QuillAudits

Audit Report
May, 2021

b-cube.ai

Contents

Introduction

On **26th May 2021** – QuillAudits Team performed a security audit for **BCUBE** smart contracts. The code for audit was taken from the following link:

Branch: feat/new-sale-contracts
https://github.com/b-cube-ai/b-cube-ico/blob/feat/new-sale-contracts/contracts/B3NewSale.sol
[Commit Hash: 13f9f8a991a65f3e0ca60a2078121fdd5a921fb0]

https://github.com/b-cube-ai/b-cube-ico/blob/feat/new-sale-contracts/contracts/NewSaleTreasury.sol
[Commit Hash: 13f9f8a991a65f3e0ca60a2078121fdd5a921fb0]

Updated Contracts:
https://github.com/b-cube-ai/b-cube-ico/blob/feat/new-sale-contracts/contracts/B3NewSale.sol
[Commit Hash: 67717250db1b72929121181e3341dcb4485b6a82]

Overview of Contract

BCUBE is a secure, efficient, and easy-to-use blockchain-based platform providing solutions for the common pitfalls faced by crypto traders.
An all-in-one AI-driven Platform that offers:

- Trading on autopilot for CEX & DEX, cutting-edge DeFi solutions
- In-house strategies using AI/ML dynamically adapting to the market
- The marketplace of premium quality Crypto Trading Signals & Bots
- Build your own AI/ML bot on top of our already working strategies
- Educational courses, webinars, community & consultancy

BCUBE Features

- The marketplace of best quality trading signals & bots
- AI-driven Quantitative Finance
- The in-house institutional-grade execution engine
- The in-house Sentiment Analysis engine
- Build your own AI/ML bots
- Non-custodial & regulated
- Profits-sharing Smart Contract
- High-quality educational contents

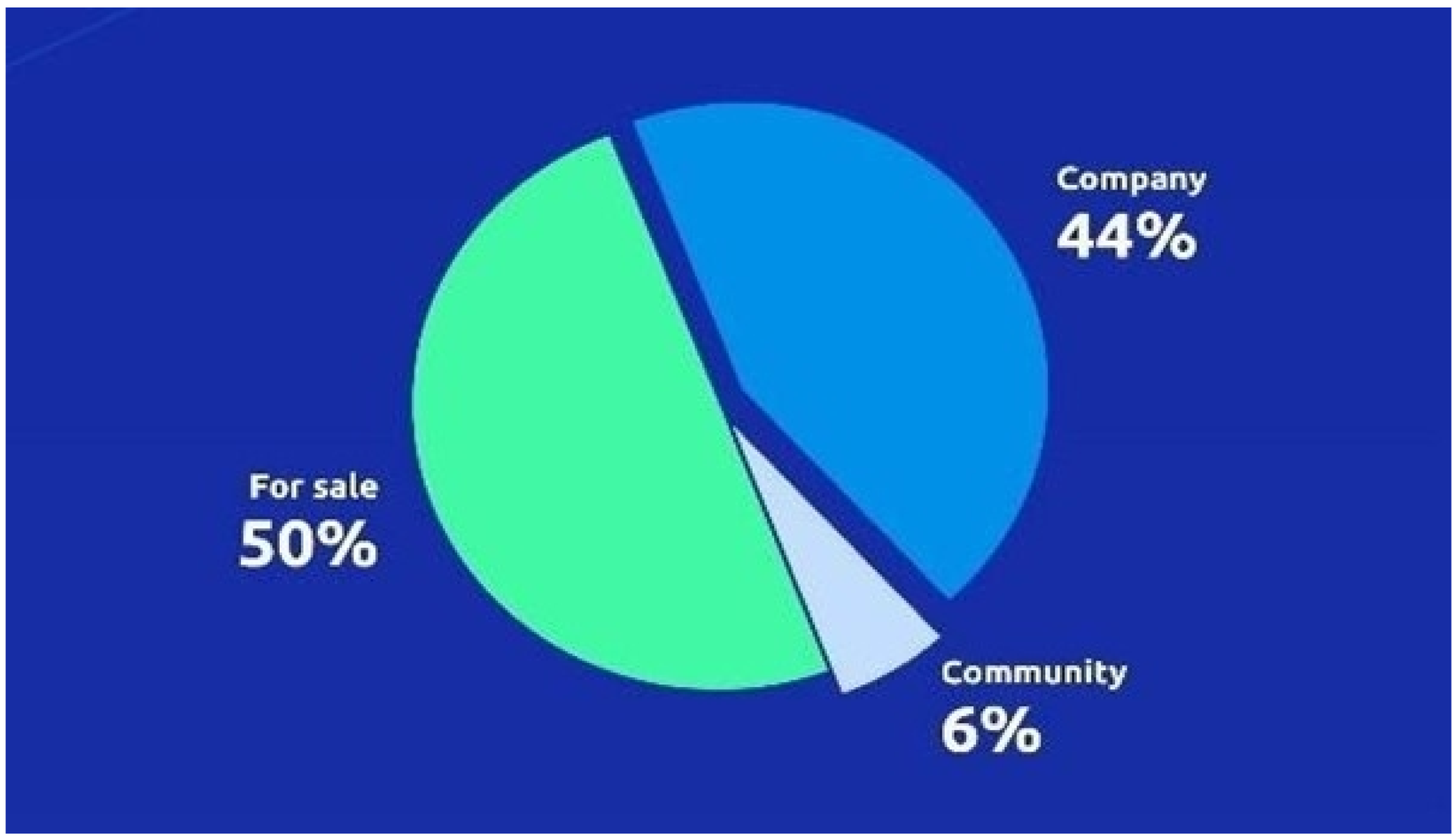
- Solutions for CeFI & DeFI
- Easy-to-use platform

