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

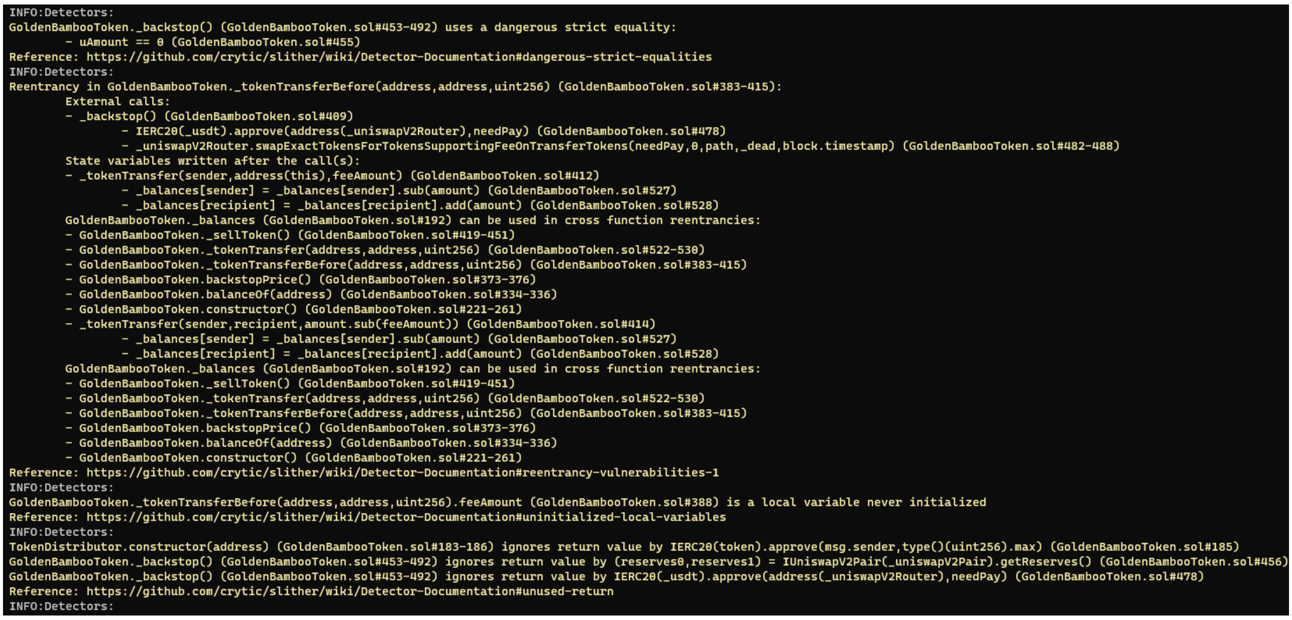
Token Address: - 0x1fC3905830DAf1dF03615E6271BF204DF131A63a
Name: GoldenBambooToken
Symbol: GBT
Decimals: 18
Network: Binance smart chain
Token Type: ERC20
Owner: - 0x0000000000000000000000000000000000000000
Deployer: - 0x0000000000000000000000000000000000000000
Token Supply: 100000000000000000000000
Checksum: 936d13204efb608815228e057534586c
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/0x96ba0763ac9e8f493deda7051e7b53b43ea8e365#code

Tools:

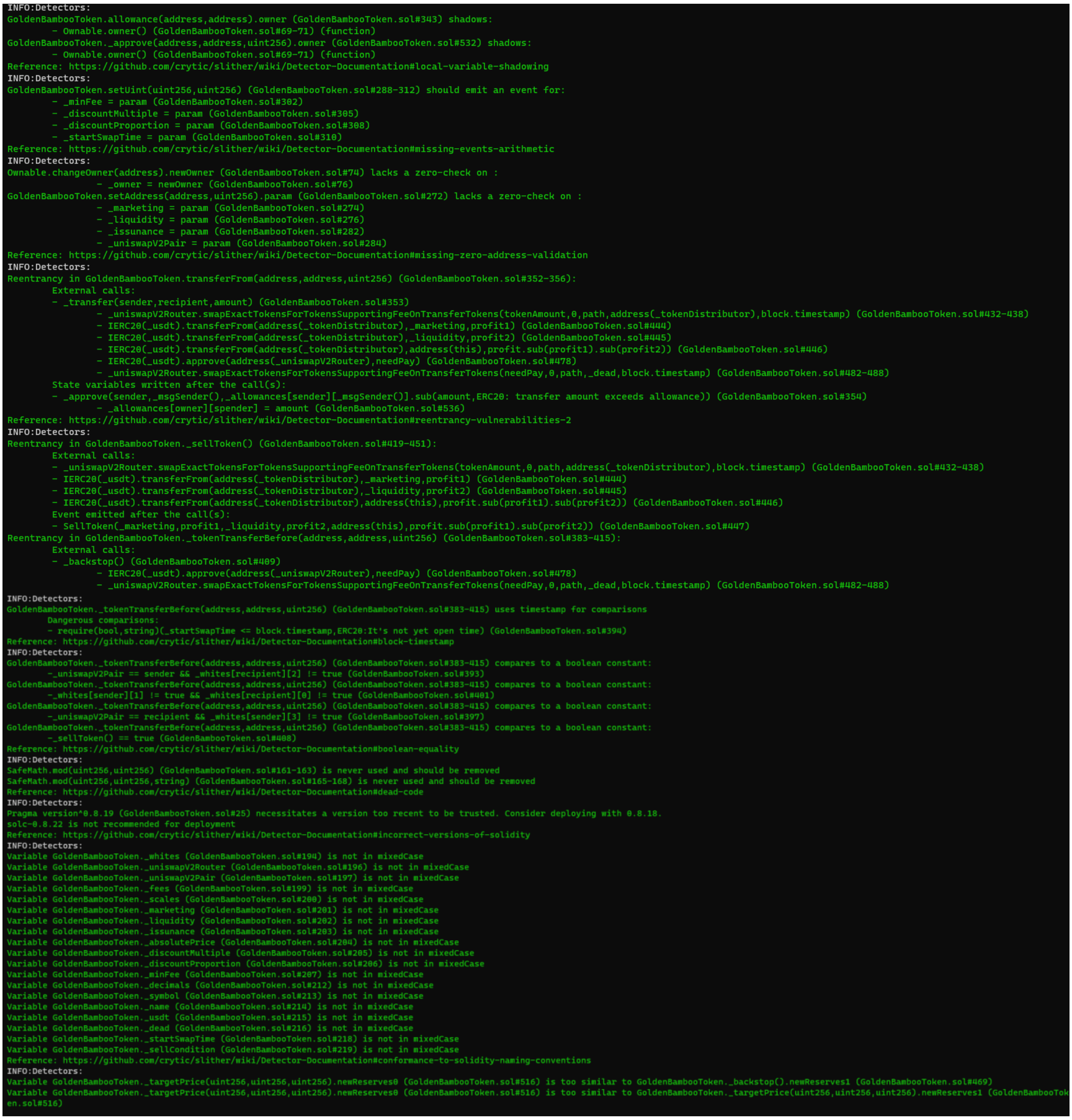
1. Manual Review: The code has undergone a line-by-line review by the **Ace** team.
2. BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.
3. Slither: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

