Table of Contents

Aquarius FV Report

May - June 2025

Introduction

Greetings.
My name is Asaf, and I am happy to submit this report for Certora's Aquarius FV contest.

A month ago, I found out about Certora and its Prover, from which I dove down the rabbit hole of DeFi, Dapps, smart contracts, and various related web3 subjects.

This is my first attempt at Formal Verification, as my background is in Physics. I therefore have decided to try to write a **complete** specification file for the contract in scope, in contrast to the contest's request. I would, therefore, appreciate it if you didn't deduct points on unnecessary rules, such as my unit tests. Any feedback would be gladly accepted, so that I may do better in the future.

You might find that there is plenty of redundancy in my rules, however my goal was to make the spec files robust and resilient to changes in the code.

GitHub File Layout

I have not made many changes in the given file layout but for the following:

Fees Collector smart contract rules are in the original file located:
fees_collector/src/certora_specs/fee_collector_rules.rs.

Access Control library rules are located separately in the following location:

fees_collector/src/certora_specs/access_conrol_rules.rs

I have also added my own utils file where I implemented a few features I needed located here:

fees_collector/src/certora_specs/asaf_utils.rs

Added functionality

I first began my journey with CVL (solidity). After learning about the missing functionality in Soroban-sdk from the documentation, I've decided to implement the functionality myself.

The features I'm using are implemented in the asaf_utils.rs file.

1. *Nondet_role()* – plainly returns a nondeterministic role.
2. *Nondet_wasm()* – Plainly returns a nondeterministic BytesN<32>, in order to implement contract upgrade rules.
3. *Nondet_func()* – implemented in fees_collector_funcs and access_control_funcs modules. This function is my implementation of a parametric function in CVL. It calls a function with non-deterministically chosen values, and returns the Action, which is an Enum describing the function called.
4. *role_to_string()* – to log the role for debugging

These could have been implemented more elegantly, but I am very new to rust and the prover, and have yet to understand all the inner workings and available features.

Changes in Source Code

Changes were done in accordance with the answers received in the discord's help desk section.

1. access_control/src/lib.rs – changed visibility of storage mod to pub
2. access_control/src/roles.rs – changed visibility of the following to public
    a. fn *has_many_users()*
    b. fn *is_transfer_delayed()*
3. access_control/src/storage.rs – changed visibility of the following to pub
    a. *enum DataKey*
    b. *trait StorageTrait*
4. upgrade/src/lib.rs – changed the visibility of storage mod to pub
5. upgrade/src/storage – changed the visibility of *DataKey*

Note: In the rule names you'll see duplicates with suffix *…_fees_collector* or *…_access_control.* This was done due to the separation in those spec files. Nondet_func() only calls the respective contract's or library's functions. For each spec file.

Properties

Property	Function Name/Comment
Fees Collector Rules	
Emergency mode changed => EmergencyAdmin != None	emergency_mode_changed_emergency_admin_is_some
Emergency mode changed => Emergency admin called set emergency mode	emergency_mode_state_transition
Admin not None => Can't call init_admin()	no_init_if_admin_exists
Roles and Role Transfer Logic	
One address per role unless role.has_many_users	one_address_per_role
Role changed => apply transfer was called	role_only_changes_if_apply_transfer
Revert called => apply does nothing	cant_apply_transfer_if_revert_called
Role transfered after apply => (deadline <= blocktimestamp)	role_cant_transfer_within_deadline
Admin can transfer his role	admin_can_transfer Note: rule fails as is unreachable for Admin. See from_symbol() bug
Emergency Admin can transfer his role	emergency_admin_can_transfer Note: This rule was only written to prove that the role transferring issue is with Admin only.
Cant transfer role to None	cant_transfer_role_to_none
Role has deadline => role.is_transfer_delayed	role_has_deadline_is_transfer_delayed
Role.is_transfer_delayed => Role has a deadline	role_is_transfer_delayed_has_deadline
Transferring a role doesn't affect the other roles	To save time I separated cases of role and other role has_many_users or not: • one_role_at_a_time_both_not_has_many_ • one_role_at_a_time_transfering_role_has_many_ • one_role_at_a_time_other_role_has_many_users – Fails, fue • one_role_at_a_time_both_has_many_users
Transfer deadline changed from zero => deadline > current time	deadline_state_transition_transfer Written under the same rule
Transfer deadline changed from nonzero => deadline = 0	
Transfer Deadline changed to nonzero value => commit was called	deadline_changed_due_to_commit

