Submitted audit for **GbtBroker** on 23 November 2023
Audit result: **Passed**

Token Address: - 0x285A158Db7FBAD6414E36c7eA8EF20392AF23a02
Name: GoldenBambooToken
Symbol: GBT
Decimals: 18
Network: Binance smart chain
Token Type: ERC20
Owner: - 0x0000000000000000000000000000000000000000
Deployer: - 0x0000000000000000000000000000000000000000
Token Supply: 100000000000000000000
Checksum: 210f1528fea91805f3ae5aaac06bbed9
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/0x3eab9a6afeb8745bb17c6d30c3402622854c4f1d#code

Tools:

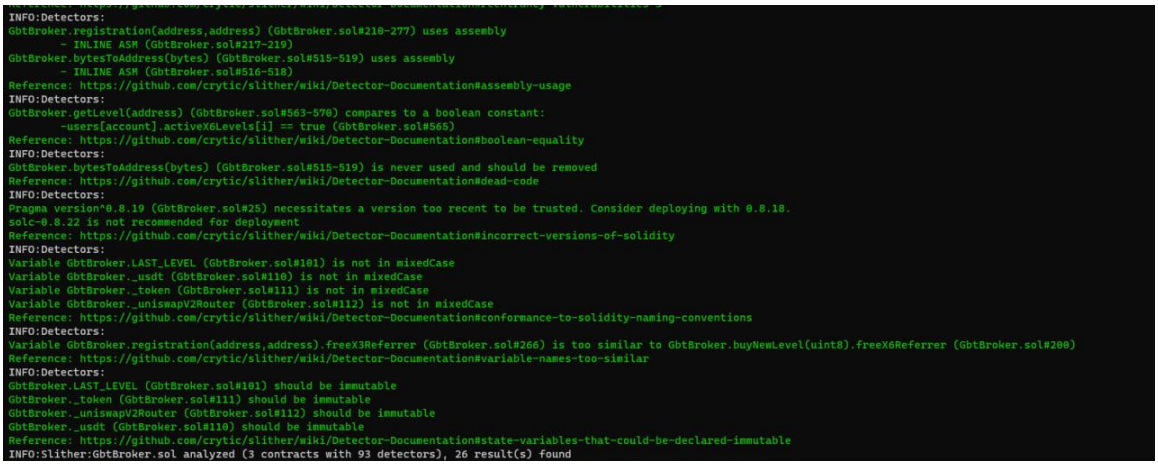
1. Manual Review: The code has undergone a line-by-line review by the **Ace** team.
2. BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.
3. Slither: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

