QuillAudits

Audit Report
August, 2021

BitBook

Contents

Introduction

This Audit Report highlights the overall security of the BitBook 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: BitBook
Staking Contract: 0x8cB3e609Bcff4d777dDA5189E310c9f9e49aB132
Updated Staking: 0x046564e17dD76df34fF06200527b58B9173b4fdE
Languages: Solidity (Smart contract), Javascript (Unit Testing)
Platforms and Tools: Remix IDE, Truffle, Truffle Team, Ganache, Slither, Surya

Summary of the Smart Contract

QuillAudits conducted a security audit of a smart contract of BitBook. BitBook contracts are used for token and Staking contracts.

- Staking

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	0	0	0	0
Closed	2	0	1	0

Manual Audit

High level severity issues

1. Reentrancy Vulnerability (Staking.sol)

Reentrancy in BitBookStake.withdraw (Staking.sol#224-236):
External calls:
- safeTransfer(msg.sender,amount) (Staking.sol#232)
State variables written after the call(s):
- userDetails (Staking.sol#233)

Do not update a state variable after a transfer call.

1.1. **Reentrancy Vulnerability (Staking.sol)**

Reentrancy in BitBookStake.deposit (Staking.sol#169-182):
External calls:
- safeTransferFrom(msg.sender,address(this),_amount)
(Staking.sol#172)
State variables written after the call(s):
- stakeID (Staking.sol#180)
- stakings (Staking.sol#177)
- totalSupply (Staking.sol#181)
- userDetails (Staking.sol#174)
- userDetails (Staking.sol#175)
- userDetails (Staking.sol#176)

Do not update a state variable after a transfer call.

Status: Closed

Medium level severity issues

No medium severity issues

Low level severity issues

1. **Function should be declared External ()**

   Check: external-function
   Severity: Optimization
   Confidence: High

   Ownable.owner (Staking.sol#109-111) should be declared external
   Ownable.transferOwnership (Staking.sol#118-120) should be declared external
   BitBookStake.deposit (Staking.sol#169-182) should be declared external
   BitBookStake.setFeePercentage (Staking.sol#194-199) should be declared external
   BitBookStake.viewFeePercentage (Staking.sol#201-204) should be declared external
   BitBookStake.withdraw (Staking.sol#224-236) should be declared external
   BitBookStake.stakingId (Staking.sol#247-249) should be declared external
   BitBookStake.updateReward (Staking.sol#251-254) should be declared external
   BitBookStake.failsafe (Staking.sol#256-261) should be declared external

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

   Status: Closed

2. **Unused Return (Staking.sol)**

Check: unused-return
Severity: Low
Confidence: Medium

BitBookStake.safeTransferFrom (Staking.sol#) does not use the value returned by external calls:
-bitBookToken.transferFrom(from,to,amount) (Staking.sol#)
BitBookStake.safeTransfer (Staking.sol#) does not use the value returned by external calls:
-bitBookToken.transfer(_to,_amount) (Staking.sol#)

The return value of an external call is not stored in a local or state variable.

Ensure that all the return values of the function calls are used.

Status: Closed

Functional test

Function test has been done for multiple functions of three files. Results are below:

Staking.sol

- **deposit** deposit tokens to stake
  -- > PASS

- **setFeePercentage** set fee by owner only
  --> PASS

- **updateReward** update rewards for users
  --> PASS

- **transferOwnership** transfer contract ownership to another address
  --> PASS

