Tribe Turbo Audit by Team1 (imagawa)

TT-001:

Tools/Techniques: Manual
Difficulty+Impact: High

Details
slurp() isn't authenticated. The function looks pretty safe, except for vault.withdraw() in slurp. If the vault is just a lending platform then it's probably no issue, but if it's a Yearn vault for instance, withdraw() will trigger a swap for some strategies which I think opens an attack vector with sandwich attacks

Mitigation
Ensure slurp is an authenticated function and use MEV protection like flashbots when calling it.

TT-002:

Tools/Techniques: Manual
Difficulty+Impact: Medium

Details
Yearn V2 has a MAX_LOSS variable in the withdraw() method. Given ERC4626 doesn't have this argument, the MAX_LOSS will need to be hardcoded or left at the default (1bip). If that were the case and vault.withdraw() would revert if the vault suffered losses > MAX_LOSS

Mitigation
Extend the ERC4626 to accept a `slippage` or `MAX_LOSS` argument in withdraw(). The can then be used configured in slurp()

TT-003:

Tools/Techniques: Manual
Difficulty+Impact: High

Details
slurp() withdraws from to itself from the master, rather than to the master from itself.

Mitigation

Change the order of the parameters in the vault.withdraw() call on line 264 of slurp().