Static Analysis

A static analysis of the code was performed using Slither.

```
INFO:Detectors:
Reentrancy in GbtBroker.buyNewLevel(uint8) (GbtBroker.sol#190-208):
        External calls:
        - _pay(uAmount) (GbtBroker.sol#203)
                - IERC20(_usdt).transferFrom(msg.sender,address(this),uAmount) (GbtBroker.sol#547)
                - IERC20(_usdt).approve(address(_uniswapV2Router),uAmount) (GbtBroker.sol#549)
                - _uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(uAmount,0,path,address(this),block.timestamp) (GbtBroker.sol#553-559)
        - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
                - IERC20(_token).transfer(address(userAddress),left) (GbtBroker.sol#508)
        State variables written after the call(s):
        - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
                - users[users[referrerAddress].x6Matrix[level].firstLevelReferrals[0]].x6Matrix[level].firstLevelReferrals.push(userAddress) (GbtBroker.sol#401)
                - users[referrerAddress].x6Matrix[level].firstLevelReferrals.push(userAddress) (GbtBroker.sol#319)
                - users[userAddress].x6Matrix[level].currentReferrer = referrerAddress (GbtBroker.sol#323)
                - users[userAddress].x6Matrix[level].currentReferrer = users[referrerAddress].x6Matrix[level].firstLevelReferrals[0] (GbtBroker.sol#405)
                - users[users[referrerAddress].x6Matrix[level].firstLevelReferrals[1]].x6Matrix[level].firstLevelReferrals.push(userAddress) (GbtBroker.sol#407)
                - users[users[referrerAddress].x6Matrix[level].currentReferrer].x6Matrix[level].closedPart = referrerAddress (GbtBroker.sol#426)
                - users[userAddress].x6Matrix[level].currentReferrer = users[referrerAddress].x6Matrix[level].firstLevelReferrals[1] (GbtBroker.sol#411)
                - users[users[referrerAddress].x6Matrix[level].currentReferrer].x6Matrix[level].closedPart = referrerAddress (GbtBroker.sol#429)
                - users[ref].x6Matrix[level].secondLevelReferrals.push(userAddress) (GbtBroker.sol#330)
                - users[referrerAddress].x6Matrix[level].firstLevelReferrals = new address[](0) (GbtBroker.sol#434)
                - users[referrerAddress].x6Matrix[level].secondLevelReferrals = new address[](0) (GbtBroker.sol#435)
                - users[referrerAddress].x6Matrix[level].closedPart = address(0) (GbtBroker.sol#436)
                - users[referrerAddress].x6Matrix[level].reinvestCount ++ (GbtBroker.sol#438)
                - users[referrerAddress].x6Matrix[level].secondLevelReferrals.push(userAddress) (GbtBroker.sol#361)
        GbtBroker.users (GbtBroker.sol#103) can be used in cross function reentrancies:
        - GbtBroker.buyNewLevel(uint8) (GbtBroker.sol#190-208)
        - GbtBroker.constructor() (GbtBroker.sol#131-182)
        - GbtBroker.findFreeX3Referrer(address,uint8) (GbtBroker.sol#453-462)
        - GbtBroker.findFreeX6Referrer(address,uint8) (GbtBroker.sol#464-473)
        - GbtBroker.getLevel(address) (GbtBroker.sol#563-570)
        - GbtBroker.getProductCount(address,uint8,uint256) (GbtBroker.sol#581-591)
        - GbtBroker.getReinvestCount(address,uint8) (GbtBroker.sol#573-578)
        - GbtBroker.getSellCount(address,uint8,uint256) (GbtBroker.sol#594-612)
        - GbtBroker.isUserExists(address) (GbtBroker.sol#499-501)
        - GbtBroker.registration(address,address) (GbtBroker.sol#210-277)
        - GbtBroker.updateX3Referrer(address,address,uint8) (GbtBroker.sol#279-312)
        - GbtBroker.updateX6(address,address,uint8,bool) (GbtBroker.sol#399-413)
        - GbtBroker.updateX6Referrer(address,address,uint8) (GbtBroker.sol#314-397)
        - GbtBroker.updateX6ReferrerSecondLevel(address,address,uint8) (GbtBroker.sol#415-451)
        - GbtBroker.userInvestTotal(address) (GbtBroker.sol#522-532)
        - GbtBroker.userQuota(address) (GbtBroker.sol#535-543)
        - GbtBroker.users (GbtBroker.sol#103)
        - GbtBroker.usersActiveX3Levels(address,uint8) (GbtBroker.sol#475-477)
        - GbtBroker.usersActiveX6Levels(address,uint8) (GbtBroker.sol#479-481)
        - GbtBroker.usersX3Matrix(address,uint8) (GbtBroker.sol#483-488)
        - GbtBroker.usersX6Matrix(address,uint8) (GbtBroker.sol#490-497)
Reentrancy in GbtBroker.registration(address,address) (GbtBroker.sol#210-277):
```

```
INFO:Detectors:
Reentrancy in GbtBroker.sendTokenDividends(address) (GbtBroker.sol#503-513):