Automated Testing

Slither Tool Result

```
Ownable.owner (Staking.sol#109-111) should be declared external
Ownable.transferOwnership (Staking.sol#118-120) should be declared external
BitBookStake.deposit (Staking.sol#169-182) should be declared external
BitBookStake.setFeePercentage (Staking.sol#194-199) should be declared external
BitBookStake.viewFeePercentage (Staking.sol#201-204) should be declared external
BitBookStake.withdraw (Staking.sol#224-236) should be declared external
BitBookStake.stakingId (Staking.sol#247-249) should be declared external
BitBookStake.updateReward (Staking.sol#251-254) should be declared external
BitBookStake.failsafe (Staking.sol#256-261) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in Staking.sol:
        - pragma solidity0.5.16 (Staking.sol#2): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Function 'Context._msgSender' (Staking.sol#35-37) is not in mixedCase
Function 'Context._msgData' (Staking.sol#39-42) is not in mixedCase
Function 'Ownable._transferOwnership' (Staking.sol#122-126) is not in mixedCase
Struct 'BitBookStake._StakeingId' (Staking.sol#145-147) is not in CapWords
Parameter '_bitBookToken' of BitBookStake. (Staking.sol#159) is not in mixedCase
Parameter '_amount' of BitBookStake.deposit (Staking.sol#169) is not in mixedCase
Parameter '_to' of BitBookStake.safeTransfer (Staking.sol#188) is not in mixedCase
Parameter '_amount' of BitBookStake.safeTransfer (Staking.sol#188) is not in mixedCase
Parameter '_from' of BitBookStake.setFeePercentage (Staking.sol#194) is not in mixedCase
Parameter '_to' of BitBookStake.setFeePercentage (Staking.sol#194) is not in mixedCase
Parameter '_pcent' of BitBookStake.setFeePercentage (Staking.sol#194) is not in mixedCase
Parameter '_from' of BitBookStake.viewFeePercentage (Staking.sol#201) is not in mixedCase
Parameter '_to' of BitBookStake.viewFeePercentage (Staking.sol#201) is not in mixedCase
Parameter '_account' of BitBookStake.calculateFee (Staking.sol#208) is not in mixedCase
Parameter '_stakerID' of BitBookStake.calculateFee (Staking.sol#208) is not in mixedCase
Parameter '_stakerID' of BitBookStake.withdraw (Staking.sol#224) is not in mixedCase
Parameter '_user' of BitBookStake.rewardCalc (Staking.sol#238) is not in mixedCase
Parameter '_id' of BitBookStake.rewardCalc (Staking.sol#238) is not in mixedCase
Parameter '_staker' of BitBookStake.stakingId (Staking.sol#247) is not in mixedCase
Parameter '_daily' of BitBookStake.updateReward (Staking.sol#251) is not in mixedCase
Parameter '_monthly' of BitBookStake.updateReward (Staking.sol#251) is not in mixedCase
Parameter '_to' of BitBookStake.failsafe (Staking.sol#256) is not in mixedCase
Parameter '_amount' of BitBookStake.failsafe (Staking.sol#256) is not in mixedCase
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Reentrancy in BitBookStake.withdraw (Staking.sol#224-236):
        External calls:
        - safeTransfer(msg.sender,amount) (Staking.sol#232)
        State variables written after the call(s):
        - userDetails (Staking.sol#233)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
BitBookStake.safeTransferFrom (Staking.sol#184-186) does not use the value returned by external calls:
        -bitBookToken.transferFrom(from,to,amount) (Staking.sol#185)
BitBookStake.safeTransfer (Staking.sol#188-190) does not use the value returned by external calls:
        -bitBookToken.transfer(_to,_amount) (Staking.sol#189)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#unused-return
INFO:Detectors:
Reentrancy in BitBookStake.deposit (Staking.sol#169-182):
        External calls:
        - safeTransferFrom(msg.sender,address(this),_amount) (Staking.sol#172)
        State variables written after the call(s):
        - stakeID (Staking.sol#180)
        - stakings (Staking.sol#177)
        - totalSupply (Staking.sol#181)
        - userDetails (Staking.sol#174)
        - userDetails (Staking.sol#175)
        - userDetails (Staking.sol#176)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
BitBookStake.calculateFee (Staking.sol#208-222) uses timestamp for comparisons
        Dangerous comparisons:
        - 864000 >= staketime (Staking.sol#213-221)
        - 7776000 >= staketime (Staking.sol#217-221)
        - 259200 >= staketime (Staking.sol#211-221)
        - 2592000 >= staketime (Staking.sol#215-221)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#block-timestamp
INFO:Detectors:
Ownable.owner (Staking.sol#109-111) should be declared external
Ownable.transferOwnership (Staking.sol#118-120) should be declared external
BitBookStake.deposit (Staking.sol#169-182) should be declared external
BitBookStake.setFeePercentage (Staking.sol#194-199) should be declared external
BitBookStake.viewFeePercentage (Staking.sol#201-204) should be declared external
```

```
INFO:Detectors:
BitBook.allowance.owner (local variable @ bitbook.sol#201) shadows:
        - Ownable.owner (function @ bitbook.sol#114-116)
BitBook._approve.owner (local variable @ bitbook.sol#280) shadows:
        - Ownable.owner (function @ bitbook.sol#114-116)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#local-variable-shadowing
INFO:Detectors:
BitBook.mint (bitbook.sol#230-238) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp >= oneMonth (bitbook.sol#232-237)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#block-timestamp
INFO:Detectors:
BitBook._decimals should be constant (bitbook.sol#147)
BitBook._name should be constant (bitbook.sol#149)
BitBook._symbol should be constant (bitbook.sol#148)
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
Ownable.renounceOwnership (bitbook.sol#123-126) should be declared external
Ownable.transferOwnership (bitbook.sol#128-130) should be declared external
BitBook.increaseAllowance (bitbook.sol#219-222) should be declared external
BitBook.decreaseAllowance (bitbook.sol#225-228) should be declared external
BitBook.burn (bitbook.sol#240-243) should be declared external
BitBook.updateTresauryAddress (bitbook.sol#245-248) should be declared external
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#public-function-that-could-be-declared-as-external
INFO:Detectors:
Detected issues with version pragma in bitbook.sol:
        - pragma solidity0.5.16 (bitbook.sol#5): it allows old versions
Reference: https://github.com/trailofbits/slither/wiki/Detectors-Documentation#incorrect-version-of-solidity
INFO:Detectors:
Function 'Context._msgSender' (bitbook.sol#38-40) is not in mixedCase
Function 'Context._msgData' (bitbook.sol#42-45) is not in mixedCase
Function 'Ownable._transferOwnership' (bitbook.sol#132-136) is not in mixedCase
Parameter '_tresauryAddr' of BitBook. (bitbook.sol#154) is not in mixedCase
Parameter '_tresauryAddr' of BitBook.updateTresauryAddress (bitbook.sol#245) is not in mixedCase
Function 'BitBook._transfer' (bitbook.sol#251-258) is not in mixedCase
Function 'BitBook._mint' (bitbook.sol#260-268) is not in mixedCase
Function 'BitBook._burn' (bitbook.sol#271-277) is not in mixedCase
Function 'BitBook._approve' (bitbook.sol#280-286) is not in mixedCase
```