AQUARIOUS FV REPORT - ASAFISH

Transfer Deadline changed to zero value => apply or revert was called	deadline_changed_due_to_revert_or_apply
Transfer Deadline > 0 => commit transfer reverts	cant_commit_before_deadline_transfer
Transfer deadline can only change zero or now()<deadline	deadline_valid_states_fees_collector
Only Admin can call transfer or upgrade functions (commit, apply, revert)	only_admin_transfers_roles_or_upgrades
Role transferred => role Must be only Admin or Emergency Admin.	only_admin_or_emergency_admin_be_transfered Note: Fails for Admin due to the from_symbol() bug. See admin_can_transfer rule
Contract address can't have a role	contract_cant_have_role Fails. See contract address bug
if future address changed => someone called commit	future_address_state_transition
Future address changed => transfer deadline changed	future_address_changed_deadline_changed_fees_collector
Future address cant become None	future_address_cant_become_none_fees_collector
Contract Upgrade Logic	
Emergency mode => Admin can apply upgrade disregarding deadline	no_upgrade_delay_if_emergency_mode
Future wasm changed => Admin called commit, and future wasm is not none	future_wasm_state_transition
future wasm changed => deadline changed	future_wasm_changed_deadline_changed
future wasm cant become None	future_wasm_cant_be_none
Upgrade deadline changed to nonzero value => commit upgrade was called	deadline_changed_due_to_commit_upgrade
Upgrade deadline changed to zero value => applyUpgrade or revertUpgrade was called	deadline_changed_due_to_revert_or_apply_upgrade
Upgrade deadline changed from 0 => current time < deadline	deadline_state_transition_upgrade
Upgrade deadline changed nonzero => deadline = 0	
Upgrade deadline != 0 => commit reverts	cant_commit_if_deadline_nonzero_upgrade

AQUARIOUS FV REPORT - ASAFISH

Unit Tests	
commit_upgrade_integrity	commit_upgrade_integrity
apply_upgrade_integrity	apply_upgrade_integrity
revert_upgrade_integrity	revert_upgrade_integrity
set_emergency_mode_integrity	set_emergency_mode_integrity
commit_transfer_ownership_integrity	commit_transfer_ownership_integrity
apply_transfer_ownership_integrity	apply_transfer_ownership_integrity
revert_transfer_ownership_integrity	revert_transfer_ownership_integrity
get_future_address_integrity_admin	get_future_address_integrity_admin Written to prove the function reverts for Admin. See from_symbol() bug
get_future_address_integrity	get_future_address_integrity_fees_collector
get_emergency_mode_integrity	get_emergency_mode_integrity

Access Control Rules	
Emergency mode changed => set_emergency_mode called	emergency_mode_state_transition_access_control
Role.has_many_user => get_role_safe reverts	role_has_many_users_get_role_safe_reverts
Role.has_many_user => get_role reverts	role_has_many_users_get_role_reverts
Role.has_many_user => set_role_address reverts	role_has_many_users_set_role_address_reverts
!role.has_many_user => set_role_addresses reverts	role_not_has_many_users_set_role_addresses_reverts
!role.has_many_user => get_role_addresses reverts	role_not_has_many_users_get_role_addresses_reverts
Role.as_symbol reverts for Admin	role_from_symbol_reverts_for_admin
Future address changed => deadline changed	future_address_changed_deadline_changed_access_control
Future address changed => commit was called	future_address_state_transition_access_control
Future address cant become none	future_address_cant_become_none_access_control
Deadline changed to nonzero => commit or put_transfer_deadline called	deadline_changed_to_nonzero_commit_or_put_transfer_deadline

AQUARIOUS FV REPORT - ASAFISH

Deadline changed to zero => revert, apply or put_transfer_deadline	deadline_changed_to_zero_revert_apply_or_put_transfer_deadline
Deadline changed => deadline > now() or deadline = 0	deadline_valid_states_access_control
Deadline != 0 => commit reverts	cant_commit_if_deadline_nonzero
Now() < deadline and role address is not none => apply reverts	cant_apply_before_deadline
Role address changed => new address == future address	role_changed_future_address_is_new_address
Role cant change to None	role_cant_change_to_none
Role changed => role is either Admin or EmergencyAdmin unless using set_role_address/es	role_changed_is_admin_or_emergency_admin
Role changed and role.has_many_users => set_role_addresses was called.	role_has_many_users_changes_due_to_set_role_addresses rule fails because of functions associated with the Vec comparison bug. Here is a passing test without the buggy functions and with vec comparison bug bypass
role.transfer delay => !role.has_many_users	role_has_transfer_delay_has_one_user
Role changed => apply was called or set_role_address/es	role_changed_due_to_apply_or_set_role
Transfering a role doesnt affect the other roles	To save time I separated cases of role and other role has_many_users or not: • one_role_at_a_time_both_not_has_many_access_control • one_role_at_a_time_transfering_role_has_many_users • one_role_at_a_time_other_role_has_many_users_access_control • one_role_at_a_time_both_has_many_users_access_control
Admin can transfer his role	admin_can_transfer_access_control Rule passes. Proving the issue is in fees_collector. See form_symbol bug for more.
Emergency Admin can transfer his role	emergency_admin_can_transfer_access_control Might be redundant, but written due to the from_symbol bug.
Revert called => apply transfer reverts	cant_apply_transfer_if_revert_called_access_control
Unit Tests	
address_has_role integrity	address_has_role_integrity Fails due to the Vec.contains bug
get_role_safe integrity	get_role_safe_integrity