        External calls:
        - IERC20(_token).transfer(address(userAddress),left) (GbtBroker.sol#508)
        State variables written after the call(s):
        - userRewards[userAddress] += left (GbtBroker.sol#511)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in GbtBroker.buyNewLevel(uint8) (GbtBroker.sol#190-208):
        External calls:
        - _pay(uAmount) (GbtBroker.sol#203)
                - IERC20(_usdt).transferFrom(msg.sender,address(this),uAmount) (GbtBroker.sol#547)
                - IERC20(_usdt).approve(address(_uniswapV2Router),uAmount) (GbtBroker.sol#549)
                - _uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(uAmount,0,path,address(this),block.timestamp) (GbtBroker.sol#553-559)
        - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
                - IERC20(_token).transfer(address(userAddress),left) (GbtBroker.sol#508)
        Event emitted after the call(s):
        - Invest(msg.sender,levelPrice[level]) (GbtBroker.sol#207)
        - Invest(referrerAddress,levelPrice[level]) (GbtBroker.sol#448)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - Invest(idToAddress[1],levelPrice[level]) (GbtBroker.sol#448)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,users[referrerAddress].x6Matrix[level].firstLevelReferrals[0],2,level,uint8(users[users[referrerAddress].x6Matrix[level].firstLevelReferrals[0]].x6Matrix[level].firstLevelRef
errals.length),IERC20(_token).balanceOf(address(this))) (GbtBroker.sol#402)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,referrerAddress,2,level,uint8(users[referrerAddress].x6Matrix[level].firstLevelReferrals.length),IERC20(_token).balanceOf(address(this))) (GbtBroker.sol#320)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,referrerAddress,2,level,2 + uint8(users[users[referrerAddress].x6Matrix[level].firstLevelReferrals[0]].x6Matrix[level].firstLevelReferrals.length),IERC20(_token).balanceOf(ad
dress(this))) (GbtBroker.sol#483)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,users[referrerAddress].x6Matrix[level].firstLevelReferrals[1],2,level,uint8(users[users[referrerAddress].x6Matrix[level].firstLevelReferrals[1]].x6Matrix[level].firstLevelRef
errals.length),IERC20(_token).balanceOf(address(this))) (GbtBroker.sol#408)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,referrerAddress,2,level,4 + uint8(users[users[referrerAddress].x6Matrix[level].firstLevelReferrals[1]].x6Matrix[level].firstLevelReferrals.length),IERC20(_token).balanceOf(ad
dress(this))) (GbtBroker.sol#409)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,ref,2,level,5,IERC20(_token).balanceOf(address(this))) (GbtBroker.sol#338)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,ref,2,level,6,IERC20(_token).balanceOf(address(this))) (GbtBroker.sol#340)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,ref,2,level,3,IERC20(_token).balanceOf(address(this))) (GbtBroker.sol#345)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,ref,2,level,4,IERC20(_token).balanceOf(address(this))) (GbtBroker.sol#347)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,ref,2,level,5,IERC20(_token).balanceOf(address(this))) (GbtBroker.sol#351)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - NewUserPlace(userAddress,ref,2,level,6,IERC20(_token).balanceOf(address(this))) (GbtBroker.sol#353)
                - updateX6Referrer(msg.sender,freeX6Referrer,level) (GbtBroker.sol#204)
        - Reinvest(referrerAddress,freeReferrerAddress,userAddress,2,level) (GbtBroker.sol#443)
```



Findings:
Critical: 0
High: 0
Medium: 0
Low: 2
Suggestions & Optimizations: 1

Centralization – Missing Zero Address

Severity: Low
function: setAddress and setUint
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 registrationExt(address referrerAddress) external {
        registration(msg.sender, referrerAddress);
    }function setUint(uint param, uint status) external onlyOwner {
        if (status == 0) {
            _rewardTimes = param;
        } else if (status == 1) {
            _minAdd = param;
        } else if (status == 2) {
            _helpLp = param;
        } else if (status == 3) {
```

```
            _calculateVirtualDecimals = 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 registrationExt(address referrerAddress) external {
        registration(msg.sender, referrerAddress);
    }
```

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.