Static Analysis

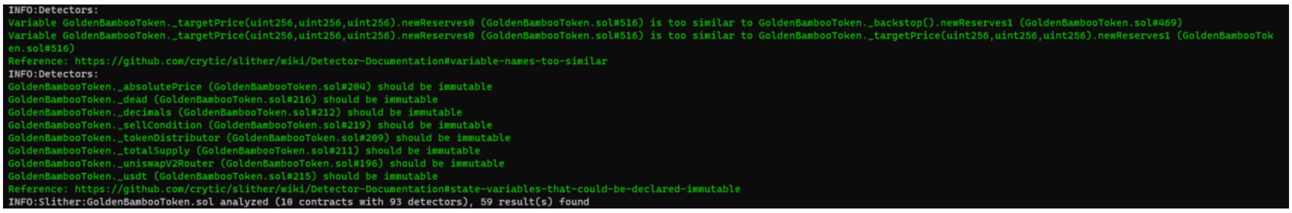
A static analysis of the code was performed using Slither.



```
INFO:Detectors:
GoldenBambooToken._backstop() (GoldenBambooToken.sol#453-492) uses a dangerous strict equality:
	- uAmount == 0 (GoldenBambooToken.sol#455)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
INFO:Detectors:
Reentrancy in GoldenBambooToken._tokenTransferBefore(address,address,uint256) (GoldenBambooToken.sol#383-415):
	External calls:
	- _backstop() (GoldenBambooToken.sol#409)
		- IERC20(_usdt).approve(address(_uniswapV2Router),needPay) (GoldenBambooToken.sol#478)
		- _uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(needPay,0,path,_dead,block.timestamp) (GoldenBambooToken.sol#482-488)
	State variables written after the call(s):
	- _tokenTransfer(sender,address(this),feeAmount) (GoldenBambooToken.sol#412)
		- _balances[sender] = _balances[sender].sub(amount) (GoldenBambooToken.sol#527)
		- _balances[recipient] = _balances[recipient].add(amount) (GoldenBambooToken.sol#528)
	GoldenBambooToken._balances (GoldenBambooToken.sol#192) can be used in cross function reentrancies:
	- GoldenBambooToken._sellToken() (GoldenBambooToken.sol#419-451)
	- GoldenBambooToken._tokenTransfer(address,address,uint256) (GoldenBambooToken.sol#522-530)
	- GoldenBambooToken._tokenTransferBefore(address,address,uint256) (GoldenBambooToken.sol#383-415)
	- GoldenBambooToken.backstopPrice() (GoldenBambooToken.sol#373-376)
	- GoldenBambooToken.balanceOf(address) (GoldenBambooToken.sol#334-336)
	- GoldenBambooToken.constructor() (GoldenBambooToken.sol#221-261)
	- _tokenTransfer(sender,recipient,amount.sub(feeAmount)) (GoldenBambooToken.sol#414)
		- _balances[sender] = _balances[sender].sub(amount) (GoldenBambooToken.sol#527)
		- _balances[recipient] = _balances[recipient].add(amount) (GoldenBambooToken.sol#528)
	GoldenBambooToken._balances (GoldenBambooToken.sol#192) can be used in cross function reentrancies:
	- GoldenBambooToken._sellToken() (GoldenBambooToken.sol#419-451)
	- GoldenBambooToken._tokenTransfer(address,address,uint256) (GoldenBambooToken.sol#522-530)
	- GoldenBambooToken._tokenTransferBefore(address,address,uint256) (GoldenBambooToken.sol#383-415)
	- GoldenBambooToken.backstopPrice() (GoldenBambooToken.sol#373-376)
	- GoldenBambooToken.balanceOf(address) (GoldenBambooToken.sol#334-336)
	- GoldenBambooToken.constructor() (GoldenBambooToken.sol#221-261)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
GoldenBambooToken._tokenTransferBefore(address,address,uint256).feeAmount (GoldenBambooToken.sol#388) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
TokenDistributor.constructor(address) (GoldenBambooToken.sol#183-186) ignores return value by IERC20(token).approve(msg.sender,type()(uint256).max) (GoldenBambooToken.sol#185)
GoldenBambooToken._backstop() (GoldenBambooToken.sol#453-492) ignores return value by (reserves0,reserves1) = IUniswapV2Pair(_uniswapV2Pair).getReserves() (GoldenBambooToken.sol#456)
GoldenBambooToken._backstop() (GoldenBambooToken.sol#453-492) ignores return value by IERC20(_usdt).approve(address(_uniswapV2Router),needPay) (GoldenBambooToken.sol#478)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
```



```
INFO:Detectors:
GoldenBambooToken.allowance(address,address).owner (GoldenBambooToken.sol#343) shadows:
        - Ownable.owner() (GoldenBambooToken.sol#69-71) (function)
GoldenBambooToken._approve(address,address,uint256).owner (GoldenBambooToken.sol#532) shadows:
        - Ownable.owner() (GoldenBambooToken.sol#69-71) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
GoldenBambooToken.setUint(uint256,uint256) (GoldenBambooToken.sol#288-312) should emit an event for:
        - _minFee = param (GoldenBambooToken.sol#302)
        - _discountMultiple = param (GoldenBambooToken.sol#305)
        - _discountProportion = param (GoldenBambooToken.sol#308)
