Moonpool
Brainaut Defi Project

This is a free audit report
Published on 22th March, 2021

Brainaut Defi Security Analysis

By MoonPool

Overview Abstract

In this report, we consider the security of the Brainaut Defi project. Our main task is to find and describe security issues in the smart contracts of the platform to the team.

Limitations and use of the report

Broadly speaking, the assessments can not uncover all vulnerabilities of the given smart contracts, thus, it is not guaranteed that the system is secured even if no vulnerabilities are found. The focus of the assessments was limited to only 2 given smart contracts, other contracts were excluded (including external libraries or third party codes).
The report is based on provided source codes via google drive, we do not guarantee the provided codes to be the same as deployed source codes.

Note: The report is not investment advice.
The team can publish this report to the community.

Summary

We have found **1** medium severity issue and **6** low severity issues.

The source codes were mainly influenced from well-known projects on Ethereum and Binance Smart Chain.

Recommendations

We recommend the team to fix all issues, as well as tests coverage to ensure the security of the contracts.

Assessment Overview

Scope of the audit

Source codes for the audit was initially taken from here with the following **2** contracts:

1. Crowdsale
2. BrainautOther

Source codes are out of scope for this report.

System Overview

We refer to the documentation on their website, as well as their whitepaper.

1. Crowdsale:

- A standard crowdsale contract that allows users to buy tokens directly using the native token (i.e: BNB on Binance Smart Chain).
- Users can buy the token by calling **buyTokens** function, or simply transfer **BNB** to the contract.
- Minimum cap per transaction: **1 BNB**
- Maximum cap per transaction: **20 BNB**
- Tokens will be released immediately.

2. Brainaut Token:

- A fee-on-transfer token that works as normal ERC20 token, except:

+ It burns **_BURN_FEE** percent of amount on each transfer.

+ It collects and re-distributes to all token holders **_TAX_FEE** percentage of

amount on each transfer.

+ **_BURN_FEE** and **_TAX_FEE** are in range [50, 1000], corresponding to [0.5%, 10%].

Findings

We have found 1 medium severity issue and 6 low severity issues.

Crowdsale

[Low] **buytokens:** No buying cap for addresses

There is a cap per transaction (1 BNB <= cap <= 20 BNB), however, there is no cap per address, it means one address can make many transactions to buy 20 BNB worth of tokens.

[Low] **_getTokenAmount:** unclear logics for calculating amount of tokens from BNB amount

The formula for calculating amount of tokens from BNB amount:
TokenAmount = BNB_Amount * rate / 1011**
There is no comment on why the formula is using **10**11** as the denominator. Consider adding comments on this formula. It should be related to the token's decimal.

[Low] **_forwardFunds**: switch to use **call()** instead of **transfer()**

When using the **transfer()** function to transfer BNB, the gas cost is fixed as 2300 and could be affected if the gas costs per operation increases. Consider switching to use **call()** function instead. Checkout this article for more information.

[Low] **Code Convention**

1. Naming convention: using local names like *Amount* in **buyTokens, _preValidatePurchase, _getTokenAmount**, or *sale* for contract name is not following coding best practices.

2. Line length: Line length should be kept no longer than 100-120 characters. Check out lines 241, 401, 409, 429, 449, 469 and recommend keeping them shorter.

3. Function orders: Incorrect function orders, orders should be: external -> public -> internal -> private and write -> read functions.

Brainaut Token

The source code is influenced from Reflect Finance

[Medium] **excludeAccount**: Invalid hardcoded address for a (Uniswap) router on Binance Smart Chain.

In the **excludeAccount** function, it is required the account must be different from the address **0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D** which is the Uniswap Router address on Ethereum. However, this is not a router on Binance Smart Chain (not Pancake/Bakery/Julswap/etc router). Consider changing the hard coded address to a correct one on BSC.

[Low] **_setTaxFee, _setBurnFee**: Invalid condition check logic and comment

The logics show that fees should be in the range of 0.05% to 10%, however, the revert messages indicate that fees must be from 1% to 10%.

[Low] **Code Convention**

1. **Line length:** Line length should be kept no longer than 100-120 characters, many lines are longer than 100 characters in the code, recommend keeping them shorter.

2. **Function orders**: Incorrect function orders, orders should be: external -> public -> internal -> private and write -> read functions.

*Note: Except the Owner, other addresses can only make a transfer transaction with at most 10M tokens (**_MAX_TX_SIZE**).*

Testing

The team has not provided any tests for given contracts. We recommend adding full tests coverage to ensure the logic works as expected and the security of the smart contracts.

Comment from the team : all source codes are influenced from some well-known projects on Ethereum and Binance Smart Chain. Some of them have been well-tested and well-audited.

Buy our team a coffee:
0xA0b91Ec92dBE8Af20fc9058bE61a2da09508255c

Request audit from our team