**Healthcare Policy Test Cases**

**1. Introduction**

The Healthcare policy is based on a health information system for an aged-care facility in rural New South Wales, Australia [1]. There are 13 roles and 37 rules in the original policy [2].

The test cases described in this document are derived from the Healthcare policy. Each test case consists of rules from the original Healthcare policy and rules or resources created by us (which are highlighted) to embed errors.

**2. Basic Test Case without Error**

**Roles:** Doctor (doctor\_1, doctor\_2, doctor\_3), Manager (manager)

Resource: OldMedicalRecords (PatinetA\_OldMedicalRecords, PatinetB\_OldMedicalRecords, PatinetC\_OldMedicalRecords), PatientPersonalInfo (PatientA\_PersonalInfo, PatientB\_PersonalInfo, PatientC\_PersonalInfo), PrivateNotes (PatientA\_PrivateNotes, PatientB\_PrivateNotes, PatientC\_PrivateNotes)

**Action:** View, Add

**Rules:**

**ManagerPolicy:**

(manager, View, PatientA\_OldMedicalRecords) ->Permit

(manager, View, PatientB\_OldMedicalRecords) ->Permit

(manager, View, PatientC\_OldMedicalRecords) ->Permit

(manager, View, PatientA\_PrivateNotes) ->Deny

(manager, View, PatientB\_PrivateNotes) ->Deny

(manager, View, PatientC\_PrivateNotes)->Deny

(manager, View, PatientA\_PersonalInfo) ->Permit

(manager, View, PatientB\_PersonalInfo) ->Permit

(manager, View, PatientC\_PersonalInfo) ->Permit

**DoctorPolicy:**

(doctor\_1, View, PatientA\_OldMedicalRecords) ->Permit

(doctor\_1, View, PatientB\_OldMedicalRecords) ->Deny

(doctor\_1, View, PatientC\_OldMedicalRecords) ->Deny

(doctor\_1, View, PatientA\_PersonalInfo) ->Deny

(doctor\_1, View, PatientB\_PersonalInfo) ->Deny

(doctor\_1, View, PatientC\_PersonalInfo) ->Deny

(doctor\_1, View, PatientA\_PrivateNotes) ->Permit

(doctor\_1, Add, PatientA\_PrivateNotes) ->Permit

(doctor\_1, Action: Any, PatientB\_PrivateNotes) ->Deny

(doctor\_1, Action: Any, PatientC\_PrivateNotes) ->Deny

(doctor\_2, View, PatientA\_OldMedicalRecords) ->Deny

(doctor\_2, View, PatientB\_OldMedicalRecords) ->Permit

(doctor\_2, View, PatientC\_OldMedicalRecords) ->Deny

(doctor\_2, View, PatientA\_PersonalInfo) ->Deny

(doctor\_2, View, PatientB\_PersonalInfo) ->Deny

(doctor\_2, View, PatientC\_PersonalInfo) ->Deny

(doctor\_2, Action: Any, PatientA\_PrivateNotes) ->Deny

(doctor\_2, View, PatientB\_PrivateNotes) ->Permit

(doctor\_2, Add, PatientB\_PrivateNotes) ->Permit

(doctor\_2, Action: Any, PatientC\_PrivateNotes) ->Deny

(doctor\_3, View, PatientA\_OldMedicalRecords) ->Deny

(doctor\_3, View, PatientB\_OldMedicalRecords) ->Deny

(doctor\_3, View, PatientC\_OldMedicalRecords) ->Permit

(doctor\_3, View, PatientA\_PersonalInfo) ->Deny

(doctor\_3, View, PatientB\_PersonalInfo) ->Deny

(doctor\_3, View, PatientC\_PersonalInfo) ->Deny

(doctor\_3, Action: Any, PatientA\_PrivateNotes) ->Deny

(doctor\_3, Action: Any, PatientB\_PrivateNotes) ->Deny

(doctor\_3, View, PatientC\_PrivateNotes) ->Permit

(doctor\_3, Add, PatientC\_PrivateNotes) ->Permit

**3. Test case with Error Type #1 <conflict>**

**Roles:** Doctor (doctor\_1, doctor\_2, doctor\_3), Manager (doctor\_3 (NOTE: this modification is used to generate conflicts in doctor\_3’s access to PatientC\_PersonalInfo and PatientC\_PrivateNotes))