        - _startSwapTime = param (GoldenBambooToken.sol#310)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Ownable.changeOwner(address).newOwner (GoldenBambooToken.sol#74) lacks a zero-check on :
                - _owner = newOwner (GoldenBambooToken.sol#76)
GoldenBambooToken.setAddress(address,uint256).param (GoldenBambooToken.sol#272) lacks a zero-check on :
                - _marketing = param (GoldenBambooToken.sol#274)
                - _liquidity = param (GoldenBambooToken.sol#276)
                - _issunance = param (GoldenBambooToken.sol#282)
                - _uniswapV2Pair = param (GoldenBambooToken.sol#284)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in GoldenBambooToken.transferFrom(address,address,uint256) (GoldenBambooToken.sol#352-356):
        External calls:
        - _transfer(sender,recipient,amount) (GoldenBambooToken.sol#353)
                - _uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount,0,path,address(_tokenDistributor),block.timestamp) (GoldenBambooToken.sol#432-438)
                - IERC20(_usdt).transferFrom(address(_tokenDistributor),_marketing,profit1) (GoldenBambooToken.sol#444)
                - IERC20(_usdt).transferFrom(address(_tokenDistributor),_liquidity,profit2) (GoldenBambooToken.sol#445)
                - IERC20(_usdt).transferFrom(address(_tokenDistributor),address(this),profit.sub(profit1).sub(profit2)) (GoldenBambooToken.sol#446)
                - IERC20(_usdt).approve(address(_uniswapV2Router),needPay) (GoldenBambooToken.sol#478)
                - _uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(needPay,0,path,_dead,block.timestamp) (GoldenBambooToken.sol#482-488)
        State variables written after the call(s):
        - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (GoldenBambooToken.sol#354)
                - _allowances[owner][spender] = amount (GoldenBambooToken.sol#536)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in GoldenBambooToken._sellToken() (GoldenBambooToken.sol#419-451):
        External calls:
        - _uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount,0,path,address(_tokenDistributor),block.timestamp) (GoldenBambooToken.sol#432-438)
        - IERC20(_usdt).transferFrom(address(_tokenDistributor),_marketing,profit1) (GoldenBambooToken.sol#444)
        - IERC20(_usdt).transferFrom(address(_tokenDistributor),_liquidity,profit2) (GoldenBambooToken.sol#445)
        - IERC20(_usdt).transferFrom(address(_tokenDistributor),address(this),profit.sub(profit1).sub(profit2)) (GoldenBambooToken.sol#446)
        Event emitted after the call(s):
        - SellToken(_marketing,profit1,_liquidity,profit2,address(this),profit.sub(profit1).sub(profit2)) (GoldenBambooToken.sol#447)
Reentrancy in GoldenBambooToken._tokenTransferBefore(address,address,uint256) (GoldenBambooToken.sol#383-415):
        External calls:
        - _backstop() (GoldenBambooToken.sol#409)
                - IERC20(_usdt).approve(address(_uniswapV2Router),needPay) (GoldenBambooToken.sol#478)
                - _uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(needPay,0,path,_dead,block.timestamp) (GoldenBambooToken.sol#482-488)
INFO:Detectors:
GoldenBambooToken._tokenTransferBefore(address,address,uint256) (GoldenBambooToken.sol#383-415) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_startSwapTime <= block.timestamp,ERC20:It's not yet open time) (GoldenBambooToken.sol#394)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
GoldenBambooToken._tokenTransferBefore(address,address,uint256) (GoldenBambooToken.sol#383-415) compares to a boolean constant:
        -_uniswapV2Pair == sender && _whites[recipient][2] != true (GoldenBambooToken.sol#393)
GoldenBambooToken._tokenTransferBefore(address,address,uint256) (GoldenBambooToken.sol#383-415) compares to a boolean constant:
        -_whites[sender][1] != true && _whites[recipient][0] != true (GoldenBambooToken.sol#401)
GoldenBambooToken._tokenTransferBefore(address,address,uint256) (GoldenBambooToken.sol#383-415) compares to a boolean constant:
