QuillAudits

Audit Report
March, 2022

For

BollyCoin

Contents

Scope of the Audit

The scope of this audit was to analyze and document the BollyStake smart contract codebase for quality, security, and correctness.

Checked Vulnerabilities

We have scanned the smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis
In this step, we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis
Static analysis of smart contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis
Manual analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption
In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit
Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis, Theo.

Issue Categories

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

Risk-level	Description
High	A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.
Medium	The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.
Low	Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.
Informational	These are severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Number of issues per severity

Type	High	Medium	Low	Informational
Open	0	0	0	0
Acknowledged	0	0	0	0
Closed	2	1	1	3

Introduction

During the period of **Mar 15, 2022 to Mar 25 , 2022** - QuillAudits Team performed a security audit for BollyStake smart contracts.

The code for the audit was taken from following the official link:

V	Date	Commit ID/URL
1	15th March	https://rinkeby.etherscan.io/address/0xFD2522DfE4DeA12909E23d7C46B56502d575d392
2	24th March	https://rinkeby.etherscan.io/address/0x93306156e03ce18cac7500e3Dc889f7D1Abaf4bf#code
3	25th March	https://rinkeby.etherscan.io/address/0x9Dc4C94e501f91146697281Ae00431c688515D5c#code

Issues Found – Code Review / Manual Testing

A. Contract - staking.sol

High severity issues

1. Users will lose their token if calling a wrong “days” in relock_stake()

Description

User A calls enter_stake(XXX, YYY), where XXX can be any amount of the token, and YYY is in range of [30,60,90,180,365,540,730]. For example: YYY can be 30 in this case.

After 3 mins, user A got some XXX expired amount and user A wants to relock_stake(YYY). As stated in the code, relock_stake() function does not sanitize the days parameter, therefore, a user can put any numbers of days to the function, YYY can be any positive number. For instance, YYY can be 10 in this case.

Unfortunately, the operation in line 1021 and 1022.

```
if (user_stake_info[s].expire < block.timestamp) {
    user_stake_info[s].locked_amount = 0;
}
```

turn his locked_amount to 0. As a result, user A cannot get those XXX back to his wallet, which also means that user A has no expired_stakes anymore and that XXX amount is added to the total_eligible_stakes forever.

Remediation

We recommend sanitizing the “days” parameter as it should be in range of [30,60,90,180,365,540,730], and adding a check for expired_amount if expired_amount > 0

Status: Fixed in version 02

2. Users will lose their tokens if it’s transferred to another user.

Description
Users will receive an equivalent amount of Bolly Stake token after staking their Bolly coins. Users can send their Bolly Stake tokens to other users on the network. Unfortunately, we've discovered that if users attempt or inadvertently transfer their Bolly Stake token to other users, their Bolly Stake token will be lost forever because the contract is not designed for transferring as described by the Auditee.

Remediation
The Auditee should prevent token transfers between users and warn stakeholders of this after they have received their Bolly Stake token.

Status: Fixed in version 02

Medium severity issues

3. Uncheck transfer

Description
The return value of an external transfer/transferFrom call is not checked, as follows:
L1011: BOLLY.transferFrom(msg.sender, address(this), _amount);
L1064: BOLLY.transfer(msg.sender, expired_amount);
L1078: BOLLY.transfer(_user, locked_amount);

Remediation
Please consider adding the require() check for those external calls or using SafeERC20, or ensure that the transfer/transferFrom return value is checked.

Status: Fixed in version 02

Low severity issues

4. Comparison to boolean constants

L1017: require(_isStakeholder==true, "only current stakeholders can relock stake");

Description
Boolean constants of the _isStakeholder can be used directly and do not need to be compared to true or false.

Remediation
We recommend removing the equality to the boolean constant.