**Resource**: OldMedicalRecords (PatinetA\_OldMedicalRecords, PatinetB\_OldMedicalRecords, PatinetC\_OldMedicalRecords), PatientPersonalInfo (PatientA\_PersonalInfo, PatientB\_PersonalInfo, PatientC\_PersonalInfo), PrivateNotes (PatientA\_PrivateNotes, PatientB\_PrivateNotes, PatientC\_PrivateNotes)

**Action:** View, Add

**Rules:**

**ManagerPolicy:**

(doctor\_3, View, PatientA\_OldMedicalRecords) ->Permit

(doctor\_3, View, PatientB\_OldMedicalRecords) ->Permit

(doctor\_3, View, PatientC\_OldMedicalRecords) ->Permit

(doctor\_3, View, PatientA\_PersonalInfo) ->Permit

(doctor\_3, View, PatientB\_PersonalInfo) ->Permit

(doctor\_3, View, PatientC\_PersonalInfo) ->Permit

(doctor\_3, View, PatientA\_PrivateNotes) ->Deny

(doctor\_3, View, PatientB\_PrivateNotes) ->Deny

(doctor\_3, View, PatientC\_PrivateNotes) ->Deny

**DoctorPolicy:**

(doctor\_1, View, PatientA\_OldMedicalRecords) ->Permit

(doctor\_1, View, PatientB\_OldMedicalRecords) ->Deny

(doctor\_1, View, PatientC\_OldMedicalRecords) ->Deny

(doctor\_1, View, PatientA\_PersonalInfo) ->Deny

(doctor\_1, View, PatientB\_PersonalInfo) ->Deny

(doctor\_1, View, PatientC\_PersonalInfo) ->Deny

(doctor\_1, View, PatientA\_PrivateNotes) ->Permit

(doctor\_1, Add, PatientA\_PrivateNotes) ->Permit

(doctor\_1, Action: Any, PatientB\_PrivateNotes) ->Deny

(doctor\_1, Action: Any, PatientC\_PrivateNotes) ->Deny

(doctor\_2, View, PatientA\_OldMedicalRecords) ->Deny

(doctor\_2, View, PatientB\_OldMedicalRecords) ->Permit

(doctor\_2, View, PatientC\_OldMedicalRecords) ->Deny

(doctor\_2, View, PatientA\_PersonalInfo) ->Deny

(doctor\_2, View, PatientB\_PersonalInfo) ->Deny

(doctor\_2, View, PatientC\_PersonalInfo) ->Deny

(doctor\_2, Action: Any, PatientA\_PrivateNotes) ->Deny

(doctor\_2, View, PatientB\_PrivateNotes) ->Permit

(doctor\_2, Add, PatientB\_PrivateNotes) ->Permit

(doctor\_2, Action: Any, PatientC\_PrivateNotes) ->Deny

(doctor\_3, View, PatientA\_OldMedicalRecords) ->Deny

(doctor\_3, View, PatientB\_OldMedicalRecords) ->Deny

(doctor\_3, View, PatientC\_OldMedicalRecords) ->Permit

(doctor\_3, View, PatientA\_PersonalInfo) ->Deny

(doctor\_3, View, PatientB\_PersonalInfo) ->Deny

(doctor\_3, View, PatientC\_PersonalInfo) ->Deny

(doctor\_3, Action: Any, PatientA\_PrivateNotes) ->Deny

(doctor\_3, Action: Any, PatientB\_PrivateNotes) ->Deny

(doctor\_3, View, PatientC\_PrivateNotes) ->Permit

(doctor\_3, Add, PatientC\_PrivateNotes) ->Permit

**Property used to check the error:**

(Manager = doctor\_3 & Doctor = doctor\_3) & (Action = View) & (PrivateNotes = PatientC\_PrivateNotes) -> decision = Permit

(Manager = doctor\_3 & Doctor = doctor\_3) & (Action = View) & (PatientPersonalInfor = PatientC\_PersonalInfo) -> decision = Permit

**Result on ACPT (NuSMV Verification for Merged Policies):**

(Manager = doctor\_3 & Doctor = doctor\_3) & (Action = View) & (PrivateNotes = PatientC\_PrivateNotes) -> decision = Permit in ManagerPolicy **is False**

(Manager = doctor\_3 & Doctor = doctor\_3) & (Action = View) & (PrivateNotes = PatientC\_PrivateNotes) -> decision = Permit in DoctorPolicy **isTrue**

(Manager = doctor\_3 & Doctor = doctor\_3) & (Action = View) & (PatientPersonalInfor = PatientC\_PersonalInfo) -> decision = Permit in ManagerPolicy **is True**

