Smart Contract Security Audit

Bamboo DeFi

2021-05-27

1. Introduction

BAMBOO DEFI is designed to be the global reference platform used to exchange, save and "cultivate" cryptocurrencies with the best possible ratio without putting in risk the stability of the project, taking advantage of well-tested and audited protocols.

The BambooDeFi project primarily derives its functionalities from Uniswap V2 and implement the SushiSwap migration, but additionally integrating functionalities, such as an incentive program based on multipliers in Yield Farming + Yield Farming with blocking Staking. Therefore, the project wants to analyze the risks derived from its implementation.

As requested by **Bamboo DeFi** and as part of the vulnerability review and management process, **Red4Sec** has been asked to perform a security code audit in order to evaluate the security of the **Binance-Bridge contracts.**

All information collected here is strictly CONFIDENTIAL and may only be distributed by Bamboo Defi with Red4Sec express authorization.

2. Disclaimer

This document only represents the results of the code audit conducted by Red4Sec Cybersecurity and should not be used in any way to make investment decisions or as investment advice on a project.

Likewise, the report should not be considered neither "endorsement" nor "disapproval" of the guarantee of the correct business model of the analyzed project.

3. Scope

The scope of this evaluation includes the following components:

- Zookeper.sol:
  760f9aa7331741bd7daecb50e6a3018d8cf2f12d0ed039dcdd83b5469b338cad
- BridgeBsc.sol:
  8091dbc9504e9f8700135720f055140f1c319cd7ca4fb2bdf0a9f7d8000f5215
- BridgeEth.sol:
  c6009227b5118f8baf834d10fe6d222bba2fc35388028eb8ee3ae724281ab315
- IToken.sol:
  4880429e477141747c2e23aeac2719c631b9ba897d11afa7fdfa0e00afe67515
- TokenBase.sol:
  9e40d2dbb37166a69a3d7b8d287d3ccf2f9cc709b248e8f699dc34c8306bade9
- TokenBsc.sol:
  daa5718bc9d816e54d02a3fcb562701265f38e1e6c42b3384f3c747d81de5923
- TokenEth.sol:
  9f9d2fed255823e6f806d82205a5eda6c4c6c0782b8c1690bb2125bf298d10e9

4. Conclusions

To this date, 27th of May 2021, the general conclusion resulting from the conducted audit, is that the **Binance-Bridge Smart Contracts are secure** and do not present any critical-high known vulnerabilities that could compromise the security of the users and their information, although Red4Sec has found a few potential improvements.

- The Bamboo DeFi team maintains some centralized parts that imply trust in the project until these permissions are transferred to future governance.
- A few low impact issues were detected and classified only as informative, but they will continue to help Bamboo improve the security and quality of its developments.
- The Bamboo team has fixed some of the issues detailed in this report and they have been updated accordingly with the provided remediations.

5. Issues and Recommendations

Contracts Management Risks

The logic design of the eth-bsc-bridge contracts imply certain risks that should be reviewed and considered for their improvement.

Arbitrary Token Minting and Burning

The current implementation of the token delegates management to the admin; such as mining without a capped supply, or the possibility of burning tokens from certain accounts. Both, especially the last mentioned, are dangerous and must be properly controlled in order to not cause harm to users.

Unsecure Ownership Transfer

Even though this logic is intentional, it is necessary to mention that the **TokenBase** contract allows the administrative role to be transferred without proper verifications, for instance it does not verify that the new admin is not *address(0)*.

The modification process of an owner is a delicate process, for this reason it is recommended to adjust the owner's modification logic, to a logic that allows to verify that the new owner is in fact valid and does exist.

The *address(0)* verification has been added in the following commit

- https://github.com/bamboo-defi/binance-bridge/commit/30ce48152e043bfffaa7f288e1e7920b642fc7bb

GAS Optimization

Software optimization is the process of modifying a software system to make an aspect of it work more efficiently or use less resources. This premise must be applied to smart contracts as well, so that they execute faster or in order to save GAS.

On Ethereum blockchain, GAS is an execution fee which is used to compensate miners for the computational resources required to power smart contracts. If the network usage is increasing, so will the value of GAS optimization.

These are some of the requirements that must be met to reduce GAS consumption:

- Short-circuiting.
- Remove redundant or dead code.
- Delete unnecessary libraries.
- Explicit function visibility.
- Use of proper data types.
- Use hard-coded CONSTANT instead of state variables.
- Avoid expensive operations in a loop.
- Pay special attention to mathematical operations and comparisons.