        -_uniswapV2Pair == recipient && _whites[sender][3] != true (GoldenBambooToken.sol#397)
GoldenBambooToken._tokenTransferBefore(address,address,uint256) (GoldenBambooToken.sol#383-415) compares to a boolean constant:
        -_sellToken() == true (GoldenBambooToken.sol#408)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality
INFO:Detectors:
SafeMath.mod(uint256,uint256) (GoldenBambooToken.sol#161-163) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (GoldenBambooToken.sol#165-168) 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 (GoldenBambooToken.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 GoldenBambooToken._whites (GoldenBambooToken.sol#194) is not in mixedCase
Variable GoldenBambooToken._uniswapV2Router (GoldenBambooToken.sol#196) is not in mixedCase
Variable GoldenBambooToken._uniswapV2Pair (GoldenBambooToken.sol#197) is not in mixedCase
Variable GoldenBambooToken._fees (GoldenBambooToken.sol#199) is not in mixedCase
Variable GoldenBambooToken._scales (GoldenBambooToken.sol#200) is not in mixedCase
Variable GoldenBambooToken._marketing (GoldenBambooToken.sol#201) is not in mixedCase
Variable GoldenBambooToken._liquidity (GoldenBambooToken.sol#202) is not in mixedCase
Variable GoldenBambooToken._issunance (GoldenBambooToken.sol#203) is not in mixedCase
Variable GoldenBambooToken._absolutePrice (GoldenBambooToken.sol#204) is not in mixedCase
Variable GoldenBambooToken._discountMultiple (GoldenBambooToken.sol#205) is not in mixedCase
Variable GoldenBambooToken._discountProportion (GoldenBambooToken.sol#206) is not in mixedCase
Variable GoldenBambooToken._minFee (GoldenBambooToken.sol#207) is not in mixedCase
Variable GoldenBambooToken._decimals (GoldenBambooToken.sol#212) is not in mixedCase
Variable GoldenBambooToken._symbol (GoldenBambooToken.sol#213) is not in mixedCase
Variable GoldenBambooToken._name (GoldenBambooToken.sol#214) is not in mixedCase
Variable GoldenBambooToken._usdt (GoldenBambooToken.sol#215) is not in mixedCase
Variable GoldenBambooToken._dead (GoldenBambooToken.sol#216) is not in mixedCase
Variable GoldenBambooToken._startSwapTime (GoldenBambooToken.sol#218) is not in mixedCase
Variable GoldenBambooToken._sellCondition (GoldenBambooToken.sol#219) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable GoldenBambooToken._targetPrice(uint256,uint256,uint256).newReserves0 (GoldenBambooToken.sol#516) is too similar to GoldenBambooToken._backstop().newReserves1 (GoldenBambooToken.sol#469)
Variable GoldenBambooToken._targetPrice(uint256,uint256,uint256).newReserves0 (GoldenBambooToken.sol#516) is too similar to GoldenBambooToken._targetPrice(uint256,uint256,uint256).newReserves1 (GoldenBambooTok
en.sol#516)
```



```
INFO:Detectors:
Variable GoldenBambooToken._targetPrice(uint256,uint256,uint256).newReserves0 (GoldenBambooToken.sol#516) is too similar to GoldenBambooToken._backstop().newReserves1 (GoldenBambooToken.sol#469)
Variable GoldenBambooToken._targetPrice(uint256,uint256,uint256).newReserves0 (GoldenBambooToken.sol#516) is too similar to GoldenBambooToken._targetPrice(uint256,uint256,uint256).newReserves1 (GoldenBambooTok
en.sol#516)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
GoldenBambooToken._absolutePrice (GoldenBambooToken.sol#204) should be immutable
GoldenBambooToken._dead (GoldenBambooToken.sol#216) should be immutable
GoldenBambooToken._decimals (GoldenBambooToken.sol#212) should be immutable
GoldenBambooToken._sellCondition (GoldenBambooToken.sol#219) should be immutable
GoldenBambooToken._tokenDistributor (GoldenBambooToken.sol#209) should be immutable
GoldenBambooToken._totalSupply (GoldenBambooToken.sol#211) should be immutable
GoldenBambooToken._uniswapV2Router (GoldenBambooToken.sol#196) should be immutable
GoldenBambooToken._usdt (GoldenBambooToken.sol#215) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:GoldenBambooToken.sol analyzed (10 contracts with 93 detectors), 59 result(s) found
```