Status: Fixed in version 02

Informational Issues

5. Conformance to Solidity naming conventions

Description
Functions other than constructors should use mixedCase. Examples: getBalance, transfer, verifyOwner, addMember, changeOwner. We've highlighted some instances as follows:

- emergency_remove_stake()
- remove_stake()
- relock_stake()
- enter_stake()
- expired_stakes()
- get_all_user_stakes()
- Eligible_stakes()
- total_eligible_Stakes()

Events should be named using the CapWords style. Examples: Deposit, Transfer, Approval, BeforeTransfer, AfterTransfer.

```
event _set_owner(address _owner, address _from) ;
  event _enter_stake(uint256 _amount, address _stake_from, uint256 _days);
  event _relock_stake(address stake_from,uint256 _days, uint256 _expired_amount);
  event _remove_stake(uint256 _amount,address _stake_to);
```

Remediation
The Auditee should follow the Solidity naming convention.

Status: Fixed in version 02

6. Unused variable

The Auditee should remove the “mapping(address => locked) users;” at line 845 if it’s not used in the contract.

Status: Fixed in version 02

7. State Variable Default Visibility

```
mapping(address => locked) users;
mapping(address => locked[7]) user_stake;
```

Description
The Visibility of the users and users_stake variable mentioned above is not defined. Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

The default is internal for state variables, but it should be made explicit.

Status: Fixed in version 03

Functional Tests

Testing ERC20 functions

- Should get correct name
- Should get correct balance of admin
- admin should approve and enter_stake (66ms)

Testing all transfer functions

- should get balance of the tokenHolder
- should revert if user has no balance (46ms)
- admin should transfer to account 1 (44ms)
- should call approval to account 2 (40ms)
- Testing decreaseAllowance() and increaseAllowance() (44ms)
- Testing TransferFrom() (47ms)

Testing all stake functions

- enter_stake() should be in range of [30,60,90,180,365,540,730] (104ms)
- stake1 passes enter_stake() (48ms)
- isStakeHolder() should return False for account1
- isStakeHolder() should return True for staker1
- stakeOf() should return correct amount of staker1
- expired_stakes() should return 0 if user_stake_info[s].expire > block.timestamp
- expired_stakes() should return correct if user_stake_info[s].expire < block.timestamp
- remove_stake() should return if expired_amount<0
- remove_stake() should return True for stake1 if expired_amount > 0 (41ms)
- relock_stake() should revert if caller is not stake holder
- [FIXED] relock_stake() should revert if days is NOT in range (93ms)
- staker1 calls enter_stake() with multiple days (69ms)
- total_eligible_Stakes() should return with correct amount
- total_eligible_Stakes() should be reduced if 30 days have passed
- total_eligible_Stakes() should be decreased if 60 days have passed
- total_eligible_Stakes() should be decreased if 90 days have passed
- expired_stakes() should be increased if 30 days have passed
- expired_stakes() should be increased if 60 days have passed
- expired_stakes() should be increased if 30 days have passed
- remove_stake() after 30 days have passed (54ms)
- remove_stake() after 60 days have passed (43ms)
- remove_stake() after 90 days have passed (45ms)

- staker1 calls enter_stake() with multiple days again (71ms)
- emergency_remove_stake() should be called only by owner
- emergency_remove_stake() should return true if all passed