Unused code

The eth-bsc-bridge project implements variables and adds imports that are unused throughout the contracts, by eliminating unused code we will improve the readability of our code and we will also reduce the cost of the executions on certain occasions.

Code Reference

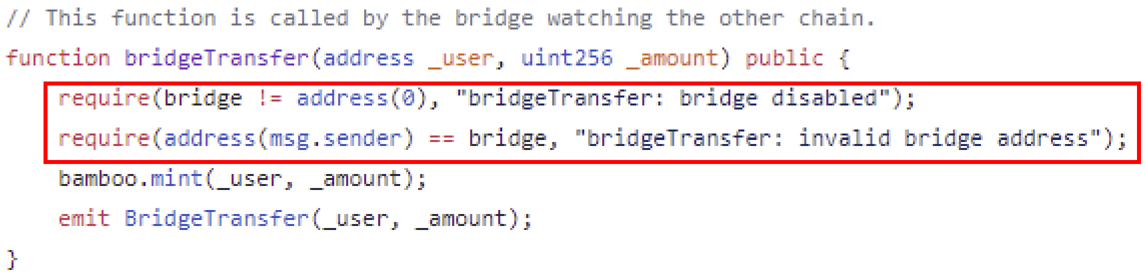
- Unused Import: BridgeBsc.sol:3
- Unused *processedNonces* variable: BridgeEth.sol:12

The remediation has been applied in the following commit

- https://github.com/bamboo-defi/binance-bridge/commit/5d1b63b0637da6d08e9f8ea6b466f7ef6fa2e6e2

Logic Optimizations

The new *bridgeTransfer* function of the **ZooKeeper** contract executes a redundant check during the verification of the bridge.



```
// This function is called by the bridge watching the other chain.
function bridgeTransfer(address _user, uint256 _amount) public {
    require(bridge != address(0), "bridgeTransfer: bridge disabled");
    require(address(msg.sender) == bridge, "bridgeTransfer: invalid bridge address");
    bamboo.mint(_user, _amount);
    emit BridgeTransfer(_user, _amount);
}
```

The first instruction *require* (line 193) results unnecessary since for the condition of the second *require* (line 194) to comply, the condition that the *msg.sender* is *address(0)* should not be given.

Code Reference

- https://github.com/bamboo-defi/bamboodefi/blob/e02a1a7649acc4c2a1e45287b8d2b0b6d35dbc84/contracts/ZooKeeper.sol#L193

The remediation has been applied in the following commit

- https://github.com/bamboo-defi/bamboodefi/commit/c5ceec9c5cdb086363b9f848b108e7174f006e54

Compiler Optimization

We must highlight that the project's contracts are not compiled with optimizations in the current configuration, enabling the compiler optimizations will save execution GAS. It is useful to enable optimization for the contracts since it will reduce the number of instructions to be executed, which will result in GAS savings.

The compiler options have been updated in the following commit

- https://github.com/bamboo-defi/binance-bridge/commit/1bae37a95eaedf15c54ed85d36075aa757a65a92

Absence of Unit Test

The absence of the Unit Test has been detected, during the security review, this is a highly recommended practice that has become mandatory in projects destined to manage large amounts of capital.

For the safety development of any project, at Red4Sec we consider that unitary tests are essential, and its periodical execution is fundamental.

Unit tests have been increased in the following commit

- https://github.com/bamboo-defi/binance-bridge/commit/1bae37a95eaedf15c54ed85d36075aa757a65a92

Outdated Third-Party Libraries

The smart contracts analyzed inherit functionalities from open-zeppelin contracts that have been labeled outdated; this does not imply a vulnerability by itself, because their logic does not present them, but it does imply that an update is not carried out by third party packages or libraries.

Currently the latest version of Open Zeppelin contracts is *4.1.0* but project's package.json sets *4.0.0-beta.0*, therefore it would be convenient to update the reference and avoid the use of untested releases.

The affected libraries have been updated in the following commit

- https://github.com/bamboo-defi/binance-bridge/commit/ec946b62322d8805ce3f5fe5f9b822983989d20f

Outdated Compiler Version

Solc frequently launches new versions of the compiler. Using an outdated version of the compiler can be problematic, especially since *0.8.0* is affected by different known bugs that have already been fixed in later versions.

The compiler has been updated in the following commit

- https://github.com/bamboo-defi/binance-bridge/commit/1ec0472cc45e0b805e65ef6c5a716362517840c3

RED4SEC

Invest in Security, invest in your future