BSCPAD BEP20 & Timelock Audit

The BlockChain Auditor

Prepared by: Jorge Martinez

Version: 1
Date: Feb. 28, 2021

Summary of Findings

This document expresses all security concerns of the BSCPAD BEP 20 and Timelock Contract as expressed by Jorge Martinez. I took care to attempt to find as many ways to improve the security, code efficiency, best practices, and overall function of the smart contracts.

Contract Status: Safe to Use

0 Critical Issue(s) found were found.
0 Medium Issue(s) were found.
0 Low Issue(s) were found.
0 Informational Issue(s) were found.

Solidity Code Coverage

Jorge's Test Suite	BSCPAD	Industry Standard
73.98%	0%	95%

For this audit, I wasn't provided with a testing suite but as part of my audit methodology I developed a test suite to verify the functionality of the BEP20 and timelock contracts, check their security, and to help reveal any underlying issues.

This audit should be seen as one step in the development process with the intent of raising awareness on the meticulous work involved in secure development and making no material statements or guarantees to the operational state of the smart contract(s) once they are deployed. This document is not an endorsement of the reliability or effectiveness of the smart contracts. This is an assessment of the smart contract logic, implementation, and best practices. I cannot take responsibility for any potential consequences of the deployment or use of the smart contract(s) related to the audit.

Test Suite Results

Jorge’s Test Suite

```
  BSCPad Test Suite
    Deployment
      ✓ name should be PB (1259ms)
      ✓ symbol should be PB
      ✓ deployer should be the owner
      ✓ should have 18 decimals
      ✓ total supply should be what I set it to when I deployed the contract
      ✓ deployer should have the total initial supply
    allowance
      ✓ allowance works as expected (211ms)
    approve
      ✓ cannot approve the zero address to move your tokens
    transferFrom
      ✓ allows you transfer an address' tokens to another address
      ✓ reverts you transfer an address' tokens to the zero address
    Minting
      ✓ doesn't allow minting after the contract is constructed
    Burning
      ✓ doesn't allow burning after the contract is constructed
    Whitelist
      ✓ the owner should be able to launch the whitelist (86ms)
    Exchanger
      ✓ only the owner can set the exchanger
      ✓ transferring tokens to the exchange for the first time starts the whitelist
      ✓ none whitelisted users can trade after whitelist period (100ms)

  Timelock Test Suite
    ✓ should set the BSCPAD token as the token to release
    ✓ should have the correct beneficiary set
    ✓ timelock contract should have tokens to release to tokens
    ✓ should release tokens after enough time has passed (65ms)
    ✓ Should not release tokens if you aren't owed anymore (123ms)


  21 passing (2s)

-------------------------|----------|----------|----------|----------|----------------|
File                     |  % Stmts | % Branch |  % Funcs |  % Lines |Uncovered Lines |
-------------------------|----------|----------|----------|----------|----------------|
 contracts/              |    73.98 |    53.33 |    65.91 |    74.19 |                |
  BSCPAD.sol             |    90.16 |    77.27 |    86.96 |    90.16 |... 139,140,152 |
  TokenMultiTimelock.sol |    58.06 |    39.47 |    42.86 |    58.73 |... 223,227,228 |
-------------------------|----------|----------|----------|----------|----------------|
All files                |    73.98 |    53.33 |    65.91 |    74.19 |                |
-------------------------|----------|----------|----------|----------|----------------|
```

Table of Contents

Audit Methodology & Techniques

The BlockChain Auditor has the following auditing process:

1. Our audits include
   a. Review of the specifications, source code, and instructions provided to the TheBlockchainAuditor to clearly identify the desired functionality of the smart contract(s).
   b. Manual line by line review of contract code to spot potential vulnerabilities.
   c. Identification of deviations between desired functionality expressed to the TheBlockchainAuditor and what the smart contract(s) are doing.
2. Automated static and symbolic analysis, as well as verifying testing coverage using the provided test suite.
   a. Automated static and symbolic analysis help determine what inputs cause each part of the smart contract to execute. Analysis of how much of the code base is tested and comparison to industry standard.
3. Examination of smart contracts and development process as a whole, ensuring best practices are followed, allowing improved efficiency and security based on established industry and academic practices.
4. Specific, itemized, and actionable recommendations to assist in securing the smart contract(s) in question.