BCUBE Token Sale

The BCUBE token is the fuel of our ecosystem for gaining access to free trading signals, bots, education courses, and various other benefits by staking the required number of BCUBE tokens, get extra benefits by holding for certain periods of time and earn interests (APR) by participating in the Liquidity Pool. The APR is paid on an hourly basis.

Start of sale
2021-05-26 13:30:00 UTC
End of sale
2021-05-28 13:30:00 UTC
Listing Time (starting of vesting period)
2021-05-30 13:30:00 UTC
Number of tokens for sale
900.000 BCUBE
Tokens exchange rate
1 BCUBE = 0.6 USD
Acceptable currencies
ETH, USDT
Minimal transaction amount
$500
Maximal transaction amount
$2500

Tokenomics



Token holders	Initial locking period after listing	Vesting rate
Clients Round, Pre-Seed & Seed participant	1 month	25 % / month
Private Sale participant	6.25% available at TGE	6.25 % / week
Public Sale participant	8.33% available at TGE	8.33 % / week
Team	6 months	12.5 % / 6 months
Advisor	6 months	25 % / 6 months
Development Fund	6 months	25 % / 6 months

Scope of Audit

The scope of this audit was to analyse **BCUBE** smart contract codebase for quality, security, and correctness. Following is the list of smart contracts included in the scope of this audit:

- B3NewSale.sol
- NewSaleTreasury.sol

OUT-OF-SCOPE: External contracts, External Oracles, other smart contracts in the repository or imported smart contracts, economic attacks.

Checked Vulnerabilities

We have scanned BCUBE smart contracts 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 (Unexpected) Throw
- DoS with (Unexpected) revert
- 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
- Address hardcoded
- Using delete for arrays
- Integer overflow/
- underflow
- Locked money
- Private modifier
- Revert/require functions
- Using var
- Visibility
- Using blockhash
- Using SHA3
- Using suicide
- Using throw
- Using inline assembly

Techniques and Methods

Throughout the audit of BCUBE smart contracts 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 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

A combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy regarding the scope of the smart contract audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of smart contracts and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

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.

Code Review / Manual Analysis
Manual Analysis or review of code was done to identify new vulnerability 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.

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

Gas Consumption
In this step, we have checked the behaviour of smart contract 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, Ganache, Solhint, Mythril, Manticore, Slither.

Issue Categories

High severity issues

Issues that must be fixed before deployment else they can create major issues.

Medium level severity issues

These issues will not create major issues in working but affect the performance of the smart contract.

Low level severity issues

These issues are more suggestions that should be implemented to refine the code in terms of gas, fees, speed and code accuracy

Number of issues per severity

	High	Medium	Low	Total Issues
Open	0	0	0	0
Closed	0	3	0	3

Issues Found – Code Review / Manual Testing

High severity issues

No Issue found under this category

Low level severity issues

No Issue found under this category

Medium severity issues

B3NewSale Contract

1. **Consider adding a check for _openingTime in the constructor**

   Code Lines: 105-119
   Consider adding a check to ensure the openingTime provided is greater than the current time.

   Auditors Remarks: Fixed

2. **Consider adding a check for _closingTime in the constructor**

   Code Lines: 105-119
   Consider adding a check to ensure the closingTime provided is greater than the opening time.

   Auditors Remarks: Fixed

3. **Consider adding a check for wallet in the constructor**

   Code Lines: 105-119

   Auditors Remarks: Fixed

NewSaleTreasury Contract

None

Automated Testing

Slither

Slither is an open-source Solidity static analysis framework. This tool provides rich information about Ethereum smart contracts and has the critical properties. It runs a suite of vulnerability detectors, prints visual information about contract details, and provides an API to easily write custom analyses.

