

Submitted audit for **GbtIssuance** on 23 November 2023

Audit result: **Passed**

**Token Address:** - 0x19bfAF9843357aCb87dB558118AC9bC73662D98F

**Name:** GoldenBambooToken

**Symbol:** GBT

**Decimals:** 18

**Network:** Binance smart chain

**Token Type:** ERC20

**Owner:-** 0x00

**Deployer:-** 0x00

**Token Supply:** 1000000000000000000000000

**Checksum:** 27af7fb5e4ae1c007d623902f5e5a456

**Testnet version:**

The tests were performed using the contract deployed on the Binance smart chain Testnet, which can be found at the following address:

<https://testnet.bscscan.com/address/0x19bfaf9843357acb87db558118ac9bc73662d98f#code>

**Tools:**

1. Manual Review: The code has undergone a line-by-line review by the **Ace** team.
2. BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.
3. Slither: The code has undergone static analysis using Slither.

## Static Analysis

A static analysis of the code was performed using Slither.

```

INFO:Detectors:
GbtIssuance.buy(uint256) (GbtIssuance.sol#182-204) performs a multiplication on the result of a division:
- sendAmount = amount.mul(1 * 10 ** uint256(IERC20(_token).decimals()))).div(backstopPrice) (GbtIssuance.sol#190)
- amount = sendAmount.mul(backstopPrice).div(1 * 10 ** uint256(IERC20(_token).decimals())) (GbtIssuance.sol#194)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
Reentrancy in GbtIssuance.buy(uint256) (GbtIssuance.sol#182-204):
  External calls:
  - backstopPrice = IERC20(_token).backstopPrice() (GbtIssuance.sol#189)
  - IERC20(_usdt).transferFrom(account,address(this),amount) (GbtIssuance.sol#197)
  - IERC20(_token).transfer(account,sendAmount) (GbtIssuance.sol#199)
  - IERC20(_usdt).transfer(_token,amount) (GbtIssuance.sol#200)
  State variables written after the call(s):
  - _userBuyTotal[account] += amount (GbtIssuance.sol#201)
  GbtIssuance._userBuyTotal (GbtIssuance.sol#148) can be used in cross function reentrancies:
  - GbtIssuance._userBuyTotal (GbtIssuance.sol#148)
  - GbtIssuance.buy(uint256) (GbtIssuance.sol#182-204)
  - GbtIssuance.getQuota(address) (GbtIssuance.sol#174-179)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
Ownable.changeOwner(address).newOwner (GbtIssuance.sol#82) lacks a zero-check on :
- _owner = newOwner (GbtIssuance.sol#84)
GbtIssuance.setAddress(address,uint256).param (GbtIssuance.sol#165) lacks a zero-check on :
- _token = param (GbtIssuance.sol#167)
- _broker = param (GbtIssuance.sol#169)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in GbtIssuance.buy(uint256) (GbtIssuance.sol#182-204):
  External calls:
  - backstopPrice = IERC20(_token).backstopPrice() (GbtIssuance.sol#189)
  - IERC20(_usdt).transferFrom(account,address(this),amount) (GbtIssuance.sol#197)
  - IERC20(_token).transfer(account,sendAmount) (GbtIssuance.sol#199)
  - IERC20(_usdt).transfer(_token,amount) (GbtIssuance.sol#200)
  Event emitted after the call(s):
  - BuyRecord(account,backstopPrice,amount,IERC20(_token).balanceOf(account).sub(balance)) (GbtIssuance.sol#203)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
SafeMath.add(uint256,uint256) (GbtIssuance.sol#99-103) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.19 (GbtIssuance.sol#25) necessitates a version too recent to be trusted. Consider deploying with 0.8.18.
solc-0.8.22 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Variable GbtIssuance._usdt (GbtIssuance.sol#140) is not in mixedCase
Variable GbtIssuance._total (GbtIssuance.sol#142) is not in mixedCase
Variable GbtIssuance._token (GbtIssuance.sol#144) is not in mixedCase
Variable GbtIssuance._broker (GbtIssuance.sol#146) is not in mixedCase
Variable GbtIssuance._userBuyTotal (GbtIssuance.sol#148) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable GbtIssuance._usdt (GbtIssuance.sol#140) is not in mixedCase
Variable GbtIssuance._total (GbtIssuance.sol#142) is not in mixedCase
Variable GbtIssuance._token (GbtIssuance.sol#144) is not in mixedCase
Variable GbtIssuance._broker (GbtIssuance.sol#146) is not in mixedCase
Variable GbtIssuance._userBuyTotal (GbtIssuance.sol#148) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
GbtIssuance._total (GbtIssuance.sol#142) should be immutable
GbtIssuance._usdt (GbtIssuance.sol#140) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:GbtIssuance.sol analyzed (6 contracts with 93 detectors), 18 result(s) found

```

## Findings:

Critical: 0

High: 0

Medium: 0

Low: 2

Suggestions & Optimizations: 2

# Centralization – Missing Zero Address

Severity: **Low**

function: setAddress

Status: Open

Overview:

functions can take a zero address as a parameter (0x00000...). If a function parameter of address type is not properly validated by checking for zero addresses, there could be serious consequences for the contract's functionality.

```
function setAddress(address param, uint status) external onlyOwner {
    if (status == 0) {
        _token = param;
    } else if (status == 1) {
        _broker = param;
    }
}
```

**Suggestion:**

It is suggested that the address should not be zero or dead.

## Optimization

**Severity:** Low

**subject:** Missing Events

**Status:** Open

**Overview:**

They serve as a mechanism for emitting and recording data onto the blockchain, making it transparent and easily accessible.

```
function setAddress(address param, uint status) external onlyOwner {
    if (status == 0) {
        _token = param;
    } else if (status == 1) {
        _broker = param;
    }
}
```

## Optimization

**Severity:** Informational

**subject:** floating Pragma Solidity version

**Status:** Open

**Overview:**

It is considered best practice to pick one compiler version and stick with it. With a floating pragma, contracts may accidentally be deployed using an outdated.

```
pragma solidity ^0.8.19;
```

**Suggestion**

Adding the latest constant version of solidity is recommended, as this prevents the unintentional deployment of a contract with an outdated compiler that contains unresolved bugs.

## Optimization

**Severity:** Informational

**subject:** Remove safe math

**Status:** Open

**Line:** 98 - 133

**Overview:**

Compiler version above 0.8.0 has the ability to control arithmetic overflow/underflow, It is recommended to remove the unwanted code in order to avoid high gas fees.