Smart Contract Security Assessment

Audit Report

For Suresh
22 October 2021

Creator Contact Info

telegram: https://t.me/krypto_dev
email: kryptodev7@gmail.com

Table of Contents

Summary

This report has been prepared for BigBullCoin to discover issues and vulnerabilities in the source code of the BigBullCoin project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

1. Project Summary

Project Name: BigBullCoin

Description: Mint and burnable BEP20 token

Platform: BSC

Language: Solidity

Codebase:

https://bscscan.com/address/0xac02b580e1dcfff6b58980b98fa5a56fca1347e4#code

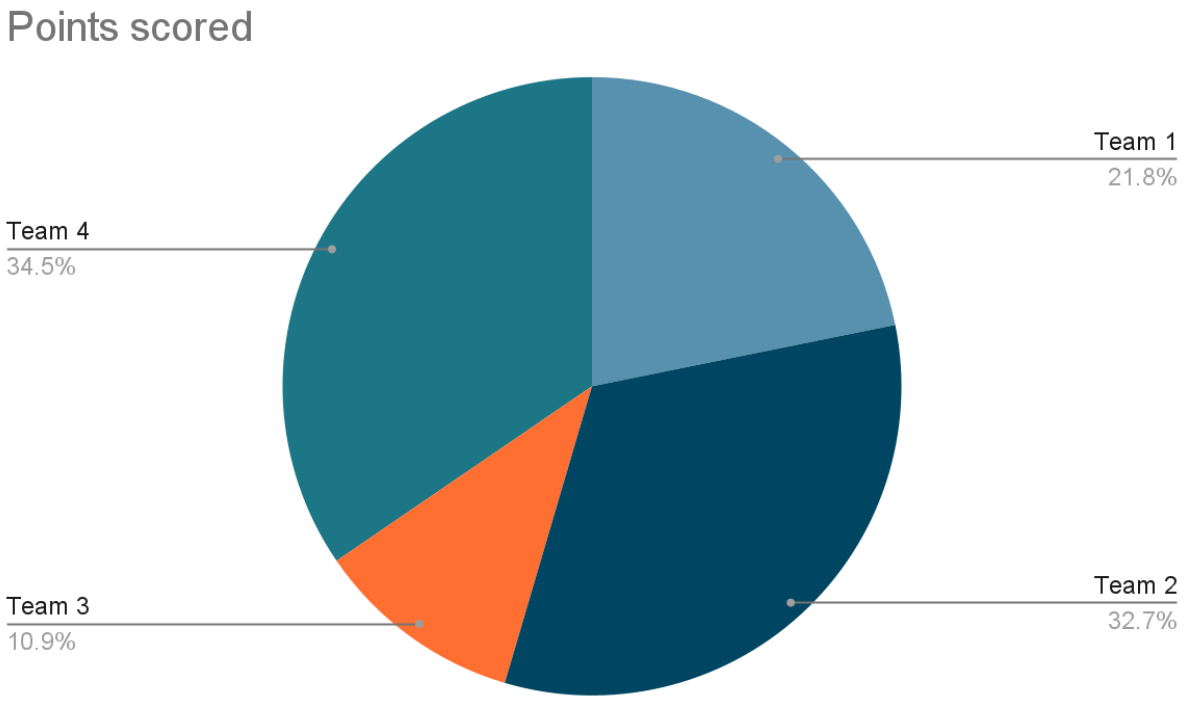
https://github.com/jiju50/BigBullCoin/blob/main/BBC.sol
Commit: 9cd4a4d4ca5f6a34952a8e840f7f305cac6aeef6

2. Audit Summary

Delivery Date: Oct 22, 2021
Audit Methodology: Static Analysis, Manual Review

3. Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
High						
Medium	2	2				
Low	2	2				
Informational	5	5				
Discussion						
Total	9	9				



4. Classification of Issues

- High: Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
- Medium: Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
- Low: Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
- Informational: Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

ID	Title	Category	Severity	Status
01	Centralization Risk	Centralization /Privilege	Medium	Pending
02	Mint Function	Centralization /Privilege	Medium	Pending
03	No validation for airdrop	Logical Issue, Inconsistency	Low	Pending
04	Receiver address can be set 0	Logical Issue, Inconsistency	Low	Pending
05	Low compiler version	Coding style	Informational	Pending
06	not-well structured contract	Coding Style	Informational	Pending
07	Needless codes	Gas Optimization	Informational	Pending
08	Symbol is lowercase	Coding Style	Informational	Pending
09	Function can be made external	Coding Style	Informational	Pending

01 - Centralization Risk

Category	Severity	Location	Status
Centralization /Privilege	Medium	BBC.sol: 190	Pending

Description

In the contract BBC, the role _owner has the authority over the following function:

- mint() which can mint tokens as much as possible

As well as:

- Reception of 1,000,000,000 tokens minted to on construction

Any compromise to the _owner account may allow the hacker to take advantage of this and drastically affect the contract state.

Recommendation

We advise the client to carefully manage the _owner account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;

- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

02 - Mint Function

Category	Severity	Location	Status
Centralization /Privilege	Medium	BBC.sol: 190	Pending

Description

The mint function could be used to pre-mint tokens for legitimate uses including, but not limited to, the injection of initial liquidity, token presale, or airdrops; however, this function may also be used to premint tokens for dumping.

Recommendation

Consider being forthright if this mint function has been used by letting your community know how much was minted, where they are currently stored, if a vesting contract was used for token unlocking, and finally the purpose of the mints.

03 - No validation for airDrop

Category	Severity	Location	Status
Logical Issue, Inconsistency	Low	BBC.sol: 225	Pending

Description

If the token balance of that airdrop function caller is less than the total amount for airdrop, it will be failed.

Recommendation

Consider adding validation that check user’s token balance is bigger than the total amount.

04 - Receiver address can be set 0

Category	Severity	Location	Status
Logical Issue, Inconsistency	Low	BBC.sol: 225	Pending

Description

If the receiver address is set as 0, tokens can be sent to 0 address and wasted.

Recommendation

Consider adding validation that checks if each receiver address is not 0.

05 - Low compiler version

Category	Severity	Location	Status
Coding style	Informational	BBC.sol: 1	Pending

Description

Current solidity compiler version is 0.5.16, it’s old version.

Recommendation

Consider updating current solidity version to up to the latest one

06 - Not-well structured

Category	Severity	Location	Status
Coding style	Informational	BBC.sol: 113 ~ 243	Pending

Description

BigBullcoin contract is based on BEP20 contract. Current Bigbullcoin contract is mixed with normal BEP20 contract and custom codes(airDrop, Mint).

Recommendation

Consider separate into normal BEP20 contract and custom Bigbullcoin contract.

07 - Needless codes

Category	Severity	Location	Status
Coding style	Informational	BBC.sol: 18, 239~242	Pending

Description

There is Needless codes that can increase Gas fee when deploying the contract

Recommendation

Consider removing needless codes from the contracts in locations mentioned above.

08 - Token Symbol is lowercase

Category	Severity	Location	Status
Coding style	Informational	BBC.sol: 127	Pending

Description

Current Token Symbol is “bbc”. It can be Uppercase letters.

Recommendation

Consider updating the current token symbol to uppercase.

09 - Mint function can be made external

Category	Severity	Location	Status
Coding style	Informational	BBC.sol: 190	Pending

Description

The mint function can be made external, which signifies that they are not used within the contract themselves.

Recommendation

Consider marking the mint function that is not used within the contract but only externally as external.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.