Secureum : https://secureum.substack.com/

CARE Project: Sushiswap / Bentobox-Startegies
Registered Email ID: zishansami102@gmail.com
Access Code: 3365586255786204
Discord Handle: nahsiz#7945
Ethereum Address (for any badges):
0x0bcd81842b22eB005BF0a55e72c971464A8a2606

Findings

1. **Title**: Unlocked pragma directive
    1.1. **Summary**: solidity version is not fixed in pragma in BaseStrategy.sol:L3
    1.2. **Details**: In BaseStrategy.sol:L3 pragma directive is not fixed. This could lead to contracts being deployed with different compiler versions than with the version they have been tested with
    1.3. **Github Permalinks**: permalink
    1.4. **Mitigation**: Choose one compiler version same as other contracts
    1.5. **Tools/Techniques**: Manual Analysis
    1.6. **Difficulty+Impact**: Low Difficulty+Low Impact
    1.7. **Checklist Numbers**: #2

2. **Title**: Multiple compiler versions used in different files.
    2.1. **Summary**: solidity versions are different in BaseStrategy.sol, AaveStartegy.sol and BentoBox.sol
    2.2. **Details**: It is best practice to keep compiler versions the same in all the files to avoid ensuring bugs and security checks of different versions.
    2.3. **Github Permalinks**: permalink permalink permalink
    2.4. **Mitigation**: Ensure single version across files
    2.5. **Tools/Techniques**: Manual Analysis
    2.6. **Difficulty+Impact**: Low Difficulty+Low Impact
    2.7. **Checklist Numbers**: #3

3. **Title**: Access control seems missing in skim() function in BaseStrategy.sol
    3.1. **Summary**: skim() at BaseStrategy.sol:L142 has no access control
    3.2. **Details**: Since skim() has no access control, any fund present in the strategy contract at any time could be invested back to the Aave protocol by anyone.
    3.3. **Github Permalinks**: permalink
    3.4. **Mitigation**: access should be only given to onlyBentoBox as is the case with withdraw() harvest() and exit()
    3.5. **Tools/Techniques**: Manual Analysis
    3.6. **Difficulty+Impact**: Low Difficulty+Medium Impact
    3.7. **Checklist Numbers**: #4

4. **Title**: skim() can still be called even after exit in BaseStrategy.sol
    4.1. **Summary**: skim() at BaseStrategy.sol:L142 has no isActive modifier present
    4.2. **Details**: The absence of the isActive modifier will allow to still send tokens to the contract and invest which is not the intended flow as explained at permalink
    4.3. **Github Permalinks**: permalink
    4.4. **Mitigation**: isActive Modifier should be enabled for skim()
    4.5. **Tools/Techniques**: Manual Analysis
    4.6. **Difficulty+Impact**: Low Difficulty+Medium Impact
    4.7. **Checklist Numbers**: #141

5. **Title**: withdraw() function in BaseStrategy.sol can withdraw more than asked
    5.1. **Summary**: if there are rewards tokens present in the strategy contract then the withdraw() function will withdraw all the tokens and not just the requested amount due to BaseStrategy.sol:L235.
    5.2. **Details**: In IStrategy.sol:L17 it is specified that “The `actualAmount` should be very close to the amount” due to rounding, but if there are some reward tokens present in the token contract, the withdraw function will withdraw all those tokens present in the contract plus the asked amount which can be a significant amount.
    5.3. **Github Permalinks**: permalink
    5.4. **Mitigation**: store the contract balance in a variable before withdrawing from aave protocol and subtract it from the contract balance afterwards and withdraw only that amount
    5.5. **Tools/Techniques**: Manual Analysis
    5.6. **Difficulty+Impact**: Medium Difficulty+Medium Impact
    5.7. **Checklist Numbers**: #179

6. **Title**: return value from exit() is not being maintained in balance in bentobox which could lead to misreport of amount added in multiple calls to exit()
    6.1. **Summary**: return value from exit() is not being maintained in balance in bentobox which could lead to misreport of amount added in multiple calls to exit()
    6.2. **Details**: exit() at BaseStrategy.sol:L240 could report wrong amoundAdded if the input balance is wrong and which is a possibility in case multiple exit() calls are made as there are no balance being maintained in bentobox after exit and there could still be token remained in the strategy and therefore (actualBalance - balance) at BaseStrategy:L240 would misreport the amountAdded
    6.3. **Github Permalinks**: permalink
    6.4. **Mitigation**: Need to maintain balance of the exited strategy to be used later for recalling the exit() in BaseStrategy.sol for full exit.
    6.5. **Tools/Techniques**: Manual Analysis
    6.6. **Difficulty+Impact**: High Difficulty+Medium Impact
    6.7. **Checklist Numbers**: #142

7. **Title**: multiple calls to setAllowedPath() parallely could lead to confusion of index
    7.1. **Summary**: No index being returned for the owner to know the index of the path being set at BaseStrategy.sol:L169.
    7.2. **Details**: So in case of multiple calls parallely to the setAllowedPath() will cause confusion of index and will have to be resolved by manually checking the indexes for all paths using getAllowedPath() function.
    7.3. **Github Permalinks**: permalink
    7.4. **Mitigation**: Recommendation is to return the index being set to let the owner know the index of the paths being set.
    7.5. **Tools/Techniques**: Manual Analysis
    7.6. **Difficulty+Impact**: Medium Difficulty+Low Impact
    7.7. **Checklist Numbers**: #142

8. **Title**: return value of low level call not checked
    8.1. **Summary**: In BaseStrategy.sol:L262, an external call is being made to an address taken from the input which might not be present
    8.2. **Details**: If there is no contract present with the call data provided, the return value will still be true and therefore will be misleading.
    8.3. **Github Permalinks**: permalink
    8.4. **Mitigation**: Recommendation is to apply require() statements on contract code size on the address greater than zero before running the call to avoid misleading output.
    8.5. **Tools/Techniques**: Manual Analysis
    8.6. **Difficulty+Impact**: Low Difficulty+High Impact
    8.7. **Checklist Numbers**: #38

9. **Title**: external calls inside loops
    9.1. **Summary**: _swap() function in BaseStrategy.sol:L312 has external swap calls in loop which could lead to out of gas error
    9.2. **Details**: due to expensive operations inside the loop, it could cause out of gas error and therefore should be handled with smaller groups
    9.3. **Github Permalinks**: permalink
    9.4. **Mitigation**: Need to use smaller swap groups to avoid out of gas error. or A separate method to do individual swaps would mitigate the risk of not being able to do full swaps from reward tokens to strategy tokens.
    9.5. **Tools/Techniques**: Manual Analysis
    9.6. **Difficulty+Impact**: Low Difficulty+Medium Impact
    9.7. **Checklist Numbers**: #42 #43

10. **Title**: ownership transfer in one step using Ownable from OZ
    10.1. **Summary**: ownership transfer in Ownable.sol from Open Zeppelin is being done in a one step process which can lead to inaccessibility if not done right in BaseStrategy.sol:L16

10.2. **Details**: Critical address change should be a two step process where the first step is to approve the ownership transfer and the second is to claim the ownership. It avoids inaccessibility in case of bad address input.
10.3. **Github Permalinks**: permalink
10.4. **Mitigation**:
10.5. **Tools/Techniques**: Manual Analysis
10.6. **Difficulty+Impact**: Low Difficulty+High Impact
10.7. **Checklist Numbers**: #50

Appendix

[Anything else here]