get_role integrity	get_role_integrity
set_role_address integrity	set_role_address_integrity
get_role_addresses_integrity	get_role_addresses_integrity
set_role_addresses_integrity	set_role_addresses_integrity
get_emergency_mode_integrity_access_control	get_emergency_mode_integrity_access_control
require_rewards_admin_or_owner_integrity	require_rewards_admin_or_owner_integrity
require_operations_admin_or_owner_integrity	require_operations_admin_or_owner_integrity
require_pause_or_emergency_pause_admin_or_owner_integrity	require_pause_or_emergency_pause_admin_or_owner_integrity Fails due to the Vec.contains bug
require_pause_admin_or_owner_integrity	require_pause_admin_or_owner_integrity
assert_address_has_role_integrity	assert_address_has_role_integrity Fails due to the Vec.contains bug
commit_transfer_ownership_integrity_access_control	commit_transfer_ownership_integrity_access_control
apply_transfer_ownership_integrity_access_control	apply_transfer_ownership_integrity_access_control
revert_transfer_ownership_integrity_access_control	revert_transfer_ownership_integrity_access_control
get_future_address_integrity	get_future_address_integrity
put_transfer_ownership_deadline_integrity	put_transfer_ownership_deadline_integrity
get_transfer_ownership_deadline_integrity	get_transfer_ownership_deadline_integrity

Bugs Found

1. Role::from_symbol() bug

Description – When calling the function

access_control::role::Role::from_symbol(e: &Env, value: &symbol)

with the symbol of Role::Admin as a value, it reverts. I'm not sure why. The following functions are impacted by the bug:

- Fees_collector::commit_transfer_ownership
- Fees_collector::apply_transfer_ownership
- Fees_collector::revert_transfer_ownership
- Fees_collector::get_future_address

Proof can be found in the following rules:

- admin_can_transfer
- get_future_address_integrity_admin
- role_from_symbol_reverts_for_admin
- admin_can_transfer_access_control

2. Contract Address bug

Documentation says that is_auth(contract address) should revert, however, I show that it is possible to transfer or initiate the role to Admin. This bug renders the contract useless, due to lack of built-in functionality to handle such an event, nor are there preventative measures in place.

Proof can be found in the following rules:

- contract_cant_have_role

3. Vec.contains bug

As per this discord discussion help-desk post, the .contains bug has not yet been implemented for the vector type variables. Therefore, interaction with roles that have many users are compromised.

The following functions are impacted from this bug:

- access_control::access::address_has_role
- access_control::access::assert_address_has_role
- access_control::utils::require_pause_or_emergency_pause_admin_or_owner

Proof can be found in the following rules:

- address_has_role_integrity
- require_pause_or_emergency_pause_admin_or_owner_integrity
- assert_address_has_role_integrity
- role_has_many_users_changes_due_to_set_role_addresses

4. Vec comparison bug

AccessControl::get_role_addresses(&role) produces the address vector, or a new vector instance. The bug comes up when comparing two new vector instances. Instead of comparing the contents of the vectors and returning a true value to the '==' operator, it returns false.

This happens due to the *unwrap_or(Vec::new(&e))* implemented in the function.

I have implemented the rules around this bug, but this is still a bug.

Proof can be found in the following rules:

- role_has_many_users_changes_due_to_set_role_addresses – removed buggy functions from the Vec.contains bug and it still fails due to this bug
- compare_two_poissibly_empty_vectors
- compare_two_vectors_at_least_one_isnt_empty

Auditor's Notes

- In the role transferring logic, there should be a way to require approval signature of the recipient of the new role just as a precautionary.
- There is no way to set the delay constants, which might be problematic. Consider a bug in which the delay constant is a year from now, and the owner mistakenly calls init_admin with the wrong address. The owner must wait a year to transfer the role back to him, and assuming that there is no default *emergency_admin*, the contract cannot be upgraded. Only from the Admin's account.
- The Soroban-sdk doc claim *is_auth(contract address)* reverts, however, I find that not to be the case and am wondering if a deployed contract would work differently.
- I am looking forward to following CVLR improvements:
    1. better panic handling options in CVLR. Something like the @withrevert in CVL.
    2. An elegant way to create "if and only if" (⇔) assertions
    3. Better logging support
    4. Automatically producing the global states in call trace
- Having begun this contest entry quite early, I only recently became aware of the parametric function example that was added to the original repository's README. I therefore request that my work not be penalized for its lack of adaptation to this newly provided example, and that my creativity be judged independently.

Thank you for the opportunity to participate in the contest.

I'm looking forward to the feedback and results 😊

Asafish.