Results

Some false positive errors have been reported by the tool; all other errors have been covered in issues explained above, under low-level severity issues.

Implementation Recommendations

Function 'Context._msgSender' (bitbook.sol#38-40) is not in mixedCase

Function 'Context._msgData' (bitbook.sol#42-45) is not in mixedCase

Function 'Ownable._transferOwnership' (bitbook.sol#132-136) is not in mixedCase

Parameter '_tresauryAddr' of BitBook. (bitbook.sol#154) is not in mixedCase

Parameter '_tresauryAddr' of BitBook.updateTresauryAddress (bitbook.sol#245) is not in mixedCase

Function 'BitBook._transfer' (bitbook.sol#251-258) is not in mixedCase

Function 'BitBook._mint' (bitbook.sol#260-268) is not in mixedCase

Function 'BitBook._burn' (bitbook.sol#271-277) is not in mixedCase

Function 'BitBook._approve' (bitbook.sol#280-286) is not in mixedCase

Function 'Context._msgSender' (Staking.sol#35-37) is not in mixedCase

Function 'Context._msgData' (Staking.sol#39-42) is not in mixedCase

Function 'Ownable._transferOwnership' (Staking.sol#122-126) is not in mixedCase

Struct 'BitBookStake._StakeingId' (Staking.sol#145-147) is not in CapWords

Parameter '_bitBookToken' of BitBookStake. (Staking.sol#159) is not in mixedCase

Parameter '_amount' of BitBookStake.deposit (Staking.sol#169) is not in mixedCase

Parameter '_to' of BitBookStake.safeTransfer (Staking.sol#188) is not in mixedCase

Parameter '_amount' of BitBookStake.safeTransfer (Staking.sol#188) is not in mixedCase

Parameter '_from' of BitBookStake.setFeePercentage (Staking.sol#194) is not in mixedCase

Parameter '_to' of BitBookStake.setFeePercentage (Staking.sol#194) is not in mixedCase

Parameter '_pcent' of BitBookStake.setFeePercentage (Staking.sol#194) is not in mixedCase

Parameter '_from' of BitBookStake.viewFeePercentage (Staking.sol#201) is not in mixedCase

Parameter '_to' of BitBookStake.viewFeePercentage (Staking.sol#201) is not in mixedCase

Parameter '_account' of BitBookStake.calculateFee (Staking.sol#208) is not in mixedCase

Parameter '_stakerID' of BitBookStake.calculateFee (Staking.sol#208) is not in mixedCase

Parameter '_stakerID' of BitBookStake.withdraw (Staking.sol#224) is not in mixedCase

Parameter '_user' of BitBookStake.rewardCalc (Staking.sol#238) is not in mixedCase

Parameter '_id' of BitBookStake.rewardCalc (Staking.sol#238) is not in mixedCase

Parameter '_staker' of BitBookStake.stakingId (Staking.sol#247) is not in mixedCase

Parameter '_daily' of BitBookStake.updateReward (Staking.sol#251) is not in mixedCase

Parameter '_monthly' of BitBookStake.updateReward (Staking.sol#251) is not in mixedCase

Parameter '_to' of BitBookStake.failsafe (Staking.sol#256) is not in mixedCase

Parameter '_amount' of BitBookStake.failsafe (Staking.sol#256) is not in mixedCase

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the BitBook 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 BitBook 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; all of them are fixed and reviewed now.

BitBook

QuillAudits

Canada, India, Singapore and United Kingdom

audits.quillhash.com

audits@quillhash.com