(Manager = doctor\_3 & Doctor = doctor\_3) & (Action = View) & (PatientPersonalInfor = PatientC\_PersonalInfo) -> decision = Permit in DoctorPolicy **is False**

**4. Test case with Error Type #2 <no object>**

**Roles:** Doctor (doctor\_1, doctor\_2, doctor\_3), Manager (manager)

Resource: OldMedicalRecords (PatinetA\_OldMedicalRecords, PatinetB\_OldMedicalRecords, PatinetC\_OldMedicalRecords, PatinetD\_OldMedicalRecords(NOTE: this resource is added to generate type#2 error, because it is not protected by any rule)), PatientPersonalInfo (PatientA\_PersonalInfo, PatientB\_PersonalInfo, PatientC\_PersonalInfo), PrivateNotes (PatientA\_PrivateNotes, PatientB\_PrivateNotes, PatientC\_PrivateNotes)

**Action:** View, Add

**Rules:**

ManagerPolicy:

(manager, View, PatientA\_OldMedicalRecords) ->Permit

(manager, View, PatientB\_OldMedicalRecords) ->Permit

(manager, View, PatientC\_OldMedicalRecords) ->Permit

(manager, View, PatientA\_PrivateNotes) ->Deny

(manager, View, PatientB\_PrivateNotes) ->Deny

(manager, View, PatientC\_PrivateNotes)->Deny

(manager, View, PatientA\_PersonalInfo) ->Permit

(manager, View, PatientB\_PersonalInfo) ->Permit

(manager, View, PatientC\_PersonalInfo) ->Permit

DoctorPolicy:

(doctor\_1, View, PatientA\_OldMedicalRecords) ->Permit

(doctor\_1, View, PatientB\_OldMedicalRecords) ->Deny

(doctor\_1, View, PatientC\_OldMedicalRecords) ->Deny

(doctor\_1, View, PatientA\_PersonalInfo) ->Deny

(doctor\_1, View, PatientB\_PersonalInfo) ->Deny

(doctor\_1, View, PatientC\_PersonalInfo) ->Deny

(doctor\_1, View, PatientA\_PrivateNotes) ->Permit

(doctor\_1, Add, PatientA\_PrivateNotes) ->Permit

(doctor\_1, Action: Any, PatientB\_PrivateNotes) ->Deny

(doctor\_1, Action: Any, PatientC\_PrivateNotes) ->Deny

(doctor\_2, View, PatientA\_OldMedicalRecords) ->Deny

(doctor\_2, View, PatientB\_OldMedicalRecords) ->Permit

(doctor\_2, View, PatientC\_OldMedicalRecords) ->Deny

(doctor\_2, View, PatientA\_PersonalInfo) ->Deny

(doctor\_2, View, PatientB\_PersonalInfo) ->Deny

(doctor\_2, View, PatientC\_PersonalInfo) ->Deny

(doctor\_2, Action: Any, PatientA\_PrivateNotes) ->Deny

(doctor\_2, View, PatientB\_PrivateNotes) ->Permit

(doctor\_2, Add, PatientB\_PrivateNotes) ->Permit

(doctor\_2, Action: Any, PatientC\_PrivateNotes) ->Deny

(doctor\_3, View, PatientA\_OldMedicalRecords) ->Deny

(doctor\_3, View, PatientB\_OldMedicalRecords) ->Deny

(doctor\_3, View, PatientC\_OldMedicalRecords) ->Permit

(doctor\_3, View, PatientA\_PersonalInfo) ->Deny

(doctor\_3, View, PatientB\_PersonalInfo) ->Deny

(doctor\_3, View, PatientC\_PersonalInfo) ->Deny

(doctor\_3, Action: Any, PatientA\_PrivateNotes) ->Deny

(doctor\_3, Action: Any, PatientB\_PrivateNotes) ->Deny

(doctor\_3, View, PatientC\_PrivateNotes) ->Permit

(doctor\_3, Add, PatientC\_PrivateNotes) ->Permit

**Property used to check the error:**

(Manager = manager) & (Action = View) & (OldMedicalRecords = PatinetD\_OldMedicalRecords) -> decision = Permit

**Result on ACPT (NuSMV Verification for Combined Policies):**

(Manager = manager) & (Action = View) & (OldMedicalRecords = PatinetD\_OldMedicalRecords) -> decision = Permit **is False**