Automated Tests

Slither

```
INFO:Detectors:
Bollystake.enter_stake(uint256,uint256) (Bollystake.sol#982-1014) ignores return value by BOLLY.transferFrom(msg.sender,address(this),_amount) (Bollystake.sol#10
11)
Bollystake.remove_stake() (Bollystake.sol#1052-1066) ignores return value by BOLLY.transfer(msg.sender,expired_amount) (Bollystake.sol#1064)
Bollystake.emergency_remove_stake(address) (Bollystake.sol#1068-1080) ignores return value by BOLLY.transfer(_user,locked_amount) (Bollystake.sol#1078)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
Reentrancy in Bollystake.enter_stake(uint256,uint256) (Bollystake.sol#982-1014):
        External calls:
        - BOLLY.transferFrom(msg.sender,address(this),_amount) (Bollystake.sol#1011)
        State variables written after the call(s):
        - _mint(msg.sender,_amount) (Bollystake.sol#1012)
                - _balances[account] += amount (Bollystake.sol#713)
        - _mint(msg.sender,_amount) (Bollystake.sol#1012)
                - _totalSupply += amount (Bollystake.sol#712)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in Bollystake.emergency_remove_stake(address) (Bollystake.sol#1068-1080):
        External calls:
        - BOLLY.transfer(_user,locked_amount) (Bollystake.sol#1078)
        Event emitted after the call(s):
        - _remove_stake(locked_amount,_user) (Bollystake.sol#1079)
Reentrancy in Bollystake.enter_stake(uint256,uint256) (Bollystake.sol#982-1014):
        External calls:
        - BOLLY.transferFrom(msg.sender,address(this),_amount) (Bollystake.sol#1011)
        Event emitted after the call(s):
        - Transfer(address(0),account,amount) (Bollystake.sol#714)
                - _mint(msg.sender,_amount) (Bollystake.sol#1012)
        - _enter_stake(_amount,msg.sender,_days) (Bollystake.sol#1013)
Reentrancy in Bollystake.remove_stake() (Bollystake.sol#1052-1066):
        External calls:
        - BOLLY.transfer(msg.sender,expired_amount) (Bollystake.sol#1064)
        Event emitted after the call(s):
        - _remove_stake(expired_amount,msg.sender) (Bollystake.sol#1065)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
INFO:Detectors:
Bollystake.get_all_user_stakes() (Bollystake.sol#950-967) uses timestamp for comparisons
        Dangerous comparisons:
        - user_stake[stakeholders[s]][k].expire > block.timestamp (Bollystake.sol#959)
Bollystake.expired_stakes(address) (Bollystake.sol#969-979) uses timestamp for comparisons
        Dangerous comparisons:
        - user_stake_info[s].expire < block.timestamp (Bollystake.sol#974)
Bollystake.relock_stake(uint256) (Bollystake.sol#1015-1048) uses timestamp for comparisons
        Dangerous comparisons:
        - user_stake_info[s].expire < block.timestamp (Bollystake.sol#1021)
Bollystake.remove_stake() (Bollystake.sol#1052-1066) uses timestamp for comparisons
        Dangerous comparisons:
        - user_stake_info[s].expire < block.timestamp (Bollystake.sol#1058)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Bollystake.relock_stake(uint256) (Bollystake.sol#1015-1048) compares to a boolean constant:
        -require(bool,string)(_isStakeholder == true,only current stakeholders can relock stake) (Bollystake.sol#1017)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
INFO:Detectors:
Different versions of Solidity is used:
        - Version used: ['^0.8.0', '^0.8.7']
        - ^0.8.0 (Bollystake.sol#10)
        - ^0.8.0 (Bollystake.sol#240)
        - ^0.8.0 (Bollystake.sol#267)
        - ^0.8.0 (Bollystake.sol#345)
        - ^0.8.0 (Bollystake.sol#430)
        - ^0.8.0 (Bollystake.sol#460)
        - ^0.8.7 (Bollystake.sol#817)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Context._msgData() (Bollystake.sol#257-259) is never used and should be removed
SafeMath.div(uint256,uint256) (Bollystake.sol#141-143) is never used and should be removed
SafeMath.div(uint256,uint256,string) (Bollystake.sol#197-206) is never used and should be removed
SafeMath.mod(uint256,uint256) (Bollystake.sol#157-159) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (Bollystake.sol#223-232) is never used and should be removed
SafeMath.mul(uint256,uint256) (Bollystake.sol#127-129) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (Bollystake.sol#174-183) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (Bollystake.sol#28-34) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (Bollystake.sol#70-75) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (Bollystake.sol#82-87) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (Bollystake.sol#53-63) is never used and should be removed
SafeMath.trySub(uint256,uint256) (Bollystake.sol#41-46) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
INFO:Detectors:
Pragma version^0.8.0 (Bollystake.sol#10) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (Bollystake.sol#240) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (Bollystake.sol#267) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (Bollystake.sol#345) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (Bollystake.sol#430) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (Bollystake.sol#460) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.7 (Bollystake.sol#817) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.7 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Struct Bollystake.locked (Bollystake.sol#841-844) is not in CapWords
Event Bollystake_set_owner(address,address) (Bollystake.sol#1083) is not in CapWords
Event Bollystake_enter_stake(uint256,address,uint256) (Bollystake.sol#1084) is not in CapWords
Event Bollystake_relock_stake(address,uint256,uint256) (Bollystake.sol#1085) is not in CapWords
Event Bollystake_remove_stake(uint256,address) (Bollystake.sol#1086) is not in CapWords
Parameter Bollystake.isStakeholder(address)._address (Bollystake.sol#854) is not in mixedCase
Parameter Bollystake.addStakeholder(address)._stakeholder (Bollystake.sol#869) is not in mixedCase
Parameter Bollystake.removeStakeholder(address)._stakeholder (Bollystake.sol#880) is not in mixedCase
Parameter Bollystake.stakeOf(address)._stakeholder (Bollystake.sol#897) is not in mixedCase
Function Bollystake.total_eligible_Stakes() (Bollystake.sol#921-935) is not in mixedCase
Function Bollystake.Eligible_stakes() (Bollystake.sol#937-948) is not in mixedCase
Function Bollystake.get_all_user_stakes() (Bollystake.sol#950-967) is not in mixedCase
Function Bollystake.expired_stakes(address) (Bollystake.sol#969-979) is not in mixedCase
Parameter Bollystake.expired_stakes(address)._stakeholder (Bollystake.sol#969) is not in mixedCase
Function Bollystake.enter_stake(uint256,uint256) (Bollystake.sol#982-1014) is not in mixedCase
Parameter Bollystake.enter_stake(uint256,uint256)._amount (Bollystake.sol#982) is not in mixedCase
Parameter Bollystake.enter_stake(uint256,uint256)._days (Bollystake.sol#982) is not in mixedCase
Function Bollystake.relock_stake(uint256) (Bollystake.sol#1015-1048) is not in mixedCase
Parameter Bollystake.relock_stake(uint256)._days (Bollystake.sol#1015) is not in mixedCase
Function Bollystake.remove_stake() (Bollystake.sol#1052-1066) is not in mixedCase
Function Bollystake.emergency_remove_stake(address) (Bollystake.sol#1068-1080) is not in mixedCase
Parameter Bollystake.emergency_remove_stake(address)._user (Bollystake.sol#1068) is not in mixedCase
Variable Bollystake.BOLLY (Bollystake.sol#826) is not in mixedCase
Variable Bollystake.user_stake (Bollystake.sol#846) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable Bollystake._TIMELOCK_30 (Bollystake.sol#832) is too similar to Bollystake._TIMELOCK_60 (Bollystake.sol#833)
Variable Bollystake._TIMELOCK_30 (Bollystake.sol#832) is too similar to Bollystake._TIMELOCK_90 (Bollystake.sol#834)
Variable Bollystake._TIMELOCK_60 (Bollystake.sol#833) is too similar to Bollystake._TIMELOCK_90 (Bollystake.sol#834)
Variable Bollystake.stakeOf(address)._stakeholder (Bollystake.sol#897) is too similar to Bollystake.stakeholders (Bollystake.sol#847)
Variable Bollystake.expired_stakes(address)._stakeholder (Bollystake.sol#969) is too similar to Bollystake.stakeholders (Bollystake.sol#847)
Variable Bollystake.addStakeholder(address)._stakeholder (Bollystake.sol#869) is too similar to Bollystake.stakeholders (Bollystake.sol#847)
Variable Bollystake.removeStakeholder(address)._stakeholder (Bollystake.sol#880) is too similar to Bollystake.stakeholders (Bollystake.sol
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
Bollystake.users (Bollystake.sol#845) is never used in Bollystake (Bollystake.sol#823-1088)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (Bollystake.sol#316-318)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (Bollystake.sol#324-327)
name() should be declared external:
        - ERC20.name() (Bollystake.sol#517-519)
symbol() should be declared external:
        - ERC20.symbol() (Bollystake.sol#525-527)
decimals() should be declared external:
        - ERC20.decimals() (Bollystake.sol#542-544)
totalSupply() should be declared external:
        - ERC20.totalSupply() (Bollystake.sol#549-551)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (Bollystake.sol#556-558)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (Bollystake.sol#568-571)
allowance(address,address) should be declared external:
        - ERC20.allowance(address,address) (Bollystake.sol#576-578)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (Bollystake.sol#587-590)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (Bollystake.sol#605-619)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (Bollystake.sol#633-636)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (Bollystake.sol#652-660)
total_eligible_Stakes() should be declared external:
        - Bollystake.total_eligible_Stakes() (Bollystake.sol#921-935)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

