '_amount' of CLEToken.calculateLiquidityFee (CLE.sol#979) is not in mixedCase
Function 'CLEToken._approve' (CLE.sol#1007-1013) is not in mixedCase
Function 'CLEToken._transfer' (CLE.sol#1015-1059) is not in mixedCase
Function 'CLEToken._tokenTransfer' (CLE.sol#1102-1120) is not in mixedCase
Function 'CLEToken._transferStandard' (CLE.sol#1122-1130) is not in mixedCase
Function 'CLEToken._transferToExcluded' (CLE.sol#1132-1141) is not in mixedCase
Function 'CLEToken._transferFromExcluded' (CLE.sol#1143-1152) is not in mixedCase
Variable 'CLEToken._taxFee' (CLE.sol#698) is not in mixedCase
Variable 'CLEToken._communityFee' (CLE.sol#701) is not in mixedCase
Variable 'CLEToken._liquidityFee' (CLE.sol#703) is not in mixedCase
Variable 'CLEToken._maxTxAmount' (CLE.sol#712) is not in mixedCase
Ownable.transferOwnership (CLE.sol#438-442) should be declared external
Ownable.geUnlockTime (CLE.sol#444-446) should be declared external
Ownable.lock (CLE.sol#449-454) should be declared external
Ownable.unlock (CLE.sol#457-462) should be declared external
CLEToken.totalSupply (CLE.sol#754-756) should be declared external
IERC20.totalSupply (CLE.sol#48) should be declared external
IERC20.transfer (CLE.sol#62) should be declared external
CLEToken.transfer (CLE.sol#763-766) should be declared external
CLEToken.allowance (CLE.sol#768-770) should be declared external
IERC20.allowance (CLE.sol#71) should be declared external
CLEToken.approve (CLE.sol#772-775) should be declared external
IERC20.approve (CLE.sol#87) should be declared external
CLEToken.transferFrom (CLE.sol#777-781) should be declared external
IERC20.transferFrom (CLE.sol#98) should be declared external
CLEToken.name (CLE.sol#742-744) should be declared external
CLEToken.symbol (CLE.sol#746-748) should be declared external
CLEToken.decimals (CLE.sol#750-752) should be declared external
CLEToken.increaseAllowance (CLE.sol#783-786) should be declared external
CLEToken.decreaseAllowance (CLE.sol#788-791) should be declared external
CLEToken.isExcludedFromReward (CLE.sol#793-795) should be declared external
CLEToken.totalFees (CLE.sol#797-799) should be declared external
CLEToken.deliver (CLE.sol#801-808) should be declared external
CLEToken.reflectionFromToken (CLE.sol#810-819) should be declared external
CLEToken.excludeFromReward (CLE.sol#827-835) should be declared external
CLEToken.excludeFromFee (CLE.sol#862-864) should be declared external
CLEToken.includeInFee (CLE.sol#866-868) should be declared external
CLEToken.setSwapAndLiquifyEnabled (CLE.sol#898-901) should be declared external
CLEToken.isExcludedFromFee (CLE.sol#1003-1005) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
```

```
Reentrancy in CLEToken._transfer (CLE.sol#1015-1059):
        External calls:
        - swapAndLiquify(contractTokenBalance) (CLE.sol#1046)
        State variables written after the call(s):
        - _communityFee (CLE.sol#1058)
        - _liquidityFee (CLE.sol#1058)
        - _previousCommunityFee (CLE.sol#1058)
        - _previousLiquidityFee (CLE.sol#1058)
        - _previousTaxFee (CLE.sol#1058)
        - _tFeeTotal (CLE.sol#1058)
        - _taxFee (CLE.sol#1058)
Reentrancy in CLEToken.transferFrom (CLE.sol#777-781):
        External calls:
        - _transfer(sender,recipient,amount) (CLE.sol#778)
        State variables written after the call(s):
        - _allowances (CLE.sol#779)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Address.isContract uses assembly (CLE.sol#307-316)
        - CLE.sol#314
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#assembly-usage
INFO:Detectors:
CLEToken._decimals should be constant (CLE.sol#696)
CLEToken._name should be constant (CLE.sol#694)
CLEToken._symbol should be constant (CLE.sol#695)
CLEToken._tTotal should be constant (CLE.sol#690)
CLEToken.numTokensSellToAddToLiquidity should be constant (CLE.sol#713)
CLEToken.uniswapV2Pair should be constant (CLE.sol#707)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
Ownable.renounceOwnership (CLE.sol#429-432) should be declared external
Ownable.transferOwnership (CLE.sol#438-442) should be declared external
Ownable.geUnlockTime (CLE.sol#444-446) should be declared external
Ownable.lock (CLE.sol#449-454) should be declared external
Ownable.unlock (CLE.sol#457-462) should be declared external
CLEToken.totalSupply (CLE.sol#754-756) should be declared external
IERC20.totalSupply (CLE.sol#48) should be declared external
IERC20.transfer (CLE.sol#62) should be declared external
```

```
INFO:Detectors:
CLEToken.uniswapV2Router (CLE.sol#706) is never initialized. It is used in:
        - swapTokensForEth (CLE.sol#1082-1098)
CLEToken.uniswapV2Pair (CLE.sol#707) is never initialized. It is used in:
        - _transfer (CLE.sol#1015-1059)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#uninitialized-state-variables
INFO:Detectors:
Address.isContract (CLE.sol#307-316) is declared view but contains assembly code
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#constant-functions-changing-the-stat
INFO:Detectors:
Reentrancy in CLEToken._transfer (CLE.sol#1015-1059):
        External calls:
        - swapAndLiquify(contractTokenBalance) (CLE.sol#1046)
        State variables written after the call(s):
        - _rOwned (CLE.sol#1058)
        - _rTotal (CLE.sol#1058)
        - _tOwned (CLE.sol#1058)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
CLEToken.includeInReward (CLE.sol#837-848) does not use the value returned by external calls:
        -_excluded.pop() (CLE.sol#844)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#unused-return
INFO:Detectors:
CLEToken.allowance.owner (local variable @ CLE.sol#768) shadows:
        - Ownable.owner (function @ CLE.sol#410-412)
CLEToken._approve.owner (local variable @ CLE.sol#1007) shadows:
        - Ownable.owner (function @ CLE.sol#410-412)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#local-variable-shadowing
INFO:Detectors:
CLEToken.includeInReward has external calls inside a loop:
        - _excluded.pop() (CLE.sol#844)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation/_edit#calls-inside-a-loop
INFO:Detectors:
```

Implementation Recommendations

Function 'Context._msgSender' (CLE.sol#275-277) is not in mixedCase

Function 'Context._msgData' (CLE.sol#279-282) is not in mixedCase

Parameter '' of IUniswapV2Factory.allPairs (CLE.sol#474) is not in mixedCase

Parameter '' of IUniswapV2Factory.setFeeTo (CLE.sol#479) is not in mixedCase

Parameter '' of IUniswapV2Factory.setFeeToSetter (CLE.sol#480) is not in mixedCase

Function 'IUniswapV2Pair.DOMAIN_SEPARATOR' (CLE.sol#500) is not in mixedCase

Function 'IUniswapV2Pair.PERMIT_TYPEHASH' (CLE.sol#501) is not in mixedCase

Function 'IUniswapV2Pair.MINIMUM_LIQUIDITY' (CLE.sol#518) is not in mixedCase

Parameter '' of IUniswapV2Pair.initialize (CLE.sol#533) is not in mixedCase

Parameter '_scope_0' of IUniswapV2Pair.initialize (CLE.sol#533) is not in mixedCase

Function 'IUniswapV2Router01.WETH' (CLE.sol#539) is not in mixedCase

Function 'CLEToken._transferBothExcluded' (CLE.sol#850-860) is not in mixedCase

Parameter '_enabled' of CLEToken.setSwapAndLiquifyEnabled (CLE.sol#898) is not in mixedCase

Function 'CLEToken._reflectFee' (CLE.sol#906-909) is not in mixedCase

Function 'CLEToken._getValues' (CLE.sol#911-915) is not in mixedCase

Function 'CLEToken._getTValues' (CLE.sol#917-923) is not in mixedCase

Function 'CLEToken._getRValues' (CLE.sol#925-932) is not in mixedCase

Function 'CLEToken._getRate' (CLE.sol#934-937) is not in mixedCase

Function 'CLEToken._getCurrentSupply' (CLE.sol#939-949) is not in mixedCase

Function 'CLEToken._takeLiquidity' (CLE.sol#951-957) is not in mixedCase

Function 'CLEToken._takeCommunity' (CLE.sol#959-965) is not in mixedCase

Parameter '_amount' of CLEToken.calculateTaxFee (CLE.sol#967) is not in mixedCase

Parameter '_amount' of CLEToken.calculateCommunityFee (CLE.sol#973) is not in mixedCase

Parameter '_amount' of CLEToken.calculateLiquidityFee (CLE.sol#979) is not in mixedCase

Function 'CLEToken._approve' (CLE.sol#1007-1013) is not in mixedCase

Function 'CLEToken._transfer' (CLE.sol#1015-1059) is not in mixedCase

Function 'CLEToken._tokenTransfer' (CLE.sol#1102-1120) is not in mixedCase

Function 'CLEToken._transferStandard' (CLE.sol#1122-1130) is not in mixedCase

Function 'CLEToken._transferToExcluded' (CLE.sol#1132-1141) is not in mixedCase

Function 'CLEToken._transferFromExcluded' (CLE.sol#1143-1152) is not in mixedCase

Variable 'CLEToken._taxFee' (CLE.sol#698) is not in mixedCase

Variable 'CLEToken._communityFee' (CLE.sol#701) is not in mixedCase

Variable 'CLEToken._liquidityFee' (CLE.sol#703) is not in mixedCase

Variable 'CLEToken._maxTxAmount' (CLE.sol#712) is not in mixedCase

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the CHAMPIONS LEAGUE contract. Securing smart contracts is a multistep process; therefore, running a bug bounty program as a complement to this audit is strongly recommended.

Summary

The use case of the smart contract is very well designed and Implemented. Overall, the code is well written and demonstrates effective use of abstraction, separation of concerns, and modularity. The CHAMPIONS LEAGUE development team demonstrated high technical capabilities, both in the design of the architecture and in the implementation.

Some low-severity issues have been reported and documented above; we recommend that the CHAMPIONS LEAGUE team fix them.

QuillAudits

Canada, India, Singapore and United Kingdom

audits.quillhash.com

audits@quillhash.com