**5. Test case with Error Type #3 <undecided>**

**Roles:** Doctor (doctor\_1, doctor\_2, doctor\_3), Manager (manager)

Resource: OldMedicalRecords (PatinetA\_OldMedicalRecords, PatinetB\_OldMedicalRecords, PatinetC\_OldMedicalRecords), PatientPersonalInfo (PatientA\_PersonalInfo, PatientB\_PersonalInfo, PatientC\_PersonalInfo), PrivateNotes (PatientA\_PrivateNotes, PatientB\_PrivateNotes, PatientC\_PrivateNotes, PatientD\_PrivateNotes (NOTE: this resource is added to generate type#3 error. It can only be accessed by doctor\_3 and manager, but no access right to it is assigned to doctor\_1 and doctor\_2))

**Action:** View, Add

**Rules:**

**ManagerPolicy:**

(manager, View, PatientA\_OldMedicalRecords) ->Permit

(manager, View, PatientB\_OldMedicalRecords) ->Permit

(manager, View, PatientC\_OldMedicalRecords) ->Permit

(manager, View, PatientA\_PrivateNotes) ->Deny

(manager, View, PatientB\_PrivateNotes) ->Deny

(manager, View, PatientC\_PrivateNotes)->Deny

(manager, View, PatientA\_PersonalInfo) ->Permit

(manager, View, PatientB\_PersonalInfo) ->Permit

(manager, View, PatientC\_PersonalInfo) ->Permit

(manager, View, PatientD\_PrivateNotes) ->Deny

**DoctorPolicy:**

(doctor\_1, View, PatientA\_OldMedicalRecords) ->Permit

(doctor\_1, View, PatientB\_OldMedicalRecords) ->Deny

(doctor\_1, View, PatientC\_OldMedicalRecords) ->Deny

(doctor\_1, View, PatientA\_PersonalInfo) ->Deny

(doctor\_1, View, PatientB\_PersonalInfo) ->Deny

(doctor\_1, View, PatientC\_PersonalInfo) ->Deny

(doctor\_1, View, PatientA\_PrivateNotes) ->Permit

(doctor\_1, Add, PatientA\_PrivateNotes) ->Permit

(doctor\_1, Action: Any, PatientB\_PrivateNotes) ->Deny

(doctor\_1, Action: Any, PatientC\_PrivateNotes) ->Deny

(doctor\_2, View, PatientA\_OldMedicalRecords) ->Deny

(doctor\_2, View, PatientB\_OldMedicalRecords) ->Permit

(doctor\_2, View, PatientC\_OldMedicalRecords) ->Deny

(doctor\_2, View, PatientA\_PersonalInfo) ->Deny

(doctor\_2, View, PatientB\_PersonalInfo) ->Deny

(doctor\_2, View, PatientC\_PersonalInfo) ->Deny

(doctor\_2, Action: Any, PatientA\_PrivateNotes) ->Deny

(doctor\_2, View, PatientB\_PrivateNotes) ->Permit

(doctor\_2, Add, PatientB\_PrivateNotes) ->Permit

(doctor\_2, Action: Any, PatientC\_PrivateNotes) ->Deny

(doctor\_3, View, PatientA\_OldMedicalRecords) ->Deny

(doctor\_3, View, PatientB\_OldMedicalRecords) ->Deny

(doctor\_3, View, PatientC\_OldMedicalRecords) ->Permit

(doctor\_3, View, PatientA\_PersonalInfo) ->Deny

(doctor\_3, View, PatientB\_PersonalInfo) ->Deny

(doctor\_3, View, PatientC\_PersonalInfo) ->Deny

(doctor\_3, Action: Any, PatientA\_PrivateNotes) ->Deny

(doctor\_3, Action: Any, PatientB\_PrivateNotes) ->Deny

(doctor\_3, View, PatientC\_PrivateNotes) ->Permit

(doctor\_3, Add, PatientC\_PrivateNotes) ->Permit

(doctor\_3, View, PatientD\_PrivateNotes) ->Permit

(doctor\_3, Add, PatientD\_PrivateNotes) ->Permit

**Property** **used to check the error**:

(Doctor = doctor\_1) & (Action = View) & (PrivateNotes = PatientD\_PrivateNotes) -> decision = Permit

**Result on ACPT (NuSMV Verification for Combined Policies):**

(Doctor = doctor\_1) & (Action = View) & (PrivateNotes = PatientD\_PrivateNotes) -> decision = Permit **is False**

**References**

[1] M. Evered and S. Bögeholz. A case study in access control requirements for a health information system. In Proc. Australasian Information Security Workshop 2004 (AISW), volume 32 of Conferences in Research and Practice in Information Technology, 2004.

[2] http://www3.cs.stonybrook.edu/~stoller/ccs2007/healthcare.txt