```
INFO:Detectors:
Pragma version>=0.6.0<0.8.0 (@openzeppelin/contracts/access/Ownable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (@openzeppelin/contracts/math/SafeMath.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (@openzeppelin/contracts/token/ERC20/IERC20.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (@openzeppelin/contracts/token/ERC20/SafeERC20.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (@openzeppelin/contracts/utils/ReentrancyGuard.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (@openzeppelin/contracts/utils/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (@openzeppelin/contracts/utils/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (@openzeppelin/contracts/utils/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (@openzeppelin/contracts/utils/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (@openzeppelin/contracts/utils/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (@openzeppelin/contracts/utils/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (@openzeppelin/contracts/utils/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (@openzeppelin/contracts/utils/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Event StakingVaultunStaked(address,uint256,uint256) (StakingVault.sol#43) is not in CapWords
Event StakingVaultrewardPercentChanged(uint256) (StakingVault.sol#44) is not in CapWords
Event StakingVaultrewardIntervalChanged(uint256) (StakingVault.sol#45) is not in CapWords
Parameter StakingVault.getuserStakedAmount(address)._user (StakingVault.sol#63) is not in mixedCase
Parameter StakingVault.changeRewardPercent(uint256)._rewardPercent (StakingVault.sol#76) is not in mixedCase
Parameter StakingVault.changeLockDuration(uint256)._rewardInterval (StakingVault.sol#87) is not in mixedCase
Parameter StakingVault.addReward(uint256)._rewardAmount (StakingVault.sol#105) is not in mixedCase
Parameter StakingVault.lockStakingTokenToParticipate(uint256)._stakeAmount (StakingVault.sol#119) is not in mixedCase
Parameter StakingVault.withdrawRewardToken(uint256)._amount (StakingVault.sol#188) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (@openzeppelin/contracts/utils/Context.sol#21)" inContext (@openzeppelin/contracts/utils/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
```

