Continue Policy Test Cases

(InfoBeyond Technology LLC)

Abstract

This document demonstrates access control test cases using Security Policy Tool, a software tool for Access Control Security Managers, Policy Authors, and other IT Security Professionals specializing in the performance of access control systems. Access control policies are designed to protect the accessibility of online resources in networks, IoTs, healthcare systems, financial service systems, enterprise IT and clouds, military systems, and other online environments. There are several challenges in building robust access control models for these systems including (i) effectively composing secure policies and rules, (ii) testing these policies systematically, (iii) verifying these policies to prevent access control leaks. Security Policy Tool solves these issues by providing powerful access control policy modeling, testing, and verification features that empower organizations to close the door to access control leaks.

Index Terms

Access control, attribute-based access control (abac), role-based access control (rbac), security policy editing, test, verification, deployment, access control leaks, XACML, software tool.

1 INTRODUCTION TO TEST CASES

This document and linked Security Policy Tool– Project Files have been designed to help you gain an understanding of what common access control errors look like, how they are created, and how to resolve them. Organizations who leverage Security Policy Tool's systematic modeling, testing and verification features are empowered to efficiently identify errors and close the door to access control leaks.

These Continue Policy test cases are based on examples previously created by the <u>National</u> <u>Institute of Standards & Technology (NIST)</u> to demonstrate commonly found errors in access <u>control policy logic similarly.</u> This policy is used for a web-based software called "Continue" which provides tools for organizing processes required to finalize conference papers (e.g., submission, review, discussion, etc.). These test cases consist of policies/rules from NIST's example as well as modifications to better illustrate how Security Policy Tool enhances access control security. The goal of these test cases is to provide a starting point for what to expect as you go on to use Security Policy Tool to analyze your own policy verification results for errors.

2 SETTING UP THE POLICIES – TEST CASE 1 (RULE CONFLICT)

This continue example contains two policies (PCmember Policy & Reviewer Policy). The Attribute/Attribute Values included in these policies are as shown in Figure 1.

[•] Contact us at: E-mail: Info@Securitypolicytool.com

Security Policy Tool (www.Securitypolicytool.com) is a commercial version of NIST(National Institute of Standards and Technology)'s ACPT (Access Control Policy Tool). With tremendous consultation with NIST experts, Security Policy Tool substantially enhances and expands the NIST's ACPT design with advanced features for achieving high security confidence access control levels such that it can be commercialized. The development of Security Policy Tool is financially sponsored by NIST via a SBIR (Small Business Innovation Research) Phase I and II programs. It specifically improves the NIST's ACPT design to provide a robust, unified, professional, and functionally powerful access control policy tool.

		•		ContinueTestCase1.s
ntinueTestCase1.spt Attribute	^	Attribute		5 rows out of 5
Subject		Туре	Total Attribute(s)	Total Attribute Value(s)
- Reviewer ; http://www.w3.org/2001/XMLSchema#string		Subject	2	4
o pcmember_1		Resource	2	6
o pcmember_2	Subjects	Action	1	3
	- Oubjeets	Environment	0	0
o pcmember_3		Condition	1	2
- OPCmember ; http://www.w3.org/2001/XMLSchema#string				
pcmember				
Resource				
- Paper ; http://www.w3.org/2001/XMLSchema#string		Inheritance		2 rows out of 2
🙋 Paper_1				
- O Paper 2		T	/pe	No of Beneficiarie
Paper 3	Resources		nheritance	3
		Resource	Inheritance	0
⊡-© Review ; http://www.w3.org/2001/XMLSchema#string				
C Review_1				
🙋 Review_2				
@ Review_3				3 rows out of 3
Action		Access Control Model		S rows out of 5
GontinAction ; http://www.w3.org/2001/XMLSchema#string		Ту	pe	No of Policy(s)
© Read	Actions	AB	AC	2
		Multi	level	0
O MakeReview		Work	flow	0
💷 🥥 Update				
🗑 Environment 🚽	No Environme	nts		
Condition				
sSubmitted ; http://www.w3.org/2001/XMLSchema#boolean	Conditions			
	Conuluons	Security Requirement		3 rows out of 3
@ True			2	11 (14) (14)
S False			Туре	No c
nheritance			al Security Requirement binatorial Test Suite	

Fig. 1. Test Case 1

3 SUBJECT INHERITANCE – TEST CASE 1 (RULE CONFLICT)

Depending on your security needs or organizational makeup you may decide to define Inheritance relationships to help you generate policy Rules more quickly. For this continue policy example we will define (3) Subject Inheritance Relationships as follows:

Subject Inheritance:

Beneficiary Values \rightarrow Reviewer (pcmember_1, pcmember_2, pcmember_3) Inherited Values \rightarrow PCmember

If we have created these Relationships correctly based on the above direction it will look like this in Security Policy Tool:

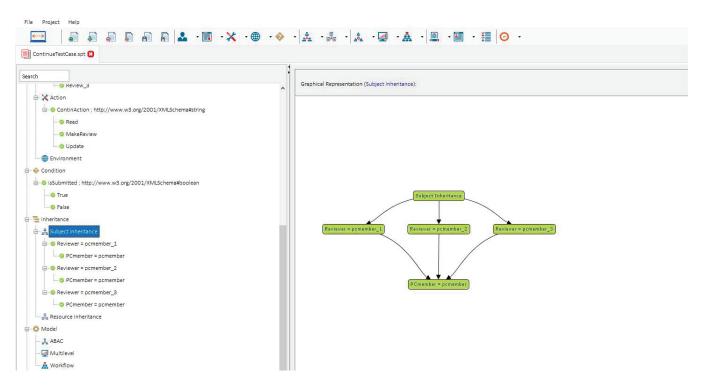


Fig. 2. Subject Inheritance Relationships

By defining these relationships, any (Originated) Rules with Decision = Permit given to PCmember = pcmember will now also be given to pcmember_1, pcmember_2, and pcmember_3 as (Inherited) Rules in our policies. Originated Rules with Decision = Deny are never Inherited. This is because typically Beneficiaries in these relationships are higher ranking/senior roles that by nature will have less restrictions (e.g., denying access) than roles that are providing the Inheritance Value (e.g., pcmember in our example).

Hence, it will authorize the Beneficiaries to obtain all privileges of Inherited Values (e.g., generally lower-level roles) while not obtaining their typically tighter restrictions. If you would like Beneficiaries to be Denied access similar to their Inherited Value you can still do so by manually creating individual rules when you begin modeling.

4 MODELING YOUR POLICY – TEST CASE 1 (RULE CONFLICT)

Now that we have entered our attributes we can model our two policies (PCmember Policy & Reviewer Policy). See the list below of the rules contained in each of these policies. You can open a "New (blank) Project" and build these policies by entering the following rules below:

PCmember Policy:

(Subject = Any Value & PCmember = pcmember, Read, Paper_1) \rightarrow Permit (Subject = Any Value & PCmember = pcmember, Read, Paper_2) \rightarrow Permit (Subject = Any Value & PCmember = pcmember, Read, Paper_3) \rightarrow Permit (Subject = Any Value & PCmember = pcmember, isSubmitted: False, Read, Review_1) \rightarrow Deny (Subject = Any Value & PCmember = pcmember, isSubmitted: False, Read, Review_2) \rightarrow Deny (Subject = Any Value & PCmember = pcmember, isSubmitted: False, Read, Review_2) \rightarrow Deny (Subject = Any Value & PCmember = pcmember, isSubmitted: False, Read, Review_3) \rightarrow Deny (Subject = You will also generate 9 additional "'Inherited Rules" due to the first 3 Permit Rules **Reviewer Policy:**

(Subject = Any Value & Reviewer = pcmember_1, isSubmitted: False, Update, Review_1) →Permit (Subject = Any Value & Reviewer = pcmember_2, isSubmitted: False, Update, Review_2) →Permit (Subject = Any Value & Reviewer = pcmember_3, isSubmitted: False, Update, Review_3) →Permit (Subject = Any Value & Reviewer = pcmember_3, isSubmitted: False, Read, Review_3) →Permit (Subject = Any Value & Reviewer = pcmember_1, isSubmitted: False, MakeReview, Paper_1) →Permit

(Subject = Any Value & Reviewer = pcmember_2, isSubmitted: False, MakeReview, Paper_2) \rightarrow Permit (Subject = Any Value & Reviewer = pcmember_3, isSubmitted: False, MakeReview, Paper_3) \rightarrow Permit

After entering the rules above your modeled policies should look like the screenshots below. If you did not create your own Project File, you can simply open Security Policy Tool – Project File: ContinueTestCase1 and these policies will have been already created for you.

member Policy Po	olicy(s) Summary			1 rows out of	1			Search 🚺
Model	Policy Name	Rule Combination	Algorithm	Policy Enforcement Algorithm	No. of Rule(s)	Time Created		Last Modified
ABAC	PCmember Policy	Deny-oven	rides	Deny Biased	15	June 14, 2018 15:54:3	4	June 14, 2018 15:54:34
ile (s) defined with	selected policy (PCmember Pc	olicy):		③ 15 ro	ws out of 15			Search
Sequence No	Subject	t	Resource	Action	Environment	Condition	Decision	Inheritance Relatio
1	Subject = Any Value & PCn	nember = pcmember	Paper = Paper_1	ContinAction = Read	Environment = Anv Value	Condition = Any Value	Permit	Originated
2	Subject = Any Value & Revi		Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited
3	Subject = Any Value & Revi		Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited
4	Subject = Any Value & Revi		Paper = Paper_1	ContinAction = Read	Environment = Anv Value	Condition = Any Value	Permit	Inherited
5	Subject = Any Value & PCn		Paper = Paper_2	ContinAction = Read	Environment = Anv Value	Condition = Any Value	Permit	Originated
6	Subject = Any Value & Revi	ewer = pcmember 1	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited
7.	Subject = Any Value & Revi	ewer = pcmember 2	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited
8	Subject = Any Value & Revi		Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited
9	Subject = Any Value & PCn		Paper = Paper_3	ContinAction = Read	Environment = Anv Value	Condition = Any Value	Permit	Originated
10	Subject = Any Value & Revi		Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited
11	Subject = Any Value & Revi	ewer = pcmember 2	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited
12	Subject = Any Value & Revi	ewer = pcmember 3	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited
13	Subject = Any Value & PCn		Review = Review_1	ContinAction = Read	Environment = Any Value	isSubmitted = False	Deny	Originated
14	Subject = Any Value & PCn	nember = pcmember	Review = Review_2	ContinAction = Read	Environment = Any Value	isSubmitted = False	Deny	Originated
15	Subject = Any Value & PCn		Review = Review 3	ContinAction = Read	Environment = Any Value	isSubmitted = False	Denv	Originated

