BSCPAD Antibot

BEP20 Audit

The BlockChain Auditor

Prepared by: Jorge Martinez

Version: 2
Date: March 22, 2021

Summary of Findings

This document expresses all security concerns of the BSCPAD BEP20 smart 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: Ready for Deployment

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
88.07%	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, security, and to help reveal any underlying issues of the BSCPAD smart contract ecosystem.

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 BEP20TokenWhitelisted (386ms)
    ✓ symbol should be B20WL
    ✓ deployer should be the owner
    ✓ should have 18 decimals
    ✓ total supply should be 1 million tokens
    ✓ deployer should have the total initial supply
  allowance
    ✓ allowance works as expected (281ms)
  approve
    ✓ cannot approve the zero address to move your tokens
  transferFrom
    ✓ allows you transfer an address' tokens to another address (38ms)
    ✓ reverts you transfer an address' tokens to the zero address
  Ownership
    ✓ only the owner can transfer ownership to another address (58ms)
    ✓ owner cannot transfer ownership to the zero address
    ✓ the owner can renounce ownership of the contract
  Whitelist
    ✓ transfers revert without the LGE whitelist
    ✓ token transfers revert without the pair address set
    ✓ creating the LGE whitelist requires duration and amountsMax of equal length
    ✓ transferring tokens to the pair address begins the LGE (88ms)
    ✓ transferring tokens reverts if you're not on the whitelist (125ms)
    ✓ whitelisters cannot buy more than the specified amount max (53ms)
    ✓ whitelisted addresses can buy up to the specified max (72ms)
    ✓ whitelist admin can add whitelist addresses using modifyLGEWhitelist (70ms)
    ✓ whitelist admin can modify the whitelist duration
    ✓ whitelist admin can modify the max tokens that can be bought during the whitelist
    ✓ whitelist admin can call the modifyLGEWhitelist and not change anything
    ✓ when the whitelist round is over, getLGEWhitelistRound returns 0
    ✓ whitelist admin cannot modify a whitelist that doesn't exist
    ✓ whitelist admin cannot set amountMax less than zero
    ✓ whitelist admin can renounce their whitelister permissions
    ✓ whitelist admin can tranfer their whitelisting permission to another address
    ✓ whitelist admin cannot transfer their whitelisting permission to the zero address


  30 passing (2s)

----------------------------|----------|----------|----------|----------|----------------|
File                        |  % Stmts | % Branch |  % Funcs |  % Lines |Uncovered Lines |
----------------------------|----------|----------|----------|----------|----------------|
 contracts/                 |    88.07 |    78.33 |    85.37 |    87.61 |                |
  BEP20TokenWhitelisted.sol |    88.07 |    78.33 |    85.37 |    87.61 |... 248,263,264 |
----------------------------|----------|----------|----------|----------|----------------|
All files                   |    88.07 |    78.33 |    85.37 |    87.61 |                |
----------------------------|----------|----------|----------|----------|----------------|
```

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

BEP20Whitelisted.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
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	Pass
Able to Arbitrarily Destroy Token	Pass
Can Suspend Transactions	Pass
Short Address Attack	Pass
Gas Optimization	
assert()/require()/revert() misused	Pass
Loop Optimization	Pass
Storage Optimization	Pass

Executive Summary

Overall Thoughts

Zero issues were uncovered in the BSCPAD BEP-20 smart contract. The modular design of the anti-bot whitelisting smart contract made it straightforward to verify all intended functionality. Projects taking advantage of the anti-bot mechanisms can simply extend BEP20Whitelisted and integrate their custom smart contract code. This allows for whitelisted “rounds”, initiated upon token transfer to the pair address for the initial seeded liquidity, in which only whitelisted addresses are authorized to trade. This method provides extendable industry standard protections against bot front running. Aside from the anti-bot mechanisms this is a standard BEP-20 token without fees. It has a stationary supply of one million tokens and projects integrating with the contract can choose to utilize the burn and mint functions as they deem fit.

Appendix A

File Fingerprints

BEP20Whitelisted.sol	04eba1ffc8073645097fb45c46a22691

The Blockchain Auditor is honored to have the opportunity to help verify the functionality of BSCPAD's Antibot BEP20 smart contract. The team behind BSCPAD continues to innovate and I can't wait to see what they do next.

The BlockChain Auditor

- Jorge Martinez