Functional Tests

Router (PCS V2):

1- Approve (passed):
https://testnet.bscscan.com/tx/0x71f6c60f049f17002fd0a7f73960129f99199f19d6a426c8889e7c093c1036f3

2- Increase Allowance (passed):
https://testnet.bscscan.com/tx/0xcd789653d8365b5d84432eb9aa582ab6eb7a734c6b5b0700a3baa9a9af028cdc

3- Decrease Allowance (passed):
https://testnet.bscscan.com/tx/0x6f27bc1ac0591aac0babf0f77414f4dde388d204069a6f1715a84accdf6058d7

4- **Set Address** (passed):
https://testnet.bscscan.com/tx/0xd0b29a86b6940b1398a3b0a892582b6d31c7d0152a78e4dbae68f93349094e47

5- Set Uint (passed):
https://testnet.bscscan.com/tx/0x29d2efde77eedef86ea32258aed43e0d56d71fe106e85bcb1d838be388df75f7

6- Set Whites (passed):
https://testnet.bscscan.com/tx/0x70f03bdaf4613c88f6ca16709346d95ed83253a39e13caf30b6ddab64b03f3f6

Summary:
- The owner can renounce the ownership.
- The owner can transfer the ownership.
- The owner can set the whites.
-The owner can set the address.
- The owner can set the uint.