Fig. 3. PCmember Policy

eviewer Policy Pol	licy(s) Summary			I rows out of 1				Search
Model	Policy Name	Rule Combinatio	n Algorithm	Policy Enforcement Algorithm	No. of Rule(s)	Time Created		Last Modified
ABAC	Reviewer Policy	Deny-ove	rrides	Deny Biased	7	June 14, 2018 16:01:32	2	une 14, 2018 16:01:32
de (c) defined wit	h selected policy (Device an Delice	det		7 rows of	it of 7		Č.	Search
	h selected policy (Reviewer Policy	y):		7 rows ou			L 200 200	Search
equence No	Subject		Resource	Action	Environment	Condition	Decision	Inheritance Relatio
equence No	Subject Subject = Any Value & Review	er = pcmember_1	Review = Review_1	Action ContinAction = Update	Environment Environment = Any Value	isSubmitted = False	Decision Permit	Inheritance Relatio
equence No 1 2	Subject	er = pcmember_1	Review = Review_1 Review = Review_2	Action ContinAction = Update ContinAction = Update	Environment	isSubmitted = False isSubmitted = False	Decision Permit Permit	Inheritance Relatio Originated Originated
equence No	Subject Subject = Any Value & Review Subject = Any Value & Review	er = pcmember_1	Review = Review_1	Action ContinAction = Update	Environment Environment = Any Value	isSubmitted = False	Decision Permit	Inheritance Relatio
equence No 1 2	Subject Subject = Any Value & Review Subject = Any Value & Review Subject = Any Value & Review	er = pcmember_1 er = pcmember_2	Review = Review_1 Review = Review_2	Action ContinAction = Update ContinAction = Update	Environment Environment = Any Value Environment = Any Value	isSubmitted = False isSubmitted = False	Decision Permit Permit	Inheritance Relatio Originated Originated
Sequence No 1 2 3	Subject Subject = Any Value & Review Subject = Any Value & Review Subject = Any Value & Review Subject = Any Value & Review	er = pcmember_1 er = pcmember_2 er = pcmember_3	Review = Review_1 Review = Review_2 Review = Review_3	Action ContinAction = Update ContinAction = Update ContinAction = Update	Environment Environment = Any Value Environment = Any Value Environment = Any Value	isSubmitted = False isSubmitted = False isSubmitted = False	Decision Permit Permit Permit	Inheritance Relatio Originated Originated Originated
Sequence No 1 2 3 4	Subject Subject = Any Value & Review Subject = Any Value & Review Subject = Any Value & Review Subject = Any Value & Review	rer = pcmember_1 rer = pcmember_2 rer = pcmember_3 rer = pcmember_1 rer = pcmember_2	Review = Review_1 Review = Review_2 Review = Review_3 Paper = Paper_1	Action ContinAction = Update ContinAction = Update ContinAction = Update ContinAction = MakeReview	Environment Environment = Any Value Environment = Any Value Environment = Any Value Environment = Any Value	isSubmitted = False isSubmitted = False isSubmitted = False isSubmitted = False	Decision Permit Permit Permit Permit	Inheritance Relatio Originated Originated Originated Originated

Fig. 4. Reviewer Policy

5 INDIVIDUAL SECURITY REQUIREMENTS - TEST CASE 1 (RULE CONFLICT)

The final step before analyzing these policies for errors is to create individual security requirements to use for testing. If you are building a "New (blank) Project" on your own you will enter the following security requirement below:

Individual Security Requirements:

```
(Pcmember = pcmember & Reviewer = pcmember_3) & (isSubmitted = False) & (Action = Read) & (Review = Review_3) \rightarrow decision = Permit
```

After entering the rule above your individual security requirement should look like the screenshot below. If you did not create your own Project File you can simply open Security Policy Tool

- Project File: ContinueTestCase1 and this requirement will have been already created for you.

est Case 1(s) Summary	1	1 rows	out of 1		Search	
	Access Control Security Requirement	Requi	rement Schema	No.	of Security Requirement(s)	
	Individual	т	Test Case 1		1	
ecurity Requirement (s	s) defined under selected Requirement Schema (Test Case 1):		I rows out	of 1	Search	X
ecurity Requirement (s Sequence No	s) defined under selected Requirement Schema (Test Case 1): Subject	Resource	1 rows out Action	of 1 Environment	Search	Decision

Fig. 5. Individual Security Requirements

6 POLICY VERIFICATION/ANALYZING RESULTS - TEST CASE 1 (RULE CONFLICT)

Now that we are ready to test our policies let's discuss the error we will be looking at in this first example. When policies are designed, there is potential for a "Rule Conflict" being created. A Rule Conflict occurs when two or more rules are defining opposite authorization in an access control policy.

In our example, an individual using this "'Continue"' software has a role of both PCmember and Reviewer in the system. Due to this, the individual is assigned both (PCmember: pcmember and Reviewer: pcmember_3) attribute values by the system during access evaluation. In the PCmember Policy it defines that pcmembers cannot read Review_3. However, in the Reviewer Policy it defines pcmember_3 can read Review_3 resulting in a Rule Conflict (e.g., Permitted to Read in Reviewer Policy; Denied to Read in PCmember Policy).

Next, we will run two "Single Policy" Verifications to reveal the Rule Conflict that is present in our policies. To do this, we will select PCmember Policy and Test Case 1 (security requirement) as a Single Policy Verification and also choose Reviewer Policy and Test Case 1 (security requirement) as a Single Policy Verification and analyze our two verification results. Again, this will have already been done for you if you open Project File: ContinueTestCase1.

olicy Verification	n (June 14, 2018 17:33:16)(s) Summary			1 rows out of 1			s	Search	
Status	Name	Verification Type	Verification Technique	Number of Policy(s) Combination Algori	thm Enforcement A	lgorithm	Policy List	
UpToDate	Policy Verification (June 14, 2018 17:33:16)	Standard	Single Policy	1	Deny-overrides	Deny Bia	sed	ABAC:PCmember I	Policy
esult(s) with se	lected verification (Policy Verification (June 14, 20	018 17:33:16))		@ 1 ro	ws out of 1		Se	earch	a 4
Result(s) with sel		018 17:33:16))	Resource	Action	ws out of 1 Environment	Condition	Se	earch Verification R	

Fig. 6. PCmember Policy x Test Case 1

olicy Verification (.	une 14, 2018 17:33:16)(s) Summary			I rows out of	of <u>1</u>			Search	×
Status	Name	Verification Type	Verification Techn	ique Number of Po	olicy(s) Combination Alg	orithm Enforcement	t Algorithm	Policy	List
JpToDate I	Policy Verification (June 14, 2018 17:33:16)	Standard	Single Policy	1	Deny-overrid	es Deny B	Biased	ABAC:Review	ver Polic
	ted verification (Policy Verification (June 14, 20)18 17:33:16))		۲	1 rows out of 1		1	Search	X
esult(s) with selec Requirement Schi		018 17:33:16))	Resource	Action	1 rows out of 1 Environment	Condition	Decision		_

Fig. 7. Reviewer Policy x Test Case 1

As you can see from our verification results our policies are both Permitting and Denying the individual (PCmember = pcmember/Reviewer = pcmember_3) from reading Unsubmitted resource Review_3 which is known as a Rule Conflict error.

7 RESOLVING THIS ERROR - TEST CASE 1 (RULE CONFLICT)

To solve a Rule Conflict the policy author would need to go back and either update or delete the related rules to the error. To view which specific Rules are resulting in these Verification Results we can click on our 2 Results (PCmember PolicyxTestCase1: False & Reviewer PolicyxTestCase1: True) and see which Rules have "Match Results".

See the screen shots below of our two Policy's Match Results to discover which specific rules are related to our Verification Results (e.g., False, True).

Policy Verification	n (June 14, 2018 17:33:16)(s) Summary			1 rows out	of 1			Search
Status	Name	Verification Type	Verification Techr	nique Number of Po	licy(s) Combination Al	gorithm Enf	orcement Algorithm	Policy List
UpToDate	Policy Verification (June 14, 2018 17:33:16) Standard	Single Policy	1	Deny-overr	des	Deny Biased	ABAC:PCmember Polic
Result(s) with se	lected verification (Policy Verification (June 1	4, 2018 17:33:16))		۲	1 rows out of 1			Search
Requirement S	chema Subject		Resource	Action	Environment	Conditi	on Decision	Verification Result
Test Case	1 PCmember = pcmember & Revi	ewer = pcmember 3	Review = Review 3	ContinAction = Read	Environment = Anv Value	isSubmitted	= False Permit	FALSE
	Policy Name	Bule Combin	nation Algorithm		Policy Enforcement Algorith	n	Com	bined Result
4	ABAC : PCmember Policy		-overrides		Deny Biased			Deny
		Deny						
	ABAC : PCmember Policy	Deny		Environme	Deny Biased			Deny Search
Rule(s) and Mate	ABAC : PCmember Policy	Deny ted security requirement: Resource	-overrides Action	Environme	Deny Biased	Decision	[Deny Search XII
Rule(s) and Mate	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject Subject Subject = Any Value & PCmember = pcme	Deny ted security requirement: Resource nber Paper = Paper_	-overrides Action ContinAction = I	Environment = A	Deny Blased Deny Blased Is rows out of 15 nt Condition ny Value Condition = Any	Decisio Value Permit	n Inheritance Relat	Deny Search
Rule(s) and Mate Sequence No 1	ABAC : PCmember Policy	Deny ted security requirement: Resource mber Paper = Paper_ rer_1 Paper = Paper_	-overrides Action ContinAction = ContinAction =	Environme Read Environment = A Read Environment = A	Deny Blased Deny Blased D 15 rows out of 15 nt Condition y Value Condition = Any y Value Condition = Any	Value Permit Value Permit	n Inheritance Relat Originated	Deny Search Electronic Search Not Applicabl
Rule(s) and Mate Sequence No 1 2	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject Subject = Any Value & PCmember = pcmem Subject = Any Value & Reviewer = pcmemb	Deny ted security requirement: Resource mber Paper = Paper_ rer_1 Paper = Paper_ rer_2 Paper = Paper_	Action Action ContinAction = 1 ContinAction = 1 ContinAction = 1	Environme Read Environment = A Read Environment = A Read Environment = A	Deny Blased Deny Blased Solution of 15 It Condition y Value Condition = Any yy Value Condition = Any yy Value Condition = Any	Value Permit Value Permit Value Permit Value Permit	n Inheritance Relat Originated Inherited	Search Search Not Applicabl Not Applicabl
Rule(s) and Mate Sequence No 1 2 3 4 5	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject Subject Subject = Any Value & PCmember = pcme Subject = Any Value & Reviewer = pcmem Subject = Any Value & Reviewer = pcmem	Deny ted security requirement: Resource mber Paper = Paper_ rer_1 Paper = Paper_ rer_2 Paper = Paper_ rer_3 Paper = Paper_	Action Action ContinAction = 1 ContinAction = 1 ContinAction = 1 ContinAction = 1	Environment = A Read Environment = A Read Environment = A Read Environment = A Read Environment = A	Deny Blased	Decision Value Permit Value Permit Value Permit Value Permit	n Inheritance Relat Originated Inherited	Search Result Match Result Not Applicabi Not Applicabi
Rule(s) and Mate Sequence No 1 2 3 4 5 6	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject Subject = Any Value & PCmember = pcme Subject = Any Value & Reviewer = pcmem Subject = Any Value & Reviewer = pcmem Subject = Any Value & Reviewer = pcmem	Deny ted security requirement: Resource mber Paper = Paper_ ier_1 Paper = Paper_ ier_2 Paper = Paper_ ier_3 Paper = Paper_ mber Paper = Paper_ ier_1 Paper = Paper_	Action Action ContinAction = 1	Environment = A Read Environment = A	Deny Blased	Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit	Inheritance Relat Originated Inherited Inherited Originated Inherited	Search IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
Rule(s) and Mate Sequence No 1 2 3 4 5 6 7	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject Subject = Any Value & PCmember = pcmem Subject = Any Value & Reviewer = pcmemt Subject = Any Value & Reviewer = pcmemt Subject = Any Value & Reviewer = pcmemt Subject = Any Value & Reviewer = pcmemt	Deny ted security requirement: Resource mber Paper = Paper_ ter_1 Paper = Paper_ ter_2 Paper = Paper_ ter_3 Paper = Paper_ ter_1 Paper = Paper_ ter_1 Paper = Paper_ ter_2 Paper = Paper_	Action Action ContinAction = 1 ContinAction =	Environmet = A Read Environment = A	Deny Blased ▶ 15 rows out of 15 nt Condition = Any ny Value Condition = Any ny Value Condition = Any	Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit	Inheritance Relation Originated Inherited Inherited Originated Inherited Inherited	Search IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
Rule(s) and Mate Sequence No 1 2 3 4 5 6 7 7 8	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject Subject = Any Value & PCmember = pcmem Subject = Any Value & Reviewer = pcmem Subject = Any Va	Deny ted security requirement: Resource mber Paper = Paper_ ter_1 Paper = Paper_ ter_2 Paper = Paper_ ter_3 Paper = Paper_ ter_1 Paper = Paper_ ter_1 Paper = Paper_ ter_3 Paper = Paper_ ter_3 Paper = Paper_	Action Action ContinAction = 1	Environment = A Read Environment = A	Deny Biased	Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit	Inheritance Relat Originated Inherited Inherited Originated Inherited Inherited Inherited	Deny Search Match Result Not Applicab
Rule(s) and Mate Sequence No 1 2 3 4 5 6 6 7 7 8 9	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject Subject = Any Value & PCmember = pcmem Subject = Any Value & Reviewer = pcmem Subject = Any Value & PCmember = pcme	Deny ted security requirement: Resource mber Paper = Paper_ ter_1 Paper = Paper_ ter_2 Paper = Paper_ mber Paper = Paper_ ter_1 Paper = Paper_ ter_2 Paper = Paper_ ter_3 Paper = Paper_ ter_3 Paper = Paper_ ter_3 Paper = Paper_ mber Paper = Paper_	Action Action ContinAction =	Environment = A Read Environment = A	Deny Blased ▶ 15 rows out of 15 nt Condition = Any ny Value Condition = Any ny Value Condition = Any ny Value	Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit Value Permit	n Inheritance Relat Originated Inherited Inherited Originated Inherited Inherited Inherited Inherited Originated	Deny Search Match Result Not Applicabi
Rule(s) and Mate Sequence No 1 2 3 4 5 6 7 8 9 9 10	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject = Any Value & PCmember = pcmem Subject = Any Value & Reviewer = pcmem	Deny ted security requirement: Resource mber Paper = Paper_ ter_1 Paper = Paper_ ter_2 Paper = Paper_ ter_3 Paper = Paper_ ter_1 Paper = Paper_ ter_1 Paper = Paper_ ter_3 Paper = Paper_ ter_3 Paper = Paper_ ter_1 Paper = Paper_ ter_1 Paper = Paper_ ter_1 Paper = Paper_	Action Action ContinAction = 1 ContinAct	Environment = A Read Environment = A	Deny Biased ▶ 15 rows out of 15 nt Condition = Any ny Value Condition = Any ny Value Condition = Any ny Value	Value Permit Value Permit	Inheritance Relat Originated Inherited Inherited Originated Inherited Inherited Originated Originated Inherited	Deny Search Match Result Not Applicabi
Rule(s) and Mate Sequence No 1 2 3 4 5 6 7 7 8 9 9 10 11	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject = Any Value & PCmember = pcme Subject = Any Value & Reviewer = pcmemi	Deny ted security requirement: Resource mber Paper = Paper_ rer_1 Paper = Paper_ rer_2 Paper = Paper_ rer_3 Paper = Paper_ rer_1 Paper = Paper_ rer_2 Paper = Paper_ rer_3 Paper = Paper_ rer_2 Paper = Paper_ rer_1 Paper = Paper_ rer_2 Paper = Paper_	Action Action ContinAction =1	Environment = A Read Environment = A	Deny Biased	Value Permit Value Permit	Inheritance Relation Originated Inherited Inherited Inherited Inherited Inherited Inherited Inherited Inherited Inherited Inherited	Search Match Result Mot Applicab Not Applicab
Rule(s) and Matr Sequence No 1 2 3 4 5 6 7 8 9 10 11 12	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject = Any Value & PCmember = pcmem Subject = Any Value & Reviewer = pcmem Subject = Any Value &	Deny ted security requirement: Meer Paper = Paper_ ter_1 Paper = Paper_ ter_2 Paper = Paper_ ter_3 Paper = Paper_ ter_1 Paper = Paper_ ter_2 Paper = Paper_ ter_3 Paper = Paper_ ter_3 Paper = Paper_ ter_1 Paper = Paper_ ter_1 Paper = Paper_ ter_3 Paper = Paper_	Action Action ContinAction = 1 ContinAction = 1	Environment = A Read Environment = A	Deny Biased ► 15 rows out of 15 nt Condition = Any y Value Condition = An	Value Permit Value Permit	Inheritance Relat Originated Inherited Inherited Originated Inherited Inherited Originated Inherited Inherited Inherited	Deny Search Match Result Not Applicabi
Rule(s) and Matr Sequence No 1 2 3 4 5 6 7 7 8 9 9 10 11	ABAC : PCmember Policy ching result of Selected Policy against the sele Subject = Any Value & PCmember = pcme Subject = Any Value & Reviewer = pcmemi	Deny ted security requirement: Resource mber Paper = Paper_ ier_1 Paper = Paper_ ier_2 Paper = Paper_ ier_3 Paper = Paper_ ier_1 Paper = Paper_ ier_1 Paper = Paper_ ier_2 Paper = Paper_ ier_3 Paper_ ier_3 Paper = Paper_ ier_3 Pa	Action Action ContinAction = ContinA	Environment = A Read Environment = A	Deny Biased ▶ 15 rows out of 15 nt Condition = Any ny Value Condition = Any ny Value Condition = Any ny Value	Value Permit Value Permit	Inheritance Relation Originated Inherited Inherited Inherited Inherited Inherited Inherited Inherited Inherited Inherited Inherited	Search III Match Result Not Applicabl Not Applicabl Not Applicabl Not Applicabl