```
INFO:Detectors:
Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.11', '>=0.6.2<0.8.0']
        - >=0.6.0<0.8.0 (@openzeppelin/contracts/access/Ownable.sol#3)
        - >=0.6.0<0.8.0 (@openzeppelin/contracts/math/SafeMath.sol#3)
        - >=0.6.0<0.8.0 (@openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - >=0.6.0<0.8.0 (@openzeppelin/contracts/token/ERC20/SafeERC20.sol#3)
        - >=0.6.2<0.8.0 (@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.0<0.8.0 (@openzeppelin/contracts/utils/ReentrancyGuard.sol#3)
        - >=0.6.11 (StakingVault.sol#3)
        - ABIEncoderV2 (StakingVault.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Address.functionCall(address,bytes) (@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (@openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
Context._msgData() (@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (@openzeppelin/contracts/token/ERC20/SafeERC20.sol#37-46) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (@openzeppelin/contracts/token/ERC20/SafeERC20.sol#53-56) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (@openzeppelin/contracts/token/ERC20/SafeERC20.sol#48-51) is never used and should be removed
SafeMath.div(uint256,uint256,string) (@openzeppelin/contracts/math/SafeMath.sol#190-193) is never used and should be removed
SafeMath.mod(uint256,uint256) (@openzeppelin/contracts/math/SafeMath.sol#152-155) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (@openzeppelin/contracts/math/SafeMath.sol#210-213) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (@openzeppelin/contracts/math/SafeMath.sol#170-173) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (@openzeppelin/contracts/math/SafeMath.sol#24-28) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (@openzeppelin/contracts/math/SafeMath.sol#60-63) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (@openzeppelin/contracts/math/SafeMath.sol#70-73) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (@openzeppelin/contracts/math/SafeMath.sol#45-53) is never used and should be removed
SafeMath.trySub(uint256,uint256) (@openzeppelin/contracts/math/SafeMath.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
INFO:Detectors:
Reentrancy in StakingVault.emergencyUnstake() (StakingVault.sol#174-186):
        External calls:
        - stakingVaultToken.safeTransfer(msg.sender,_amountLocked) (StakingVault.sol#180)
        Event emitted after the call(s):
        - unStaked(msg.sender,_amountLocked,0) (StakingVault.sol#184)
Reentrancy in StakingVault.lockStakingTokenToParticipate(uint256) (StakingVault.sol#119-149):
        External calls:
        - stakingVaultToken.safeTransferFrom(msg.sender,address(this),_stakeAmount) (StakingVault.sol#135-139)
        Event emitted after the call(s):
        - Staked(msg.sender,_stakeAmount) (StakingVault.sol#147)
Reentrancy in StakingVault.unLockStakingToken() (StakingVault.sol#151-172):
        External calls:
        - stakingVaultToken.safeTransfer(msg.sender,_amountUnlocked) (StakingVault.sol#168)
        - rewardVaultToken.safeTransfer(msg.sender,_rewardsUnlocked) (StakingVault.sol#169)
        Event emitted after the call(s):
        - unStaked(msg.sender,_amountUnlocked,_rewardsUnlocked) (StakingVault.sol#170)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
StakingVault.unLockStakingToken() (StakingVault.sol#151-172) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= user.userUnLockTime,lock time is not over yet) (StakingVault.sol#154-157)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Address.isContract(address) (@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (@openzeppelin/contracts/utils/Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (@openzeppelin/contracts/access/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (@openzeppelin/contracts/access/Ownable.sol#63-67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