Findings:
Critical: 0
High: 0
Medium:
Low: 5
Suggestions & Optimizations: 3

Centralization – Missing Zero Address

Severity: Low
function: setWhites
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 setWhites(address[] calldata accounts, bool transIn, bool transOut, bool
buy, bool sell) external onlyOwner {
        for (uint i; i < accounts.length; i++) {
            _whites[accounts[i]][0] = transIn;
            _whites[accounts[i]][1] = transOut;
            _whites[accounts[i]][2] = buy;
            _whites[accounts[i]][3] = sell;
        }
    }
```

Suggestion:

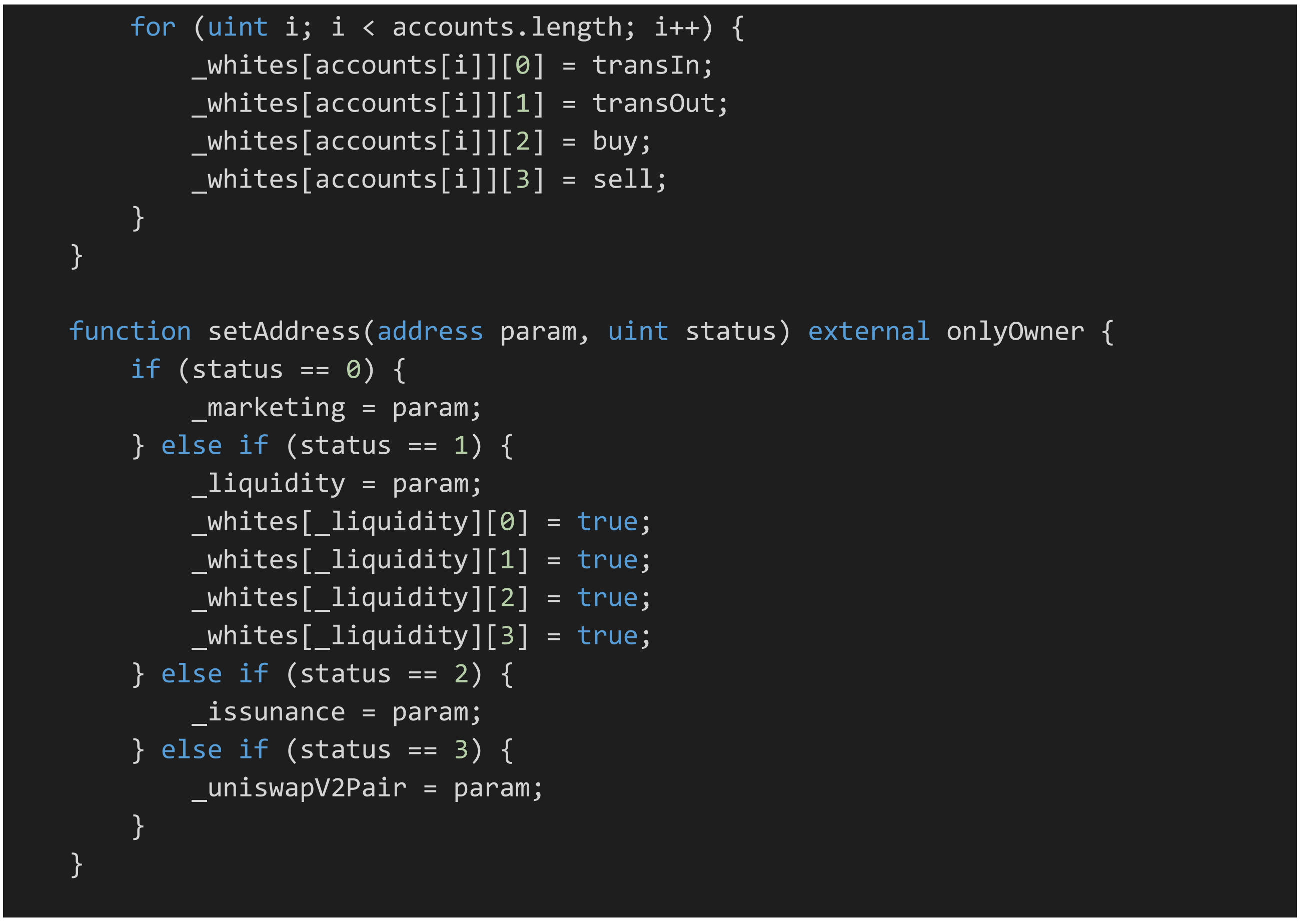
It is suggested that the address should not be zero or dead.

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 setWhites(address[] calldata accounts, bool transIn, bool transOut,
bool buy, bool sell) external onlyOwner {
```



```
        for (uint i; i < accounts.length; i++) {
            _whites[accounts[i]][0] = transIn;
            _whites[accounts[i]][1] = transOut;
            _whites[accounts[i]][2] = buy;
            _whites[accounts[i]][3] = sell;
        }
    }

    function setAddress(address param, uint status) external onlyOwner {
        if (status == 0) {
            _marketing = param;
        } else if (status == 1) {
            _liquidity = param;
            _whites[_liquidity][0] = true;
            _whites[_liquidity][1] = true;
            _whites[_liquidity][2] = true;
            _whites[_liquidity][3] = true;
        } else if (status == 2) {
            _issunance = param;
        } else if (status == 3) {
            _uniswapV2Pair = param;
        }
    }
```