Fig. 8. PCmember Policy: Match Results

WWW.SECURITYPOLICYTOOL.COM

olicy Verificatio	n (June 14, 2018 17:33:16)(s) Summary			1 rows out of	of 1				Search		×
Status	Name	Verification Type	Verification Technic	que Number of Po	olicy(s)	Combination Algorith	m Enforce	ment Algorithm		Policy Lis	st
JpToDate	Policy Verification (June 14, 2018 17:33:16)	Standard	Single Policy	1		Deny-overrides	De	eny Biased	AB	AC:Reviewer	r Polic
	elected verification (Policy Verification (June 14, 20	18 17:33:16))	2000-00000 T		1 rows out of	-	200 10000	L	Search		×
Requirement S	chema Subject		Resource	Action	Env	vironment	Condition	Decision	1	Verification	Result
Test Case	1 PCmember = pcmember & Reviewer	= pcmember_3 F	Review = Review_3	ContinAction = Read	Environm	ent = Any Value	isSubmitted = Fals	se Permit		TRUE	
		0.0000.000.00	N 9829 - 60		vs out of 1			L	Search		×
	Policy Name	Rule Combinat				ement Algorithm		Coml	bined Re		
	Policy Name ABAC : Reviewer Policy	Rule Combinat			Policy Enforce	ement Algorithm / Biased		Coml			
		Deny-ov			Policy Enforce	/ Biased		Comi	bined Re	sult	
tule(s) and Mat	ABAC : Reviewer Policy	Deny-ov			Policy Enforce Deny	/ Biased	Decision	Comi	bined Re Permit Search	sult	
	ABAC : Reviewer Policy ching result of Selected Policy against the selcted s	Deny-ov ecurity requirement:	errides	Enviro	Policy Enforce Deny	/ Biased		Com	Permit Search	sult	X Result
ule(s) and Mat	ABAC : Reviewer Policy ching result of Selected Policy against the selcted s Subject	Deny-ov ecurity requirement: Resource	Action	Environmen	Policy Enforce Deny 7 rows out	r Blased	Permit	Comi	bined Re. Permit Search	sult Match R	Result
ule(s) and Mat equence No 1	ABAC : Reviewer Policy ching result of Selected Policy against the selcted s Subject Subject = Any Value & Reviewer = pomember 1	Deny-ov ecurity requirement: Resource Review = Review_1	Action ContinAction = Up	environmen odate Environmen odate Environmen	Policy Enforce Deny 7 rows out onment tt = Any Value	v Biased	Permit Permit	Comi Comi Inheritance Rela Originated	bined Re. Permit Search ation	sult Match R Not Appli	Result
ule(s) and Mat equence No 1 2 3 4	ABAC : Reviewer Policy ching result of Selected Policy against the selcted s Subject Subject = Any Value & Reviewer = pcmember_1 Subject = Any Value & Reviewer = pcmember_2	Deny-ov ecurity requirement: Review = Review_1 Review = Review_2 Review = Review_3 Paper = Paper_1	Action U ContinAction = Up ContinAction = Up ContinAction = Make	date Environmen date Environmen date Environmen Review Environmen	Policy Enforce Deny 7 rows out onment tt = Any Value tt = Any Value	Blased of 7 Condition IsSubmitted = False IsSubmitted = False IsSubmitted = False	Permit Permit Permit Permit	Comi Comi Inheritance Rela Originated Originated Originated	bined Re. Permit Search ation	Sult Match R Not Appli Not Appli	Result icab icab
ule(s) and Mat equence No 1 2 3 4 5	ABAC : Reviewer Policy ching result of Selected Policy against the selcted s Subject Subject = Any Value & Reviewer = pcmember_1 Subject = Any Value & Reviewer = pcmember_2 Subject = Any Value & Reviewer = pcmember_3	Deny-ov ecurity requirement: Resource Review = Review_1 Review = Review_3 Paper = Paper_1 Paper = Paper_2	Action ContinAction = Up ContinAction = Up ContinAction = Up ContinAction = Make ContinAction = Make	odate Environmen date Environmen date Environmen Review Environmen Review Environmen	Policy Enforce Deny 7 rows out onment ht = Any Value ht = Any Value	Blased cof 7 Condition IsSubmitted = False IsSubmitted = False IsSubmitted = False IsSubmitted = False	Permit Permit Permit Permit Permit	Comi Comi Inheritance Reli Originated Originated Originated Originated	bined Re. Permit Search ation 	Match R Not Appli Not Appli	kesult icab icab icab
ule(s) and Mat equence No 1 2 3 4	ABAC : Reviewer Policy ching result of Selected Policy against the selcted s Subject Subject = Any Value & Reviewer = pomember_1 Subject = Any Value & Reviewer = pomember_2 Subject = Any Value & Reviewer = pomember_3 Subject = Any Value & Reviewer = pomember_1	Deny-ov ecurity requirement: Review = Review_1 Review = Review_2 Review = Review_3 Paper = Paper_1	Action U ContinAction = Up ContinAction = Up ContinAction = Make	Environmen adate Environmen eReview Environmen Review Environmen Review Environmen Review Environmen	Policy Enforce Deny Trows out annent at = Any Value at = Any Value	Blased of 7 Condition IsSubmitted = False IsSubmitted = False IsSubmitted = False	Permit Permit Permit Permit Permit Permit	Comi Comi Inheritance Rela Originated Originated Originated	bined Re. Permit Search ation	Match R Not Appli Not Appli Not Appli	Kesul icab icab icab icab

Fig. 9. Reviewer Policy: Match Results

Now that we have pinpointed our (2) Rules related to our Rule Conflict Error we can go back and make changes or possibly remove these rules. Depending on your organizational structure the policy author or access control administrator would need to decide what is the most appropriate action to take to resolve the error. There is no "right" or "wrong" solution for this, you would need to determine what is most appropriate based on your organizational needs.

For our example, if we look at all other Rules in the Reviewer Policy no other Subjects are assigned to Read any included Resources. Thus, this individual assigned both (pcmember and pcmember_3) should not be allowed to Read Review_3. Similar to the rest of the Subjects in the Reviewer Policy they can still Update and MakeReview to Review_3 because it is required for their duties. To resolve this, we will simply delete this unneeded Rule 7 which will in turn resolve the Rule Conflict.

