Secureum Workshop Day 4: Graded Assignment

Exercises in this graded assignment can be completed by defining the requested properties and verifying them by running `kontrol prove` command. The properties specified in the assignment are similar to those considered in the workshop and available in [secureum-kontrol](https://github.com/runtimeverification/secureum-kontrol). For the exercise to be considered completed, the test should pass. The first two exercises are worth 5 points each, and the remaining three are worth 10 points each. Thus, a maximum of 40 points can be earned. To complete the assignment, please submit a file that contains Solidity code for each of the exercises. The deadline for submissions is 2024-04-06 12:00 PM UTC. Please submit your solution to palina.tolmach@runtimeverification.com.

1. **Using** [**test\_totalAssets\_doesNotRevert**](https://github.com/runtimeverification/secureum-kontrol/blob/f7e0e71733c194a55ac8ede238b0745ce1f38eb0/test/ERC4626.t.sol#L31) **as reference, write a test checking the following property for the ERC4626 contract:**

***“Whoever performs the call, asset() MUST NOT revert”***

***(5 points)***

*function* test\_asset\_doesNotRevert(address caller) public {

        \_notBuiltinAddress(caller);

        vm.prank(caller);

vault.asset();

    }

1. **Write a test checking the following property for ERC4626 contract:  
   *“Whoever performs the call, underlying ERC20 token `decimals()` should be less than or equal to Vault’s `decimals()`”***

***(5 points)***

1. **Using** [**test\_totalAssets\_revertsWhenPaused**](https://github.com/runtimeverification/secureum-kontrol/blob/f7e0e71733c194a55ac8ede238b0745ce1f38eb0/test/ERC4626.t.sol#L37) **as reference, write a property that ensures that `**[**convertToShares**](https://github.com/runtimeverification/secureum-kontrol/blob/f7e0e71733c194a55ac8ede238b0745ce1f38eb0/src/tokens/ERC4626.sol#L91)**` always reverts when the contract is paused *AND* totalSupply of vault is positive. You’ll need to**

* **make `from` and *`*amount` symbolic**
* **assume `from` isn’t a built-in address**
* **assume `vault`’s `totalSupply` is positive (otherwise, it’ll be symbolic and the execution with branch)**
* **ensure the contract is paused**
* **ensure `convertToShares` is called by a symbolic `from`**
* **check that `convertToShares` always reverts**

***(10 points)***

1. **Write a `test\_transfer` test function to ensure that `Vault`’s `**[**transfer**](https://github.com/runtimeverification/secureum-kontrol/blob/aa267f23e55654a889fbc15d69c7539fb5979734/src/tokens/ERC20.sol#L73)**` function (inherited from ERC20) works correctly, i.e., that `amount` being transferred is deducted from the balance of `from` and is added to the balance of `to`. You’ll need to**

* **make `from`, `to`, and *`*amount` symbolic**
* **assume `from` and `to` aren’t built-in addresses**
* **assume `from` has enough `vault` tokens to transfer**
* **assume `from` and `to` are different addresses**
* **record `from` and `to` balances pre-`transfer`**
* **ensure `vault.transfer()` is called by a symbolic `from` with `to` and `amount` as parameters**
* **record `from` and `to` balances post-`transfer`**
* **check if the balances have been updated correctly wrt the `amount` transferred**
* **be wary of overflow checks — for the purposes of this exercise, ignore the possible overflow in `to`’s balance, as discussed with respect to** [**test\_assume\_overflow**](https://github.com/runtimeverification/secureum-kontrol/blob/f7e0e71733c194a55ac8ede238b0745ce1f38eb0/test/ERC4626.t.sol#L59)

**NOTE:To help Kontrol reason about storage mapping manipulation, please include lemmas that introduce assumptions related to** [**keccak reasoning**](https://github.com/runtimeverification/secureum-kontrol/blob/master/lemmas/keccak-lemmas.k) **and #lookup simplification by running**

***`*kontrol build --require lemmas/keccak-lemmas.k --module-import ERC4626Test:KECCAK-LEMMAS`**

***(10 points)***

1. **Write a test proving equivalence between `**[**mulWadUp**](https://github.com/runtimeverification/secureum-kontrol/blob/f7e0e71733c194a55ac8ede238b0745ce1f38eb0/test/Equivalence.t.sol#L20)**` and its Solidity implementation, similar to `**[**test\_MulWad**](https://github.com/runtimeverification/secureum-kontrol/blob/8a8f6ea8dbf89764f5389241c18d15e60b1cf875/test/Equivalence.t.sol#L18)**`. The only difference between `mulWad` and `mulWadUp` is the rounding direction — `mulWad` (that we verified in the workshop) rounds down, while `mulWadUp` rounds up.**

***(10 points)***

--smt-timeout 10000 possibly neccessary