Trend Token Audit 2 Feedback

Summary

1) “UIA” is argued as medium or minor severity
2) “MT” and “L02” errors are argued as “No change needed”
3) “FO” resolved
4) _minTrendTokenOut added to depositBNB()/deposit(), _maxTrendTokenIn added to redeem()
    - This required removal of an event and combined 2 separate require statements to not exceed contract byte size limit

Severity	Code	Description	Status	Description
Critical	UIA	Unsafe Indexing Assumption	Acknowledged	We acknowledge the risk and make sure to update appropriately, but think this is more of a “medium” or even “minor” risk severity.
Minor	MT	Mint Tokens	No change needed	This only affects supplying and redeeming supplied assets to the lending/borrowing protocol Dual Pools (fork of Venus).
Minor	FO	For-loop optimization	resolved	Applied suggested change
Minor	L02	State Variables could be Declared Constant	No change needed	State variables cannot be declared constant. All variables in storage (i.e TrendTokenStorage.sol) are changeable in other files (i.e TrendToken.sol), therefore, they cannot be declared constant
Minor	L04	Conformance to Solidity Naming Conventions	Acknowledged	
Minor	L19	Stable Compiler Version	Acknowledged	We will make sure to deploy using 0.5.16 compiler

Changes Made

1) For Loop Optimization (FO)

```
300
301     /**
302      * @notice Returns true if Trend Token is already supported
303      * @dev Is supported if added to allTrendTokens from _supportTrendToken()
304      */
305     function trendTokenSupported(ITrendToken trendToken) internal view returns(bool) {
306         for (uint i = 0; i < allTrendTokens.length; i ++) {
307             if (allTrendTokens[i] == trendToken) {
308                 return true;
309             }
310         }
311         return false;
312     }
```

2) _minTrendTokenOut Parameter added to depositBNB() and deposit()

The _minTrendTokenOut parameter allows the depositor to set the minimum amount of Trend Tokens out. This adds a layer of security incase incentiveModel contract is compromised and results in extreme fees (such as 50%).

The _minTrendTokenOut parameter is fed into depositFresh(..., _minTrendTokenOut,...) and included in a require statement as in the screenshot below

```
824    /**
825     * @notice Allows the deposit of BNB/BEP20 for Trend Tokens
826     * @dev Keeps protocolFeePerc in Pool (no longer sent to admin)
827     */
828    function depositFresh(IERC20 _depositBep20, uint _sellAmtBEP20, uint _minTrendTokenOut, address payable _referrer) internal pausedTrendToken returns(uint)  {
829
830        // Requirements
831        compTT.depositOrRedeemAllowed(address(this),_sellAmtBEP20); // above zero, unpaused, trend token active
832        IVBep20 dToken = IVBep20(compTT.returnDToken(address(_depositBep20)));
833        require(!depositsDisabled[address(_depositBep20)], "deposits disabled");  // checks deposits arent disabled
834        checkActiveToken(dToken); // checks this Trend Token is an enabled dToken
835
836        // Calculate prices, fees, and amounts
837        (uint priceToken,
838        uint trendTokenPrice,,
839        uint trendTokenAmt,
840        uint protocolFeePerc,
841        int feeOrReward) = trendTokenOutCalculations(_depositBep20, dToken, _sellAmtBEP20);
842
843        // Send fees and Trend Token to user
844        //sendPerformanceFee(mintTrendTokenAmt,trendTokenPrice); // MAYBE REMOVE FOR TAX REASONS?
845        require(trendTokenAmt >= _minTrendTokenOut,"!minOut");
846        trendToken.mint(msg.sender, trendTokenAmt);// mint and send Margin Token to Trader (after fees)
```

3) _maxTrendTokenIn added to redeem()

Similar to _minTrendTokenOut, the _maxTrendTokenIn parameter adds a layer of security in case the fee structure is compromised and more Trend Tokens are redeemed from the users wallet than expected (in the case the allowance is large).

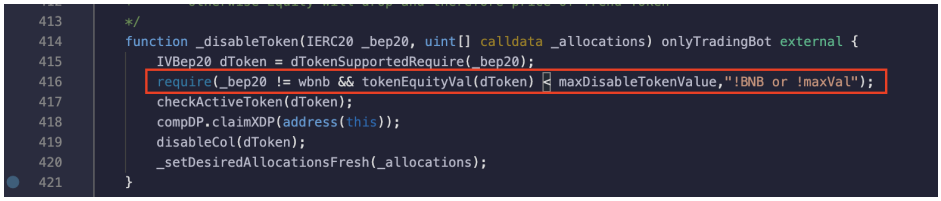
The _maxTredTokenIn parameter is fed into redeemFresh() as follows

```
936     */
937    function redeemFresh(IERC20 _redeemBep20, uint _redeemAmt, uint _maxTrendTokenIn) internal pausedTrendToken returns(uint) {
938
939        // Requirements
940        compTT.depositOrRedeemAllowed(address(this), _redeemAmt); // above zero, unpaused, trend token listed
941        IVBep20 dToken = IVBep20(compTT.returnDToken(address(_redeemBep20)));
942        checkActiveToken(dToken); // checks this Trend Token enabled dToken (must be listed)
943
944        (uint price,
945        uint trendTokenPrice,,
946        uint trendTokenInAmt,
947        uint protocolFeePerc,
948        int feeOrReward) = trendTokenInCalculations(_redeemBep20, dToken, _redeemAmt);
949
950        // Receive Trend Tokens and send Performance Fee
951        require(trendTokenInAmt <= _maxTrendTokenIn, "!maxIn");
952        trendToken.transfersFrom(msg.sender, address(this), trendTokenInAmt);
```