Reviewer Policy: Delete (1) Current Rule:

e & Reviewer = pcmember_3 Review = Review_3

```
(Rule No. = 7) \rightarrow (Subject = Any Value & Reviewer = pcmember_3) \rightarrow (Action = Read) \rightarrow (isSubmitted = False) \rightarrow (Resource = Review_3) \rightarrow decision = Permit
```

ContinAction = Read

isSubmitted = False Permit Originated

Fig. 10. Reviewer Policy: Delete Rule (7)

After we "Refresh" our previous Verification Results we no longer have a Rule Conflict occurring....

olicy Verificatio	on (June 20, 2018 17:46:45)(s) Summary			1 rows out of 1				Search
Status	Name	Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	m Enforcement A	lgorithm	Policy List
UpToDate	Policy Verification (June 20, 2018 17:46:45)	Standard	Single Policy	1	Deny-overrides	Deny Bias	sed	ABAC:PCmember Policy
lesult(s) with s	elected verification (Policy Verification (June 20, 20	018 17:46:45))		1 rows	out of 1		s	Search
Result(s) with s		018 17:46:45))	Resource	Action	out of 1 Environment	Condition	S	

Fig. 11. Updated Results: PCmember Policy (No Rule Conflict)

olicy Verification	(June 20, 2018 17:46:45)(s) Summary			1 rows out c	1			Search	×
Status	Name	Verification Type	Verification Technik	ique Number of Po	licy(s) Combination Alg	orithm Enforcemer	nt Algorithm	Policy L	st
UpToDate	Policy Verification (June 20, 2018 17:46:45)	Standard	Single Policy	1	Deny-overric	les Deny I	Biased	ABAC:Reviewe	r Polic
esult(s) with sel	ected verification (Policy Verification (June 20, 20)18 17:46:45))		۲	L rows out of 1		s	Search	×
Result(s) with sel)18 17:46:45))	Resource	Action	Lrows out of 1 Environment	Condition	S	Search Verification	Result

Fig. 12. Updated Results: Reviewer Policy (No Rule Conflict)

Now the Rule Conflict is gone. Both policies are agreeing in their Access Decision = pcmember & pcmember_3 should not be Permitted to Read resource Review_3.

8 SETTING UP THE POLICIES – TEST CASE 2 (NOT PROTECTED RESOURCE)

This continue example contains two policies (PCmember Policy & Reviewer Policy). The attributes in this example have been changed slightly from previous Test Case 1. Paper has gained a new attribute valued called "'Paper_4"'.The Attribute/Attribute Values include in these policies are as shown in Figure 13.

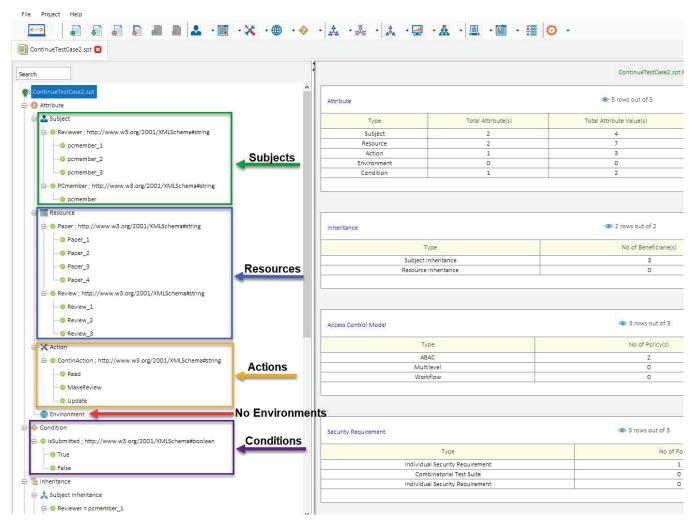


Fig. 13. Test Case 2

9 SUBJECT INHERITANCE – TEST CASE 2 (NOT PROTECTED RESOURCE)

To help us generate policy Rules more quickly we will define some Subject Inheritance relationships. Please note these are the same definitions from Test Case 1. For this continue policy example we will define (3) Subject Inheritance Relationships as follows:

Subject Inheritance:

Beneficiary Values \rightarrow Reviewer (pcmember_1, pcmember_2, pcmember_3) Inherited Values \rightarrow PCmember

If we have created these Relationships correctly based on the above direction it will look like this in Security Policy Tool:

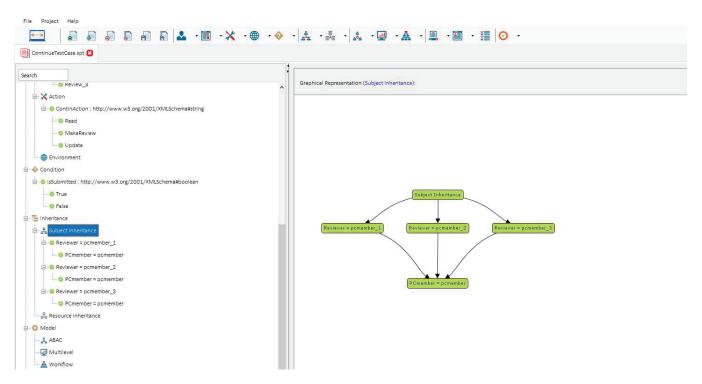


Fig. 14. Subject Inheritance Relationships

By defining these relationships, any (Originated) Rules with Decision = Permit given to PCmember = pcmember will now also be given to pcmember_1, pcmember_2, and pcmember_3 as (Inherited) Rules in our policies. Originated Rules with Decision = Deny are never Inherited. This is because typically Beneficiaries in these relationships are higher ranking/senior roles that by nature will have less restrictions (e.g., denying access) than roles that are providing the Inheritance Value (e.g., pcmember in our example).

Hence, it will authorize the Beneficiaries to obtain all privileges of Inherited Values (e.g., generally lower-level roles) while not obtaining their typically tighter restrictions. If you would like Beneficiaries to be Denied access similar to their Inherited Value you can still do so by manually creating individual rules when you begin modeling.

10 MODELING YOUR POLICY – TEST CASE 2 (NOT PROTECTED RESOURCE)

Now that we have entered our attributes we can model our two policies (PCmember Policy & Reviewer Policy). See the list below of the rules contained in each of these policies. You can open a "New (blank) Project" and build these policies by entering the following rules below:

PCmember Policy:

(PCmember = pcmember, Read, Paper_1) \rightarrow Permit (PCmember = pcmember, Read, Paper_2) \rightarrow Permit (PCmember = pcmember, Read, Paper_3) \rightarrow Permit (PCmember = pcmember, isSubmitted: False, Read, Review_1) →Deny (PCmember = pcmember, isSubmitted: False, Read, Review_2) →Deny (PCmember = pcmember, isSubmitted: False, Read, Review_3) \rightarrow Deny NOTE: You will also generate 9 additional "'Inherited Rules" due to the first 3 Permit Rules **Reviewer Policy**: (Reviewer = pcmember_1, isSubmitted: False, Update, Review_1) →Permit

- (Reviewer = pcmember_2, isSubmitted: False, Update, Review_2) →Permit
- (Reviewer = pcmember_3, isSubmitted: False, Update, Review_3) →Permit
- (Reviewer = pcmember_1, isSubmitted: False, MakeReview, Paper_1) →Permit (Reviewer = pcmember_2, isSubmitted: False, MakeReview, Paper_2) →Permit

(Reviewer = pcmember_3, isSubmitted: False, MakeReview, Paper_3) →Permit

After entering the rules above your modeled policies should look like the screenshots below. If you did not create your own Project File, you can simply open Security Policy Tool – Project File: ContinueTestCase2 and these policies will have been already created for you.

Cmember Policy Poli	cy(s) Summary		۱ (۱	rows out of 1			Search
Model	Policy Name	Rule Combination Algorithm	Policy Enforcement	Algorithm No.	of Rule(s) Time (Created	Last Modified
ABAC	PCmember Policy	Deny-overrides	Deny Biase	d	15 June 14, 20	018 15:54:34	June 14, 2018 15:54:34
le (s) defined with s	elected policy (PCmember Policy):		15 rows out of 15			Search
Sequence No	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Relation
1	PCmember = pcmember	Paper = Paper_1	ContinAction = Read	Environment = Any Val	ue Condition = Any Value	Permit	Originated
2	Reviewer = pcmember_1	Paper = Paper_1	ContinAction = Read	Environment = Any Val	ue Condition = Any Value	Permit	Inherited
3	Reviewer = pcmember_2	Paper = Paper_1	ContinAction = Read	Environment = Anv Val	ue Condition = Any Value	Permit	Inherited
4	Reviewer = pcmember_3	Paper = Paper_1	ContinAction = Read	Environment = Any Val	ue Condition = Any Value	Permit	Inherited
5	PCmember = pcmember	Paper = Paper_2	ContinAction = Read	Environment = Anv Val	ue Condition = Any Value	Permit	Originated
6	Reviewer = pcmember_1	Paper = Paper_2	ContinAction = Read	Environment = Anv Val	ue Condition = Any Value	Permit	Inherited
7	Reviewer = pcmember_2	Paper = Paper_2	ContinAction = Read	Environment = Any Val	ue Condition = Any Value	Permit	Inherited
8	Reviewer = pcmember_3	Paper = Paper_2	ContinAction = Read	Environment = Any Val	ue Condition = Any Value	Permit	Inherited
9	PCmember = pcmember	Paper = Paper_3	ContinAction = Read	Environment = Any Val	ue Condition = Any Value	Permit	Originated
10	Reviewer = pcmember_1	Paper = Paper_3	ContinAction = Read	Environment = Any Val	ue Condition = Any Value	Permit	Inherited
11	Reviewer = pcmember_2	Paper = Paper_3	ContinAction = Read	Environment = Any Val	ue Condition = Any Value	Permit	Inherited
12	Reviewer = pcmember_3	Paper = Paper_3	ContinAction = Read	Environment = Any Val			Inherited
13	PCmember = pcmember	Review = Review_1	ContinAction = Read	Environment = Any Val	ue isSubmitted = False	Deny	Originated
14	PCmember = pcmember	Review = Review_2	ContinAction = Read	Environment = Any Val	ue isSubmitted = False	Deny	Originated
15	PCmember = pcmember	Review = Review 3	ContinAction = Read	Environment = Anv Val	isSubmitted = False	Denv	Originated

Fig. 15. PCmember Policy

eviewer Policy Policy	y(s) Summary		1 rows o	ut of 1			Search
Model	Policy Name	Rule Combination Algorithm	Policy Enforcement Algori	thm No. of Rule(s)	Time Created		Last Modified
ABAC	Reviewer Policy	Deny-overrides	Deny Biased	6	June 14, 2018 16:	01:32	June 14, 2018 16:01:32
ule (s) defined with :	selected policy (Reviewer Policy):			6 rows out of 6			Search
ule (s) defined with :	selected policy (Reviewer Policy):	Resource	Action	6 rows out of 6 Environment	Condition	Decision	Search
	Subject	Resource	Action	Environment			Inheritance Relation
				Environment Environment = Any Value	Condition IsSubmitted = False IsSubmitted = False	Decision Permit Permit	
Sequence No	Subject Reviewer = pcmember_1	Resource Review = Review_1	Action ContinAction = Update	Environment Environment = Any Value Environment = Any Value	isSubmitted = False	Permit	Inheritance Relation Originated
Sequence No 1 2	Subject Reviewer = pcmember_1 Reviewer = pcmember_2	Resource Review = Review_1 Review = Review_2	Action ContinAction = Update ContinAction = Update	Environment Environment = Any Value	isSubmitted = False isSubmitted = False	Permit Permit	Inheritance Relation Originated Originated
Sequence No 1 2 3	Subject Reviewer = pcmember_1 Reviewer = pcmember_2 Reviewer = pcmember_3	Resource Review = Review_1 Review = Review_2 Review = Review_3	Action ContinAction = Update ContinAction = Update ContinAction = Update	Environment Environment = Any Value Environment = Any Value Environment = Any Value	isSubmitted = False isSubmitted = False isSubmitted = False	Permit Permit Permit	Inheritance Relation Originated Originated Originated

Fig. 16. Reviewer Policy

11 INDIVIDUAL SECURITY REQUIREMENT - TEST CASE 2 (NOT PROTECTED RE-SOURCE)

The final step before analyzing these policies for errors is to create individual security requirements to use for testing. If you are building a "New (blank) Project" on your own you will enter the following security requirement below:

Individual Security Requirement:

(Reviewer = pcmember_3) & (Action = MakeReview) & (Paper = Paper_4) \rightarrow decision = Permit

After entering the rule above your individual security requirement should look like the screenshot below. If you did not create your own Project File you can simply open Security Policy Tool – Project File: ContinueTestCase2 and this requirement will have been already created for you.

st Case 2(s) Summary			1 rows out of 1		Search	
A	ccess Control Security Requirement		Requirement Schema		No. of Security Requirement(s)	
	Individual		Test Case 2		1	
curity Requirement (s) o	defined under selected Requirement Sche	ema (Test Case 2):	۲	1 rows out of 1	Search	1
ecurity Requirement (s) o	defined under selected Requirement Schr Subject	ema (Test Case 2): Resource	Action	1 rows out of 1 Environment	Condition	Decision



12 POLICY VERIFICATION/ANALYZING RESULTS - TEST CASE 2 (NOT PROTECTED RESOURCE)

Now that we are ready to test our policies let's discuss the error we will be looking at in this second example. When policies are designed there is potential for a "Not Protected Resource" error being created by mistake. A Not Protected Resource error occurs when a resource is created but without protection from any rules.