Solhint Linter

```
Bollystake.sol
   10:1   error    Compiler version ^0.8.0 does not satisfy the ^0.5.8 semver requirement                                compiler-version
  240:1   error    Compiler version ^0.8.0 does not satisfy the ^0.5.8 semver requirement                                compiler-version
  267:1   error    Compiler version ^0.8.0 does not satisfy the ^0.5.8 semver requirement                                compiler-version
  290:5   warning  Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)  func-visibility
  325:9   warning  Error message for require is too long                                                                 reason-string
  345:1   error    Compiler version ^0.8.0 does not satisfy the ^0.5.8 semver requirement                                compiler-version
  430:1   error    Compiler version ^0.8.0 does not satisfy the ^0.5.8 semver requirement                                compiler-version
  460:1   error    Compiler version ^0.8.0 does not satisfy the ^0.5.8 semver requirement                                compiler-version
  509:5   warning  Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)  func-visibility
  613:9   warning  Error message for require is too long                                                                 reason-string
  654:9   warning  Error message for require is too long                                                                 reason-string
  681:9   warning  Error message for require is too long                                                                 reason-string
  682:9   warning  Error message for require is too long                                                                 reason-string
  687:9   warning  Error message for require is too long                                                                 reason-string
  731:9   warning  Error message for require is too long                                                                 reason-string
  736:9   warning  Error message for require is too long                                                                 reason-string
  765:9   warning  Error message for require is too long                                                                 reason-string
  766:9   warning  Error message for require is too long                                                                 reason-string
  790:24  warning  Code contains empty blocks                                                                            no-empty-blocks
  810:24  warning  Code contains empty blocks                                                                            no-empty-blocks
  817:1   error    Compiler version ^0.8.7 does not satisfy the ^0.5.8 semver requirement                                compiler-version
  826:5   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  828:5   warning  Explicitly mark visibility in function (Set ignoreConstructors to true if using solidity >=0.7.0)  func-visibility
  828:17  warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  841:6   warning  Contract name must be in CamelCase                                                                    contract-name-camelcas
  843:9   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  845:6   warning  Explicitly mark visibility of state                                                                   state-visibility
  846:6   warning  Explicitly mark visibility of state                                                                   state-visibility
  846:6   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  921:4   warning  Function name must be in mixedCase                                                                    func-name-mixedcase
  928:8   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  937:4   warning  Function name must be in mixedCase                                                                    func-name-mixedcase
  939:8   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  940:8   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  941:8   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  950:5   warning  Function name must be in mixedCase                                                                    func-name-mixedcase
  952:8   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  953:8   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  954:8   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  959:59  warning  Avoid to make time-based decisions in your business logic                                             not-rely-on-time
  969:4   warning  Function name must be in mixedCase                                                                    func-name-mixedcase
  971:8   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  972:9   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  974:46  warning  Avoid to make time-based decisions in your business logic                                             not-rely-on-time
  982:5   warning  Function name must be in mixedCase                                                                    func-name-mixedcase
  984:9   warning  Error message for require is too long                                                                 reason-string
  988:9   warning  Variable name must be in mixedCase                                                                    var-name-mixedcase
  990:37  warning  Avoid to make time-based decisions in your business logic                                             not-rely-on-time
```

```
 993:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
 996:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
 999:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1002:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1005:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1008:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1015:5   warning  Function name must be in mixedCase                        func-name-mixedcase
1017:9   warning  Error message for require is too long                     reason-string
1018:9   warning  Variable name must be in mixedCase                        var-name-mixedcase
1019:9   warning  Variable name must be in mixedCase                        var-name-mixedcase
1021:46  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1026:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1029:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1032:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1035:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1038:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1041:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1044:37  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1052:5   warning  Function name must be in mixedCase                        func-name-mixedcase
1053:9   warning  Variable name must be in mixedCase                        var-name-mixedcase
1054:9   warning  Variable name must be in mixedCase                        var-name-mixedcase
1055:9   warning  Error message for require is too long                     reason-string
1058:46  warning  Avoid to make time-based decisions in your business logic  not-rely-on-time
1068:5   warning  Function name must be in mixedCase                        func-name-mixedcase
1069:9   warning  Variable name must be in mixedCase                        var-name-mixedcase
1070:9   warning  Variable name must be in mixedCase                        var-name-mixedcase
1071:9   warning  Error message for require is too long                     reason-string
1083:5   warning  Event name must be in CamelCase                           event-name-camelcase
1084:5   warning  Event name must be in CamelCase                           event-name-camelcase
1084:41  warning  Variable name must be in mixedCase                        var-name-mixedcase
1085:5   warning  Event name must be in CamelCase                           event-name-camelcase
1085:25  warning  Variable name must be in mixedCase                        var-name-mixedcase
1085:59  warning  Variable name must be in mixedCase                        var-name-mixedcase
1086:5   warning  Event name must be in CamelCase                           event-name-camelcase
1086:41  warning  Variable name must be in mixedCase                        var-name-mixedcase
```

Results

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Closing Summary

In this report, we have considered the security of the BollyStake platform. We performed our audit according to the procedure described above.

The audit showed several high, medium, low, and informational severity issues. There have been no major or critical issues related to the codebase and all findings listed here were fixed by the Auditee.

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the BollyStake platform. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the BollyStake Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

Audit Report
March, 2022

For

BollyCoin

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com