4) Removal of an Event and Combined Two Require Statements

The following changes were required to reduce contract byte size. Any minor/informative errors this creates will be acknowledged.

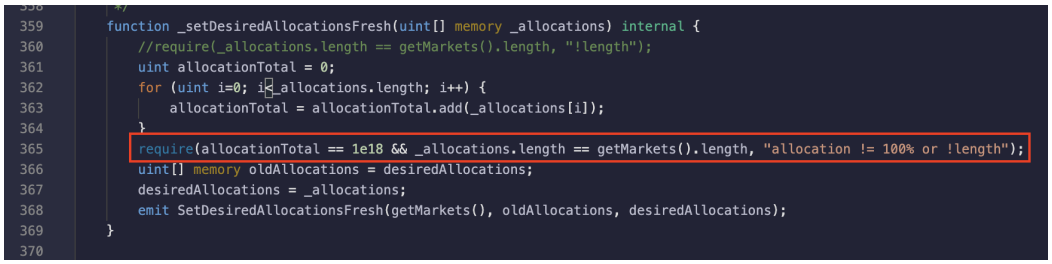
i) Combined Two Previously Separated Require Statements



```
413     */
414     function _disableToken(IERC20 _bep20, uint[] calldata _allocations) onlyTradingBot external {
415         IVBep20 dToken = dTokenSupportedRequire(_bep20);
416         require(_bep20 != wbnb && tokenEquityVal(dToken) < maxDisableTokenValue,"!BNB or !maxVal");
417         checkActiveToken(dToken);
418         compDP.claimXDP(address(this));
419         disableCol(dToken);
420         _setDesiredAllocationsFresh(_allocations);
421     }
```

*Decreased ability to debug issue will be tolerated

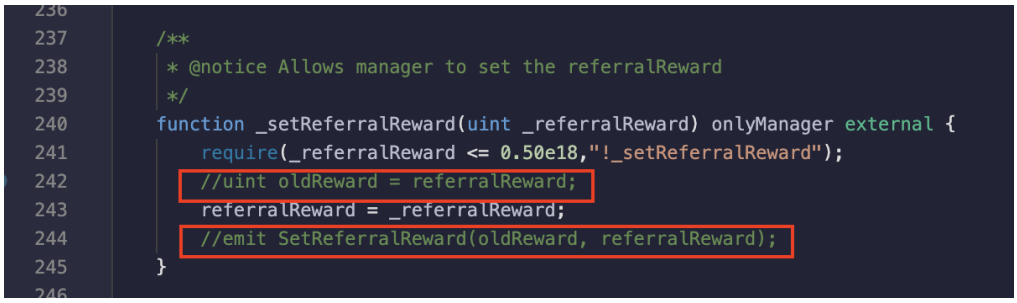
ii) Combined Two Previously Separated Require Statements



```
359     function _setDesiredAllocationsFresh(uint[] memory _allocations) internal {
360         //require(_allocations.length == getMarkets().length, "!length");
361         uint allocationTotal = 0;
362         for (uint i=0; i<_allocations.length; i++) {
363             allocationTotal = allocationTotal.add(_allocations[i]);
364         }
365         require(allocationTotal == 1e18 && _allocations.length == getMarkets().length, "allocation != 100% or !length");
366         uint[] memory oldAllocations = desiredAllocations;
367         desiredAllocations = _allocations;
368         emit SetDesiredAllocationsFresh(getMarkets(), oldAllocations, desiredAllocations);
369     }
370
```

*Decreased ability to debug issue will be tolerated

iii) Combined Two Previously Separated Require Statements



```
236
237     /**
238      * @notice Allows manager to set the referralReward
239      */
240     function _setReferralReward(uint _referralReward) onlyManager external {
241         require(_referralReward <= 0.50e18,"!_setReferralReward");
242         //uint oldReward = referralReward;
243         referralReward = _referralReward;
244         //emit SetReferralReward(oldReward, referralReward);
245     }
246
```