For example, when the policy author was designing the logic for these continue policies; the author created a resource "Paper_4" with no protections. This means there are not currently any rules defined that are giving a decision for an access request to the resource. This Not Protected Resource error is not caused by any specific rules in either of our policies; it is caused due to a lack of rules created to cover this resource.

Next, we will run one "Combined Policy" Verification to reveal the Not Protected Resource error that is present in our policies. To do this, we will select Test Case 2 (security requirement) and PCmember Policy & Reviewer Policy as a Combined Policy Verification and analyze our verification result. Again, this will have already been done for you if you open Project File: ContinueTestCase2.

Policy Verific	cation (June 14, 2018 16:59:44)(s) Summary			I rows	out of 1			Search
Status	Name	Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement Algorithm		Policy List
UpToDate	Policy Verification (June 14, 2018 16:59:44)	Standard	Combined Policy	2	Deny-overrides	Deny Biased	ABAC:PCmember I	Policy, ABAC:Reviewer Polic
	th selected verification (Policy Verification (June		<u>Arti</u>	ion	1 rows out of 1	Condition		Search
Requiren	th selected verification (Policy Verification (June ment Schema Subject t Case 2 Reviewer = ocmember 3	14, 2018 16:59:44)) Resource Paper = Paper		1977.	1 rows out of 1 Environment	Condition	Decision	Search Verification Result

Fig. 18. Combined Policy x Test Case 2

By clicking on the Verification Result, we can analyze deeper the reasoning for the "False" result we have received. Here is where we will notice we have not created any Rules that are attached to Resource = Paper_4. We see this by noticing that every "Match Result" is "Not Applicable" whereas if there were Rules protecting this resource we would have seen at least one Rule with a (Permit or Deny) Match Result.

WWW.SECURITYPOLICYTOOL.COM

olicy Verific	cation (June 14, 2	018 16:59:44)(s) Summary			1 rows	out of 1			Search	×
Status		Name	Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement Algorithm		Policy List	
pToDate	Policy Verificati	on (June 14, 2018 16:59:44)	Standard	Combined Policy	2	Deny-overrides	Deny Biased	ABAC:PCmembe	er Policy, ABAC:Revie	wer Poli
										la de la
	th selected verific	ation (Policy Verification (June	14, 2018 16:59:44))			1 rows out of 1			Search	
Result(s) wit	th selected verific ment Schema	ation (Policy Verification (June Subject	14, 2018 16:59:44)) Resource	Acti	ion	I rows out of 1 Environment	Condition	Decision	Search	Result

Policy(s) and Matching result against	the selcted security requirement:	2 rows	out of 2	Search
Sequence No	Policy Name	Rule Combination Algorithm	Policy Enforcement Algorithm	Combined Result
1	ABAC : PCmember Policy	Deny-overrides	Deny Biased	Deny
2	ABAC : Reviewer Policy	Deny-overrides	Deny Biased	Deny

ule(s) and Matchi	ing result of Selected Policy again	ist the selcted security requ	irement:	۲	15 rows out of 15		Sear	ch 🚺
Sequence No	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Relation	Match Result
1	PCmember = pcmember	Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
2	Reviewer = pcmember_1	Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
3	Reviewer = pcmember_2	Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
4	Reviewer = pcmember_3	Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
5	PCmember = pcmember	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicabl
6	Reviewer = pcmember_1	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
7	Reviewer = pcmember_2	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
8	Reviewer = pcmember_3	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
9	PCmember = pcmember	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
10	Reviewer = pcmember_1	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
11	Reviewer = pcmember_2	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
12	Reviewer = pcmember_3	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicabl
13	PCmember = pcmember	Review = Review_1	ContinAction = Read	Environment = Any Value	isSubmitted = False	Deny	Originated	Not Applicabl
14	PCmember = pcmember	Review = Review_2	ContinAction = Read	Environment = Any Value	isSubmitted = False	Deny	Originated	Not Applicabl
15	PCmember = pcmember	Review = Review_3	ContinAction = Read	Environment = Any Value	isSubmitted = False	Deny	Originated	Not Applicabl

Fig. 19. PCmember Policy: Match Results

olicy Verification	(June 14, 2018 16:59:44)(s) Sum	mary		1 rows	out of 1			Search	×	1
Status	Name	Verification	Type Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement Algorithm	n	Policy List		
pToDate Poli	cy Verification (June 14, 2018 16	:59:44) Standar	d Combined Policy	2	Deny-overrides	Deny Biased	ABAC:PCmember	Policy, ABAC:Revi	awer Po	oli
Result(s) with sele	ected verification (Policy Verificati	on (June 14, 2018 16:5	9:44))		1 rows out of 1			Search	×	
Requirement So	chema Subject	R	source Ac	ion	Environment	Condition	Decision	Verification	n Result	t
Test Case :	2 Reviewer = pcme	mber 3 Paper	= Paper 4 ContinAction	= MakeReview	ovironment = Any Value	Condition = Any Valu	e Permit	FALS		-
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	ching result against the selcted se	101 104 14CM			2 rows out of 2	2 757 757	[	Search	×	
olicy(s) and Mate		curity requirement: Policy Name	Rule Co	mbination Algorithm		nforcement Algorithm.	[	Search Combined Result	-	
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	nce No	101 104 14CM				nforcement Algorithm Deny Biased	[	N 1455 M 0 44	-	
Sequer	nce No	Policy Name	r E	mbination Algorithm			[Combined Result	1	
Sequer 1 2 ule(s) and Match	nce No	Policy Name ABAC : PCmember Policy ABAC : Reviewer Policy nst the selcted security i	equirement:	mbination Algorithm eny-overrides eny-overrides	Policy E	Deny Biased	[Combined Result Deny Deny Search		
Sequen 1 2 ule(s) and Match Sequence No	ning result of Selected Policy again	Policy Name ABAC : PCmember Policy ABAC : Reviewer Policy nst the selcted security (Resource	equirement:	mbination Algorithm eny-overrides	Policy E One of the condition	Deny Biased Deny Biased Deny Biased Decision	Inheritance Relatio	Combined Result Deny Deny Search		
Sequent 1 2 ule(s) and Match Sequence No 1	ning result of Selected Policy again Subject Reviewer = pcmember_1	Policy Name ABAC : PCmember Policy ABAC : Reviewer Policy not the selcted security (Resource Review = Review_1	equirement: Action ContinAction = Update	mbination Algorithm eny-overrides eny-overrides Environmert Environment = An	Policy E Policy E Submitted =	Deny Blased Deny Blased Decision False Permit	Inheritance Relatio Originated	Combined Result Deny Deny Search Search Match Not App	Image: Second	
Sequen 1 2 ule(s) and Match Sequence No 1 2	ning result of Selected Policy again Subject Reviewer = pcmember_1 Reviewer = pcmember_2	Policy Name ABAC : PCmember Policy ABAC : Reviewer Policy nst the selcted security in Resource Review = Review_1 Review = Review_2	equirement:	mbination Algorithm env-overrides env-overrides Environment = An Environment = An	Policy E Policy E Condition Y Value IsSubmitted = y Value	Deny Biased Deny Biased Deny Biased Decision False Permit False Permit	Inheritance Relatio Originated Originated	Combined Result Deny Deny Search Search Not App Not App	Result Dicabl	
Sequen 1 2 ule(s) and Match Sequence No 1 2 3	ning result of Selected Policy again Subject Reviewer = pcmember_1 Reviewer = pcmember_2 Reviewer = pcmember_3	Policy Name BBAC : POmember Policy ABAC : Reviewer Policy ast the selcted security in Resource Review = Review_1 Review = Review_2 Review = Review_3	equirement:	mbination Algorithm env-overrides env-overrides Environment = An Environment = An Environment = An	Policy E Policy E Policy E Condition Volue isSubmitted = vValue isSubmitted =	Deny Biased Deny Biased Decision False Permit False Permit False Permit	Inheritance Relatio Originated Originated Originated	Combined Result Deny Deny Search Not App Not App Not App	Result plicabl plicabl	
Sequer 1 2 tule(s) and Match Sequence No 1 2	ning result of Selected Policy again Subject Reviewer = pcmember_1 Reviewer = pcmember_2	Policy Name ABAC : PCmember Policy ABAC : Reviewer Policy nst the selcted security in Resource Review = Review_1 Review = Review_2	equirement:	mbination Algorithm eny-overrides eny-overrides Environmert = An Environment = An Environment = An Environment = An	Policy E Policy E Policy E Submitted = yValue isSubmitted = yValue isSubmitted = yValue isSubmitted =	Deny Biased Deny Biased Decision Palse Permit False Permit False Permit False Permit	Inheritance Relatio Originated Originated	Combined Result Deny Deny Search Search Not App Not App	Result Result Dicable Dicable Dicable	

Fig. 20. Reviewer Policy: Match Results

13 RESOLVING THIS ERROR - TEST CASE 2 (NOT PROTECTED RESOURCE)

To eliminate a Not Protected Resource vulnerability the policy author would need to define a specific rule for the unprotected resource (Paper_4) and then test again to verify the intended

access decision is being made based on the new rule's design.

For example, if we're to add this new Rule (7) below to the Reviewer Policy...

Reviewer Policy: Add (1) New Rule:

 $(Rule No. = 7) \rightarrow (Reviewer = pcmember_3) \rightarrow (Action = MakeReview) \rightarrow (Resource = Paper_4) \rightarrow decision = Permit$

7	Reviewer = pcmember_3	Paper = Paper_4	ContinAction = MakeReview	Environment = Any Value	Condition = Any Value	Permit	Originated

Fig. 21. Reviewer Policy: New Rule (7)

Then retest using the same Policy Verification selections as last time we will get the same False Verification result due to how we defined our Combination Algorithms and Enforcement Algorithms. However, we can see in the Match Results that we have provided a rule for the system to evaluate for pcmember_3 accessing this resource.