B3NewSale.sol

```
INFO:Printers:
Compiled with solc
Number of lines: 1222 (+ 0 in dependencies, + 0 in tests)
Number of assembly lines: 3
Number of contracts: 15 (+ 0 in dependencies, + 0 tests)

Number of optimization issues: 11
Number of informational issues: 28
Number of low issues: 3
Number of medium issues: 0
Number of high issues: 0

ERCs: ERC20

+-----------------------+-------------+-------+-----------------------+--------------+----------------------+
|          Name         | # functions | ERCS  |       ERC20 info      | Complex code |       Features       |
+-----------------------+-------------+-------+-----------------------+--------------+----------------------+
| AggregatorV3Interface |      5      |       |                       |      No      |                      |
|         Roles         |      3      |       |                       |      No      |                      |
|     SignedSafeMath    |      5      |       |                       |      No      |                      |
|        Pausable       |      13     |       |                       |      No      |                      |
|        SafeMath       |      8      |       |                       |      No      |                      |
|         IERC20        |      6      | ERC20 |      No Minting       |      No      |                      |
|                       |             |       | Approve Race Cond.    |              |                      |
|                       |             |       |                       |              |                      |
|       SafeERC20       |      6      |       |                       |      No      | Tokens interaction   |
|        Address        |      3      |       |                       |      No      |      Send ETH        |
|                       |             |       |                       |              |      Assembly        |
|       SafeCast        |      5      |       |                       |      No      |                      |
|       B3NewSale       |      34     |       |                       |      No      |    Receive ETH       |
|                       |             |       |                       |              |      Send ETH        |
+-----------------------+-------------+-------+-----------------------+--------------+----------------------+
INFO:Slither:B3NewSale.sol analyzed (15 contracts)
```

NewSaleTreasury.sol

```
INFO:Printers:
Compiled with solc
Number of lines: 1369 (+ 0 in dependencies, + 0 in tests)
Number of assembly lines: 3
Number of contracts: 16 (+ 0 in dependencies, + 0 tests)

Number of optimization issues: 10
Number of informational issues: 31
Number of low issues: 6
Number of medium issues: 11
Number of high issues: 0

ERCs: ERC20

+-----------------------+-------------+-------+----------------------+---------------+---------------------+
|          Name         | # functions | ERCS  |      ERC20 info      | Complex code  |       Features      |
+-----------------------+-------------+-------+----------------------+---------------+---------------------+
| AggregatorV3Interface |      5      |       |                      |      No       |                     |
|         Roles         |      3      |       |                      |      No       |                     |
|     SignedSafeMath    |      5      |       |                      |      No       |                     |
|        Pausable       |      13     |       |                      |      No       |                     |
|        SafeMath       |      8      |       |                      |      No       |                     |
|         IERC20        |      6      | ERC20 |     No Minting       |      No       |                     |
|                       |             |       | Approve Race Cond.   |               |                     |
|                       |             |       |                      |               |                     |
|       SafeERC20       |      6      |       |                      |      No       | Tokens interaction  |
|        Address        |      3      |       |                      |      No       |      Send ETH       |
|                       |             |       |                      |               |      Assembly       |
|       SafeCast        |      5      |       |                      |      No       |                     |
|    NewSaleTreasury    |      38     |       |                      |      Yes      |     Receive ETH     |
|                       |             |       |                      |               |      Send ETH       |
+-----------------------+-------------+-------+----------------------+---------------+---------------------+
INFO:Slither:NewSaleTreasury.sol analyzed (16 contracts)
```

Slither didn't raise any critical issue with smart contracts. The smart contracts were well tested and all the minor issues that were raised have been documented in the report. Also, all other vulnerabilities of importance have already been covered in the Findings and Tech Details section of the report.

Mythril

Mythril is a security analysis tool for EVM bytecode. It detects security vulnerabilities in smart contracts built for Ethereum. It uses symbolic execution, SMT solving and taint analysis to detect a variety of security vulnerabilities.

Mythril did not detect any high severity issue. All the considerable issues raised by Mythril are already covered in the **Issues Found** section of this report.

Manticore

Manticore is a symbolic execution tool for the analysis of smart contracts and binaries. During Symbolic Execution / EVM bytecode security assessment did not detect any high severity issue. All the considerable issues are already covered in the Findings and Tech Details of this report.

Closing Summary

Overall, the smart contract code is extremely well documented, follows a high-quality software development standard, contains many utilities and automation scripts to support continuous deployment/ testing/ integration, and does NOT contain any obvious exploitation vectors that QuillAudits was able to leverage within the timeframe of testing allotted. Overall, the smart contracts adhered to ERC20 guidelines. No critical or major vulnerabilities were found in the audit. Several issues of **medium severity were found and reported during the audit.**

The outcome of this security audit is satisfactory; due to time and resource constraints, only testing and verification of essential properties was performed to achieve objectives and deliverables set in the scope. QuillAudits recommends performing further testing to validate extended safety and correctness in context to the whole set of contracts.

The issues have been fixed by the team.

Disclaimer

QuillHash audit is not a security warranty, investment advice, or an endorsement of the BCUBE 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 BCUBE Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

b-cube.ai

QuillAudits

Canada, India, Singapore and Uniked Kingdom

audits.quillhash.com

audits@quillhash.com