Optimization

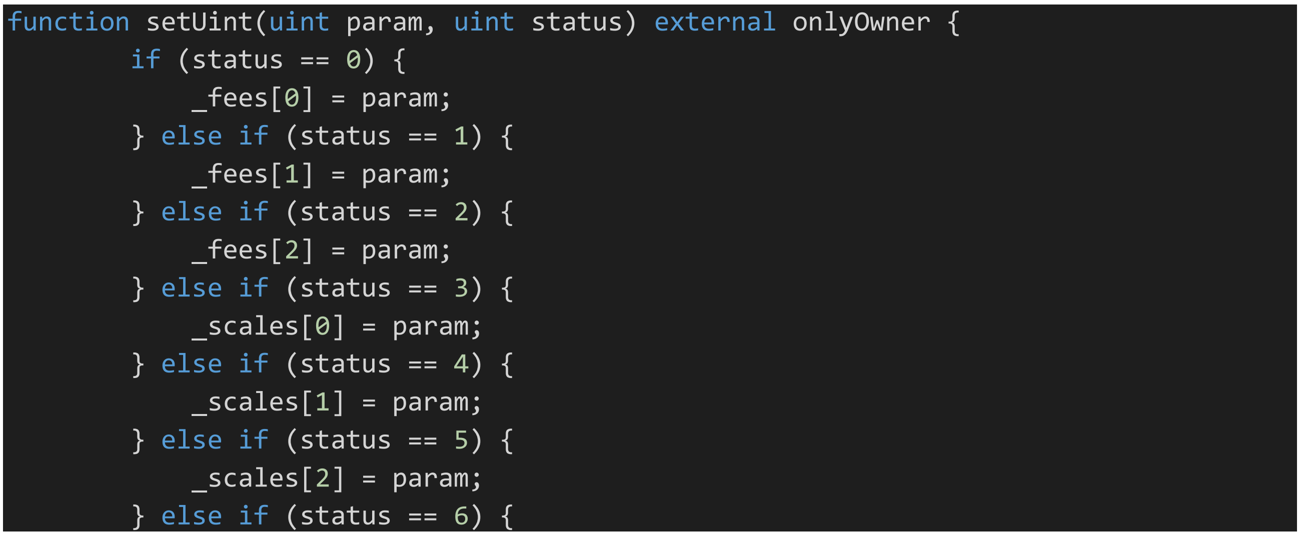
Severity: Low

subject: Missing error message

Status: Open

Overview:

Missing requires an error message.



```
function setUint(uint param, uint status) external onlyOwner {
        if (status == 0) {
            _fees[0] = param;
        } else if (status == 1) {
            _fees[1] = param;
        } else if (status == 2) {
            _fees[2] = param;
        } else if (status == 3) {
            _scales[0] = param;
        } else if (status == 4) {
            _scales[1] = param;
        } else if (status == 5) {
            _scales[2] = param;
        } else if (status == 6) {
            _minFee = param;
        } else if (status == 7) {
            require(param > 1);
            _discountMultiple = param;
        } else if (status == 8) {
            require(param < 10000);
            _discountProportion = param;
        } else if (status == 9) {
            _startSwapTime = param;
        }
    }
 function _tokenTransfer(
        address sender,
        address recipient,
        uint256 amount
    ) private {
        _balances[sender] = _balances[sender].sub(amount);
        _balances[recipient] = _balances[recipient].add(amount);
        emit Transfer(sender, recipient, amount);
    }
```

Suggestion:
It is suggested that to pass some error messages in the required check.

Centralization – Missing Visibility

Severity: Low
Subject: Missing Visibility
Status: Open
Overview:
No visibility specified

```
bool _swapping;
```

Suggestion:
You can easily silence the warning by adding the modifier public:

Centralization – Local variable Shadowing

Severity: Low

Subject: Variable Shadowing

Status: Open

Overview:

```
function allowance(address owner, address spender) external view returns (uint256)
{
        return _allowances[owner][spender];
    }
```

Suggestion:

Rename the local variables that shadow another component.

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: uint256

Status: Open

Overview:

Use uit256 instead of uint. uint is an alias for uint256 and is not recommended for use. The variable size should be clarified, as this can cause issues when encoding data with selectors if the alias is mistakenly used within the signature string.

```
function setUint(uint param, uint status) external onlyOwner {
```

Optimization

Severity: **Informational**

subject: Remove Safe Math

Status: Open

Line: 119 - 178

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.