Result(s) with sele	cted verification (Policy Verifica	ation (June 20, 2018 18:30	:51))	In	ows out of 1			Search	×	
Requirement So	hema Subjec	t Res	ource Actio	on Env	rironment	Condition	Decision	Verificati	on Result	t
Test Case 2	2 Reviewer = pcn	nember_3 Paper =	Paper_4 ContinAction =	MakeReview Environm	ent = Anv Value Co	ndition = Any Value	Permit	FAL	LSE	
Policy(s) and Mate	ching result against the selcted	security requirement:		🐼 2 rows o	ut of 2			Search	X	ķ
Sequen	ce No	Policy Name	Rule Com	bination Algorithm	Policy Enforcer	nent Algorithm		Combined Resul	lt	
1		ABAC : PCmember Policy	Der	ny-overrides	Deny I	Biased		Deny		-
2		ABAC : Reviewer Policy	Der	ny-overrides	Deny I	Biased		Permit		
ule(s) and Match	ing result of Selected Policy ag	ainst the selcted security re	quirement:	۲	7 rows out of 7			Search	×	1
Sequence No	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Relation	on Mate	ch Result	
1	Reviewer = pcmember_1	Review = Review_1	ContinAction = Update	Environment = Anv Value	isSubmitted = False	Permit	Originated	Not Ar	pplicable	le
2	Reviewer = pcmember_2	Review = Review_2	ContinAction = Update	Environment = Any Value	isSubmitted = False	Permit	Originated		pplicable	-
3	Reviewer = pcmember_3	Review = Review_3	ContinAction = Update	Environment = Any Value	isSubmitted = False	Permit	Originated	Not Ar	pplicable	le
4	Reviewer = pcmember_1	Paper = Paper_1	ContinAction = MakeReview	Environment = Any Value	isSubmitted = False	Permit	Originated	Not Ar	pplicable	1
5	Reviewer = pcmember_2	Paper = Paper_2	ContinAction = MakeReview	Environment = Any Value	isSubmitted = False	Permit	Originated	Not Ap	pplicable	1
5							O 1 1 1 1			
5	Reviewer = pcmember_3	Paper = Paper_3	ContinAction = MakeReview	Environment = Any Value	isSubmitted = False	Permit	Originated	Not Ap	pplicabl	ļ

Fig. 22. Updated Policy: Resource Now Protected

14 SETTING UP THE POLICIES – TEST CASE 3 (UNDECIDED RULE)

This continue example contains two policies (PCmember Policy & Reviewer Policy). The attributes in this example have been changed slightly from previous Test Case 2. Paper no longer has attribute value "Paper_4" and now Review has gained attribute value "Review_4". The Attribute/Attribute Values included in these policies are as shown in Figure 23.

Attribute Subject Subject Procementer_1 procementer_2 procementer_3 procementer_3 procementer_3 procementer_3 procementer_3 procementer_3 procementer_3 procementer_3 procementer_3 Procert, http://www.w3.org/2001/XMLSchema#string Procer_1 Procer_2 Procer_3 Procer_4 Procer_3 Procer_4 Procer_4 Procer_5 Procer_4 Procer_4 Procer_4 Procer_5 Procer_4 Procer_4 Procer_5 Procer_4 Procer_4 <		•		ContinueTestCase3.s
Reviewer; http://www.w3.org/2001/XMLSchema#string prmember; Subject Subj		Attribute		5 rows out of 5
Bource Control Action Control Model Subjects Subj	1	Туре	Total Attribute(s)	Total Attribute Value(s)
Action 1 pcmember_3 Subjects Pcmember_3 Condition pcmember 0 Paper_3 Paper_3 Review; http://www.w3.org/2001/XMLSchema#string Review; http://www.w3.org/2001/XMLSchema#string Review; A Review, 3 Review; A Actions Action Actions Action Multivel Review, 4 No Environments	/ww.w3.org/2001/XMLSchema#string	Subject	2	4
Action 1 prmember_3 Condition prmember_3 Condition prmember Condition Resource Paper_1 Paper_2 Paper_3 Review_1 Review_3 Review_4 Action Action Inheritance Type Action Actions Actions </td <td></td> <td>Resource</td> <td>2</td> <td>7</td>		Resource	2	7
Promember_3 Promember_3 Promember_ihttp://www.w3.org/2001/XMLSchema#string Paper_ihttp://www.w3.org/2001/XMLSchema#string Paper_3 Paper_3 Paper_3 Review_1 Review_1 Review_2 Review_3 Review_4 Actions Actions Actions Actions Multievel Multievel Multievel Multievel Workflow Workflow Workflow	Subject	Action	1	3
PCmember; http://www.w3.org/2001/XMLSchems#string Paper; http://www.w3.org/2001/XMLSchems#string Paper_1 Paper_2 Paper_3 Paper_3 Review_1 Review_1 Review_2 Review_2 Review_4 Action ContinAction ; http://www.w3.org/2001/XMLSchems#string Review_4 Actions Actions Multilevel Multilevel Workflow Workflow No Environments	- Subject	Environment		0
Resource Paper; http://www.w3.org/2001/XMLSchema#string Paper_1 Paper_2 Paper_3 Review ; http://www.w3.org/2001/XMLSchema#string Review ; http://www.w3.org/2001/XMLSchema#string Review_1 Review_2 Review_3 Review_4 Actions Actions MakeReview Update No Environments		Condition	1	2
Resource Paper; http://www.w3.org/2001/XMLSchema#string Paper_3 Review; http://www.w3.org/2001/XMLSchema#string Review; http://www.w3.org/2001/XMLSchema#string Review_1 Review_2 Review_3 Review_4 Actions Actions Actions Multilevel Multile	/www.w3.org/2001/XMLSchema#string			
Paper; http://www.w3.org/2001/XMLSchema#string Paper; http://www.w3.org/2001/XMLSchema#string Paper; http://www.w3.org/2001/XMLSchema#string Review; http://www.w3.org/2001/XMLSchema#string Review; A Re				
Paper_1 Paper_2 Paper_3 Review_1 Review_2 Review_2 Review_4 ContinAction ; http://www.w3.org/2001/XMLSchema#string Review_4 Action Review_4 Actions Actions Actions Actions No Environment No Environments				
Paper_2 Paper_3 Review_1 Review_2 Review_3 Review_4 ContinAction ; http://www.w3.org/2001/XM/LSchema#string Review_4 Access Control Model Access Control Model Type Access Control Model Type MakeReview Workflow No Environments	v.w3.org/2001/XMLSchema#string	Inheritance		2 rows out of 2
Paper_2 Paper_3 Review ; http://www.w3.org/2001/XMLSchema#string Review_1 Review_2 Review_4 Review_4 ContinAction ; http://www.w3.org/2001/XMLSchema#string Review_4 Actions Review_4 Actions Review_4 MakeReview Update No Environments				
Subject Inheritance Subject Inheritance Review_1 Review_2 Review_3 Review_4 ContinAction ; http://www.w3.org/2001/XM/LSchema#string Review_4 Actions Actions Actions Multilevel Mo Environments No Environments		Ту	pe	No of Beneficiarie
Review ; http://www.w3.org/2001/XMLSchema#string Review_1 Review_2 Review_3 Review_4 Action ContinAction ; http://www.w3.org/2001/XMLSchema#string Review_4 Actions Actions Actions Multilevel Multilevel Workflow Mo Environments	Bassura	Subject In	heritance	3
Review_1 Review_2 Review_3 Review_4 Action ContinAction ; http://www.w3.org/2001/XMLSchema#string Read MakeReview Update No Environments		Resource In	nheritance	0
Review_2 Review_4 Action ContinAction ; http://www.w3.org/2001/XMLSchems#string Read MakeReview Update Environment No Environments	w.w3.org/2001/XMLSchema#string			
Access Control Model Access Contro				
Access Control Model Access Contro				
Action Control.Action; http://www.w3.org/2001/XMLSchema#string Control.Action; http://www.w3.org/2001/XMLSchema#string Control.MakeReview Workflow No Environments				
Action ContinAction ; http://www.w3.org/2001/XMLSchems#string ContinAction ; http://www.w3.org/2001/XMLSchems#string Read ContinAction ; http://www.w3.org/2001/XMLSchems#string		Access Control Model		3 rows out of 3
Actions ContinAction ; http://www.w3.org/2001/XMLScheme#string Read Multilevel ModeReview Update Environment No Environments		Typ	e	No of Policy(s)
Contraction i http://www.wo.org/2001/xMitscheme#string Actions Multilevel Workflow Dupdate No Environments	and a second s			2
	o://www.w3.org/2001/XMLSchema#string			0
MakeReview Update No Environments	Actions			0
Environment No Environments	-		1999/04/2	
Dervironment No Environments				
andition	ALC: Provide			
andition Security Requirement		onments		
		Security Requirement		3 rows out of 3
IsSubmitted ; http://www.w3.org/2001/XMLSchema#boolean Conditions	ww.w3.org/2001/XMLSchema#boolean Conditio	ons		
- о Тгие Туре			Туре	No
False Individual Security Requirement		Individua	Security Requirement	
Combinatorial Test Suite		Comb	inatorial Test Suite	

Fig. 23. Test Case 3

15 SUBJECT INHERITANCE – TEST CASE 3 (UNDECIDED RULE)

To help us generate policy Rules more quickly we will define some Subject Inheritance relationships. Please note these are the same definitions from Test Case 1. For this continue policy example we will define (3) Subject Inheritance Relationships as follows:

Subject Inheritance:

Beneficiary Values \rightarrow Reviewer (pcmember_1, pcmember_2, pcmember_3) Inherited Values \rightarrow PCmember

If we have created these Relationships correctly based on the above direction it will look like this in Security Policy Tool:

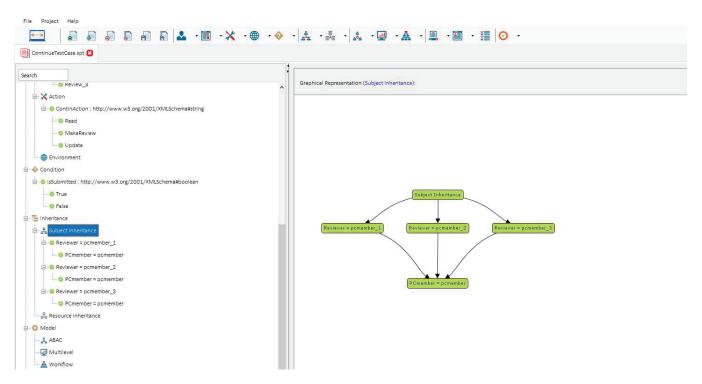


Fig. 24. Subject Inheritance Relationships

By defining these relationships, any (Originated) Rules with Decision = Permit given to PCmember = pcmember will now also be given to pcmember_1, pcmember_2, and pcmember_3 as (Inherited) Rules in our policies. Originated Rules with Decision = Deny are never Inherited. This is because typically Beneficiaries in these relationships are higher ranking/senior roles that by nature will have less restrictions (e.g., denying access) than roles that are providing the Inheritance Value (e.g., pcmember in our example).

Hence, it will authorize the Beneficiaries to obtain all privileges of Inherited Values (e.g., generally lower-level roles) while not obtaining their typically tighter restrictions. If you would like Beneficiaries to be Denied access similar to their Inherited Value you can still do so by manually creating individual rules when you begin modeling.

16 MODELING YOUR POLICY – TEST CASE 3 (UNDECIDED RULE)

Now that we have entered our attributes we can model our two policies (PCmember Policy & Reviewer Policy). See the list below of the rules contained in each of these policies. You can open a "New (blank) Project" and build these policies by entering the following rules below:

PCmember Policy:

(PCmember = pcmember, Read, Paper_1) → Permit (PCmember = pcmember, Read, Paper_2) →Permit (PCmember = pcmember, Read, Paper_3) →Permit (PCmember = pcmember, isSubmitted: False, Read, Review_1) →Deny (PCmember = pcmember, isSubmitted: False, Read, Review_2) →Deny (PCmember = pcmember, isSubmitted: False, Read, Review_3) →Deny NOTE: You will also generate 9 additional "'Inherited Rules" due to the first 3 Permit Rules **Reviewer Policy:** (Reviewer = pcmember_1, isSubmitted: False, Update, Review_1) →Permit

(Reviewer = pcmember_1, isSubnitted: False, Update, Review_1) \rightarrow Permit (Reviewer = pcmember_2, isSubnitted: False, Update, Review_2) \rightarrow Permit (Reviewer = pcmember_3, isSubnitted: False, Update, Review_3) \rightarrow Permit (Reviewer = pcmember_1, isSubnitted: False, MakeReview, Paper_1) \rightarrow Permit (Reviewer = pcmember_2, isSubnitted: False, MakeReview, Paper_2) \rightarrow Permit

(Reviewer = pcmember_3, isSubmitted: False, MakeReview, Paper_3) →Permit (Reviewer = pcmember_3, isSubmitted: False, Update, Review_4) →Permit

After entering the rules above your modeled policies should look like the screenshots below. If you did not create your own Project File, you can simply open Security Policy Tool – Project File: ContinueTestCase3 and these policies will have been already created for you.