Contract Checklist

BSCPAD.sol

Contract Vulnerability	
Integer Overflow	Pass
Race Condition	Pass
Denial of Service	Pass
Logical Vulnerability	Pass
Hardcoded Address	Pass
Function Input Parameter Check	Pass
Function Access Control Check	Pass
Random Number Generation	N/A
Random Number Use	N/A
Contract Specification	
Solidity Compiler Version	Pass
Event Use	Pass
Fallback Function Use	Pass
Constructor Use	Pass
Function Visibility Declaration	Pass
Variable Storage Declaration	Pass
Deprecated Keyword Use	Pass
BEP20/223 Standard	Pass
BEP721 Standard	N/A
Business Risk	
Able to Arbitratrily Create Token	N/A
Able to Arbitrarily Destroy Token	N/A
Can Suspend Transactions	Pass
Short Address Attack	Pass
Gas Optimization	
assert()/require()/revert() misused	Pass
Loop Optimization	Pass
Storage Optimization	Pass

Contract Checklist

TokenMultiTimelock.sol

Contract Vulnerability	
Integer Overflow	Pass
Race Condition	Pass
Denial of Service	Pass
Logical Vulnerability	Pass
Hardcoded Address	Pass
Function Input Parameter Check	Pass
Function Access Control Check	Pass
Random Number Generation	N/A
Random Number Use	N/A
Contract Specification	
Solidity Compiler Version	Pass
Event Use	Pass
Fallback Function Use	N/A
Contructor Use	Pass
Function Visibility Declaration	Pass
Variable Storage Declaration	Pass
Deprecated Keyword Use	Pass
BEP20/223 Standard	N/A
BEP721 Standard	N/A
Business Risk	
Able to Arbitratrily Create Token	N/A
Able to Arbitrarily Destroy Token	N/A
Can Suspend Transactions	N/A
Short Address Attack	N/A
Gas Optimization	
assert()/require()/revert() misused	Pass
Loop Optimization	Pass
Storage Optimization	Pass

Issue Classification

Critical

These issues in the smart contract can have catastrophic implications that could ruin your reputation, disrupt the contract's functionality, or impact the client and your users' sensitive information.

Informational

This issue relates to style and security best practices but does not pose an immediate risk.

Medium

An issue classified as medium has relatively small risk and isn't exploitable to circumvent desired functionality and could not have financial consequences but could put user's sensitive information at risk.

Acknowledged

The issue remains in the code but is a result of an intentional business or design decision.

Low

An issue classified as informational does not pose an immediate threat to disruption of functionality and could not be exploited on a recurring basis, however, it should be considered for security best practices or code integrity.

Unresolved

Although the client has been informed of the risk, it was decided to accept it because it was not relevant in the functionality of the smart contract.

Undetermined

The impact of the issue is uncertain and more investigation is required to understand the repercussions of the issue.

Mitigated

Actions were taken to minimize the impact or likelihood of the risk.

Executive Summary

Overall Thoughts

This project was pleasure to audit. OpenZeppelin libraries were used extensively which focused the scope of the audit to the whitelist and timelock functionalities.

I did not find any security vulnerabilities and was able to verify the functionality of the smart contracts. Whitelisted users will be able to trade for a specified time interval that starts as soon as BSCPAD tokens are transferred to an exchange. Afterwards, the trading will open up to non-whitelisted users. Minting and burning is not accessiable after contract construction because the functions are internal the total supply will remain stationary.

Their timelock is also well developed an heavily based off of OpenZeppelin contracts. I verified that the tokens will not be released until a specified amount of time has passed.

Appendix A

File Fingerprints

BSCPAD.sol	1afc133b35daf88f2c9874d1f51b9466
TokenMultiTimelock.sol	ed889eb317782e67e9b53bbdc2d66e16

The Blockchain Auditor is honored to have the opportunity to help verify the functionality and security for BSC Launchpad platform and associated contracts.

The BlockChain Auditor

- Jorge Martinez