Cmember Policy Poli	cy(s) Summary		@ 1	rows out of 1				Search	×
Model	Policy Name	Rule Combination Algorithm	Policy Enforcement	Algorithm N	o. of Rule(s)	Time Create	d	Last Modifi	ed
ABAC	PCmember Policy	Deny-overrides	Deny Biase	ed	15	June 14, 2018 15	5:54:34	June 14, 2018 1	5:54:34
tule (s) defined with s	elected policy (PCmember Policy)			15 rows out of 15				Search	
Sequence No	Subject	Resource	Action	Environment		Condition	Decision	Inheritance	Relation
1	PCmember = pcmember	Paper = Paper_1	ContinAction = Read	Environment = Anv \	/alue Cor	dition = Any Value	Permit	Origin	ated
2	Reviewer = pcmember_1	Paper = Paper_1	ContinAction = Read	Environment = Anv \		dition = Any Value	Permit	Inheri	ted
3	Reviewer = pcmember_2	Paper = Paper_1	ContinAction = Read	Environment = Anv \	/alue Cor	dition = Any Value	Permit	Inheri	ted
4	Reviewer = pcmember_3	Paper = Paper_1	ContinAction = Read	Environment = Anv	/alue Cor	dition = Any Value	Permit	Inheri	ted
5	PCmember = pcmember	Paper = Paper_2	ContinAction = Read	Environment = Anv \	/alue Cor	dition = Any Value	Permit	Origin	ated
6	Reviewer = pcmember_1	Paper = Paper_2	ContinAction = Read	Environment = Any \	/alue Cor	dition = Any Value	Permit	Inheri	ted
7	Reviewer = pcmember_2	Paper = Paper_2	ContinAction = Read	Environment = Any \	/alue Cor	dition = Any Value	Permit	Inheri	ted
8	Reviewer = pcmember_3	Paper = Paper_2	ContinAction = Read	Environment = Any \	/alue Cor	dition = Any Value	Permit	Inheri	ted
9	PCmember = pcmember	Paper = Paper_3	ContinAction = Read	Environment = Any \	/alue Cor	dition = Any Value	Permit	Origin	ated
10	Reviewer = pcmember_1	Paper = Paper_3	ContinAction = Read	Environment = Any \	/alue Cor	dition = Any Value	Permit	Inheri	ted
11	Reviewer = pcmember_2	Paper = Paper_3	ContinAction = Read	Environment = Any \	/alue Cor	dition = Any Value	Permit	Inheri	ted
12	Reviewer = pcmember_3	Paper = Paper_3	ContinAction = Read	Environment = Any \	/alue Cor	dition = Any Value	Permit	Inheri	ted
13	PCmember = pcmember	Review = Review_1	ContinAction = Read	Environment = Any \	/alue is:	ubmitted = False	Deny	Origin	ated
14	PCmember = pcmember	Review = Review_2	ContinAction = Read	Environment = Any \	/alue is	ubmitted = False	Deny	Origin	ated
15	PCmember = pcmember	Review = Review 3	ContinAction = Read	Environment = Anv \		ubmitted = False	Denv	Origin	ated

Fig. 25. PCmember Policy

eviewer Policy Policy	y(s) Summary		I rows o	ut of 1			Search
Model	Policy Name	Rule Combination Algorithm	Policy Enforcement Algorit	thm No. of Rule(s)	Time Created		Last Modified
ABAC	Reviewer Policy	Deny-overrides	Deny Biased	7	June 14, 2018 16:	01:32	June 14, 2018 16:01:32
ule (s) defined with s	selected policy (Reviewer Policy):		۲	7 rows out of 7			Search
ule (s) defined with s	selected policy (Reviewer Policy):		۲	7 rows out of 7			Search
ule (s) defined with s	selected policy (Reviewer Policy): Subject	Resource	Action	7 rows out of 7 Environment	Condition	Decision	Search
					Condition isSubmitted = False	Decision Permit	
Sequence No	Subject	Resource	Action	Environment			Inheritance Relation
Sequence No	Subject Reviewer = pcmember_1	Resource Review = Review_1	Action ContinAction = Update	Environment Environment = Any Value	isSubmitted = False	Permit	Inheritance Relation Originated
Sequence No 1 2	Subject Reviewer = pcmember_1 Reviewer = pcmember_2	Resource Review = Review_1 Review = Review_2	Action ContinAction = Update ContinAction = Update	Environment Environment = Any Value Environment = Any Value	isSubmitted = False isSubmitted = False	Permit Permit	Inheritance Relation Originated Originated
Sequence No 1 2 3	Subject Reviewer = pcmember_1 Reviewer = pcmember_2 Reviewer = pcmember_3	Resource Review = Review_1 Review = Review_2 Review = Review_3	Action ContinAction = Update ContinAction = Update ContinAction = Update	Environment Environment = Any Value Environment = Any Value Environment = Any Value Environment = Any Value	isSubmitted = False isSubmitted = False isSubmitted = False	Permit Permit Permit	Inheritance Relation Originated Originated Originated
Sequence No 1 2 3 4	Subject Reviewer = pcmember_1 Reviewer = pcmember_2 Reviewer = pcmember_3 Reviewer = pcmember_1	Resource Review = Review_1 Review = Review_2 Review = Review_3 Paper = Paper_1	Action ContinAction = Update ContinAction = Update ContinAction = Update ContinAction = MakeReview	Environment Environment = Any Value Environment = Any Value Environment = Any Value	isSubmitted = False isSubmitted = False isSubmitted = False isSubmitted = False	Permit Permit Permit Permit	Inheritance Relation Originated Originated Originated Originated

Fig. 26. Reviewer Policy

17 INDIVIDUAL SECURITY REQUIREMENTS - TEST CASE 3 (UNDECIDED RULE)

The final step before analyzing these policies for errors is to create individual security requirements to use for testing. If you are building a "New (blank) Project" on your own you will enter the following security requirement below:

Individual Security Requirement:

(Reviewer = pcmember_2) & (isSubmitted = False) & (Action = Update) & (Review = Review_4) → decision = Permit

After entering the rule above your individual security requirement should look like the screenshot below. If you did not create your own Project File you can simply open Security Policy Tool – Project File: ContinueTestCase3 and this requirement will have been already created for you.

est Case 3(s) Summary			I rows out of 1		Searc	n 🚺
	cess Control Security Requirement		Requirement Schema		No. of Security Requirement(s)	
	Individual		Test Case 3		1	
curity Requirement (s) di	sfined under selected Requirement Scher	ma (Test Case 3):	٩	1 rows out of 1	Search	n 🚺
curity Requirement (s) de	tfined under selected Requirement Scher Subject	na (Test Case 3): Resource	Action	1 rows out of 1 Environment	Searc	Decision

Fig. 27. Individual Security Requirement

18 POLICY VERIFICATION/ANALYZING RESULTS - TEST CASE 3 (UNDECIDED RULE)

Now that we are ready to test our policies let's discuss the error we will be looking at in this third example. When policies are designed there is potential for an "Undecided Rule" error being created. An Undecided Rule error occurs when your policy contains rules that are not entirely defined or are missing a step.

For example, when the policy author was designing the logic for these continue policies; the author created a rule for pcmember_3 to access "Review_4" but did not define access rules for pcmember_1 and pcmember_2 to access this resource. In this situation, if pcmember_1 or pcmember_2 were to attempt to take action on "Review_4," the system would be forced to make a default decision instead of a specified decision. This may create a security vulnerability due to your system's default evaluation decision being different than what you previously intended. Similar to the "Not Protected Resource" example previously, this error is caused due to the author missing rules. It is not caused due to flawed interpretation of existing rules contained in either of our policies as was the case in Test Case 1 (Rule Conflict).

Next, we will run one "Combined Policy" Verification to reveal the Undecided Rule error that is present in our policies. To do this, we will select Test Case 3 (security requirement) and PCmember Policy & Review Policy as a Combined Policy Verification and analyze our verification result. Again, this will have already been done for you if you open Project File: ContinueTestCase3.

UpToDate Policy Verification (June 14, 2018 17:14:58) Standard Combined Policy 2 Deny-overrides Deny Blased ABAC:PCmember Policy, ABAC:Reviewer	Status	Name	Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement Algorithm		Policy List
			203						
Result(s) with selected verification (Policy Verification (June 14, 2018 17:14:58)) I rows out of 1 Search	pToDate	Policy Verification (June 14, 2018 17:14:58)	Standard	Combined Policy	2	Deny-overrides	Deny Biased	ABAC:PCmember	Policy, ABAC:Reviewer Po
					tion '		Condition	-	
Test Case 3 Reviewer = premember 2 Review = Review = 4 ContinAction = Update Environment = any Value isSubmitted = False Permit False	Requirer	ment Schema Subject	Resour		ction	 1 rows out of 1 Environment 	Condition	Decision	Search Verification Re

Fig. 28. Combined Policy x Test Case 3

Like we did in the "Not Protected Resource" example, by clicking on the Verification Result we can analyze deeper the reasoning for the "False" result we have received. Here is where we would notice we have not created Rules that are attached to Subject = pcmember_2 taking action on Resource = Review_4. We can see this by noticing that every "Match Result" is "Not Applicable" whereas if there were Rules existing for pcmember_2 and Resource = Review_4 we would have at least see one Rule with a (Permit or Deny) Match Result.

WWW.SECURITYPOLICYTOOL.COM

olicy Verific	ation (June 14, 20	18 17:14:58)(s) Summary			1 rows	out of 1			Search	×	
Status		Name	Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement Algorithm		Policy List		
pToDate	Policy Verificatio	n (June 14, 2018 17:14:58)	Standard	Combined Policy	2	Deny-overrides	Deny Biased	ABAC:PCmembe	r Policy, ABAC	Reviewer Pr	olicy
	th selected verifica	tion (Policy Verification (June	14, 2018 17:14:58))			1 rows out of 1			Search	×	6
Result(s) wit	th selected verifica nent Schema	tion (Policy Verification (June Subject	14, 2018 17:14:58)) Resour	ce A	ction	I rows out of 1 Environment	Condition	Decision		ation Result	

Policy(s) and Matching result against	the selcted security requirement:	2 rows	out of 2	Search
Sequence No	Policy Name	Rule Combination Algorithm	Policy Enforcement Algorithm	Combined Result
1	ABAC : PCmember Policy	Deny-overrides	Deny Biased	Deny
2	ABAC : Reviewer Policy	Deny-overrides	Deny Biased	Deny

Sequence No	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Relation	Match Result
1	PCmember = pcmember	Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
2	Reviewer = pcmember_1	Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
3	Reviewer = pcmember_2	Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
4	Reviewer = pcmember_3	Paper = Paper_1	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicable
5	PCmember = pcmember	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
6	Reviewer = pcmember_1	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicabl
7	Reviewer = pcmember_2	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicabl
8	Reviewer = pcmember_3	Paper = Paper_2	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicabl
9	PCmember = pcmember	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicabl
10	Reviewer = pcmember_1	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicabl
11	Reviewer = pcmember_2	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicabl
12	Reviewer = pcmember_3	Paper = Paper_3	ContinAction = Read	Environment = Any Value	Condition = Any Value	Permit	Inherited	Not Applicabl
13	PCmember = pcmember	Review = Review_1	ContinAction = Read	Environment = Any Value	isSubmitted = False	Deny	Originated	Not Applicabl
14	PCmember = pcmember	Review = Review_2	ContinAction = Read	Environment = Any Value	isSubmitted = False	Deny	Originated	Not Applicabl
15	PCmember = pcmember	Review = Review_3	ContinAction = Read	Environment = Any Value	isSubmitted = False	Deny	Originated	Not Applicabl

Fig. 29. PCmember Policy: Match Results

olicy Verification (June 14, 2018 17:14:58)(s) Summary					۲	1 rows out of 1				Search	×	
tatus Name ToDate Policy Verification (June 14, 2018 17:14:58)		me		Verification Techni	que Number of Po	olicy(s) Combi	nation Algorithm	Enforcement Algorithm	n	Policy List		
oToDate Pol	licy Verification (June 14, 2018 1	7:14:58)	Standard Combined Polic		Combined Policy 2		Deny-overrides Deny Biased		ABAC:PCmember Policy, ABAC		teviewer P	Poli
esult(s) with sele	lected verification (Policy Verifica	tion (June 14	4, 2018 17:14:58))	2		1 rov	vs out of 1			Search		3
Requirement Schema Subject		it	Resou	Irce	Action	Envir	onment	Condition	Decision	Verifica	tion Resul	lt
Test Case 3 Reviewer = pcmembe		ember 2	Review = R	eview 4 Con	tinAction = Update	Environmer	ot = Any Value isSubmitted = False		Permit	FALSE		
	tching result against the selcted s			Bu	e Combination Algorit	2 rows out		forcement Algorithm		Search Combined Res	ult	3
	-									1		3
Seque	ence No	ecurity requir Policy ABAC : PCme	r Name	Ru	le Combination Algoriti Deny-overrides		Policy Er	forcement Algorithm Deny Biased		1		1
Seque	ence No	Policy	r Name ember Policy	Ru			Policy Er			Combined Res		3
Seque	nce No 1 2 ching result of Selected Policy aga	Policy ABAC : PCme ABAC : Revi	r Name ember Policy iewer Policy		Deny-overrides	hm	Policy Er	Deny Biased		Combined Res Deny Deny Search	ult	
Seque	ence No	Policy ABAC : PCme ABAC : Revi	r Name ember Policy iewer Policy		Deny-overrides Deny-overrides	hm	Policy Er rows out of 7 Condition	Deny Biased Deny Biased Decision	Inheritance Relati	Combined Res Deny Deny Search	ult	
Seque ule(s) and Matc Sequence No 1	ince No	Policy ABAC : PCme ABAC : Revi inst the selcti Res Review =	r Name ember Policy iewer Policy eed security require source = Review_1	ment: Action ContinAction = Updi	Deny-overrides Deny-overrides Environment	hm // // // // // // // // // // // // //	Policy Er rows out of 7 Condition isSubmitted = F	Deny Biased Deny Biased Deny Biased Decision alse Permit	Originated	Combined Res Deny Deny Search Search Ma Not /	ult	t
Seque ule(s) and Matc Sequence No 1 2	ince No	Policy ABAC : PCme ABAC : Revi inst the selcto Review = Review =	r Name ember Policy iewer Policy ed security require source = Review_1 = Review_2	ment: Action ContinAction = Upda ContinAction = Upda	Deny-overrides Deny-overrides Environm ate Environm te Environm	hm	Policy Er rows out of 7 Condition isSubmitted = F isSubmitted = F	Deny Blased Deny Blased Decision alse Permit alse Permit	Originated Originated	Combined Res Deny Deny Search Not / Not /	ult tch Result Applicab	t
Seque le(s) and Matc Sequence No 1 2 3	ince No	Policy ABAC : PCme ABAC : Revi inst the selcto Review = Review = Review =	Name ember Policy lever Policy ed security require source = Review_1 = Review_2 = Review_3	ment: ContinAction = Updi ContinAction = Updi ContinAction = Updi	Deny-overrides Deny-overrides Environm ste Environm ste Environm te Environm	hm vironment ment = Any Value ment = Any Value	Policy Er rows out of 7 Condition IsSubmitted = F IsSubmitted = F	Deny Blased Deny Blased Decision alse Permit alse Permit alse Permit	Originated Originated Originated	Combined Res Deny Deny Search Search Not / Not /	ult tch Result Applicab Applicab	
Seque ule(s) and Matc Sequence No 1 2 3 4	thing result of Selected Policy aga Subject Reviewer = pcmember_1 Reviewer = pcmember_2 Reviewer = pcmember_3 Reviewer = pcmember_1	Policy ABAC : PCme ABAC : Revi inst the selct Review = Review = Review = Paper =	Name ember Policy lewer Policy ed security require source = Review_1 = Review_2 = Review_3 = Paper_1	ment: Action ContinAction = Updd ContinAction = Updd ContinAction = Updd ContinAction = UpdkeRk	Deny-overrides Deny-overrides Deny-overrides Environm ate Environm exiew Environm	hm	Policy Er rows out of 7 Condition isSubmitted = F isSubmitted = F isSubmitted = F	Deny Blased Deny Blased Decision alse Permit alse Permit alse Permit	Originated Originated Originated Originated	Combined Res Deny Deny Search Search Not / Not / Not /	ult tch Result Applicab Applicab Applicab	
Seque ule(s) and Matc Sequence No 1 2 3	ince No	Policy ABAC : PCme ABAC : Revi inst the selct Review = Review = Review = Paper = Paper =	Name ember Policy lever Policy ed security require source = Review_1 = Review_2 = Review_3	ment: ContinAction = Updi ContinAction = Updi ContinAction = Updi	Deny-overrides Deny-overrides Deny-overrides Environm te Environm te Environm te Environm te Environm te Environm	hm vironment ment = Any Value ment = Any Value	Policy Er rows out of 7 Condition IsSubmitted = F IsSubmitted = F	Deny Biased Deny Biased Decision alse Permit alse Permit alse Permit alse Permit	Originated Originated Originated	Combined Res Deny Deny Search Not / Not / Not / Not /	ult tch Result Applicab Applicab	

Fig. 30. Reviewer Policy: Match Results

As you can see there has not been a rule defined for pcmember_2 \rightarrow Action \rightarrow Review_4 which is known as an Undecided Rule error.

19 RESOLVING THIS ERROR - TEST CASE 3 (UNDECIDED RULE)

To solve this error, the policy author would need to define specific rules for all subject attributes in any policies (e.g., include pcmember_2) that determine pcmember access requets to Review_4.

For example, adding the rules below to the Reviewer Policy for our specific example...

Reviewer Policy: Add (2) New Rules:

 $\begin{array}{l} (\text{Rule No. }=8) \rightarrow (\text{Reviewer} = \text{pcmember_1}) \rightarrow (\text{Action} = \text{Update}) \rightarrow (\text{isSubmitted} = \text{False}) \\ \rightarrow (\text{Resource} = \text{Review_4}) \rightarrow \text{decision} = \text{Permit} \\ (\text{Rule No. }=9) \rightarrow (\text{Reviewer} = \text{pcmember_2}) \rightarrow (\text{Action} = \text{Update}) \rightarrow (\text{isSubmitted} = \text{False}) \\ \rightarrow (\text{Resource} = \text{Review_4}) \rightarrow \text{decision} = \text{Permit} \end{array}$

Review = Review_4

Review = Review_4

Fig. 31. Reviewer Policy: New Rules (8, 9)

Reviewer = pcmember 1

Reviewer = pcmember_2

Now, looking at our Verification results and Match Results we will see that we no longer have an "Undecided Rule" error occurring. The Verification Result is still "False" due to our choices in our Combination Algorithm = Deny-overrides and Enforcement Algorithm = Deny Biased.

ContinAction = Update

ContinAction = Updat

isSubmitted = False

sSubmitted = False

Permit

Originated

Originated

For example, PCmember Policy has no rules related to the security requirement (pcmember_2 \rightarrow Update \rightarrow Review_4) we are using for testing which is why see all Match Rules = Not Applicable. Due to our selection to use Deny Biased for our Enforcement Algorithm the "Combined Result" for PCmember Policy = Deny. However, in the case of the Policy we have the Combined Result = Permit due to the new rules we added (e.g., see new Rule 9 below). Hence, we have opposing Combined Results (PCmember Policy = Deny; Reviewer Policy = Permit). Finally, the Combination Algorithm = Deny-overrides which makes a definitive answer for our Verification Results. The Deny-overrides selection overrules the Permit result from the PCmember Policy in favor of the Deny result from the Reviewer Policy to make the final Verification Result = False.

WWW.SECURITYPOLICYTOOL.COM

Requirement Schema Subject Resource Action Environment Condition Decision Verification Test Case 3 Reviewer = promember_2 Review = Review_4 ContinAction = Update Environment = Any Value IsSubmitted = False Permit FAlse Policy(s) and Matching result against the selcted security requirement: Image: Sequence No Policy Name Rule Combination Algorithm Policy Enforcement Algorithm Combined Result 1 ABAC : PCmember Policy Deny-overrides Deny Biased Deny 2 ABAC : Reviewer Policy Deny-overrides Deny Biased Permit Rule(s) and Matching result of Selected Policy against the selcted security requirement: Image: Sequence No Sequence No Reviewer Policy Deny-overrides Deny Biased Permit Rule(s) and Matching result of Selected Policy against the selcted security requirement: Image: Sequence No Sequence No Sequence No Sequence No ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 1 Reviewer = porember_3 Review = Review_3 ContinAction = Update	olicy Verification (June 21, 2018 11:32:45)(s) Summary					1 rows		Search				
Result(s) with selected verification (Policy Verification (June 21, 2018 11:32:45)) I rows out of 1 Search Requirement Schema Subject Resource Action Environment Condition Decision Verification Test Case 3 Reviewer = pomember_2 Review = Review_4 ContinAction = Update Environment = Any value IsSubmitted = False Permit FALS Policy(s) and Matching result against the selcted security requirement: Image: 2 rows out of 2 Search Search Sequence No Policy Name Rule Combination Algorithm Policy Enforcement Algorithm Combined Result 1 ABAC : Reviewer = Policy Deny-overrides Deny Biased Deny 2 ABAC : Reviewer = Policy Deny-overrides Deny Biased Permit 3 Reviewer = pomember_2 Review = Review_1 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 2 Reviewer = pomember_3 Review = Review_1 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 2 Reviewer = pomember_3 Review = Review_3	Status			Name		Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement Algorithm	6 - E	Policy List
Requirement Schema Subject Resource Action Environment Condition Decision Venfication Test Case 3 Reviewer = pomember_2 Review = Review_4 ContinAction = Update Environment = Anv Value IsSubmitted = False Permit FALS Policy(s) and Matching result against the selcted security requirement: Image: Security against the selcted security requirement: Image: Security against the selcted security requirement: Securities Deny Overrides Deny Biased Deny 1 ABAC : Pomember Policy Deny-overrides Deny Biased Permit Securities Rule(s) and Matching result of Selected Policy against the selcted security requirement: Image: Security Security Biased Permit Originated Not App Sequence No Subject Reviewer Review_1 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App Rule(s) and Matching result of Selected Policy against the selcted security requirement: Image: Security Value IsSubmitted = False Permit Originated Not App 2 Reviewer = pomember_3 Reviewer = Review_2 ContinAction = Update Environment = Any Value IsSubmitted = False	IpToDate Pr	olicy Verification (June	e 21, 2018 11:32:45)	Standard	Combined Policy	2	Deny-overrides	Deny Biased	ABAC:PCmember	Policy, ABAC:Reviewer Polic		
Requirement Schema Subject Resource Action Environment Condition Decision Venfication Test Case 3 Reviewer = pomember_2 Review = Review_4 ContinAction = Update Environment = Anv Value IsSubmitted = False Permit FALS Policy(s) and Matching result against the selcted security requirement: Image: Security against the selcted security requirement: Image: Security against the selcted security requirement: Securities Deny Overrides Deny Biased Deny 1 ABAC : Pomember Policy Deny-overrides Deny Biased Permit Securities Rule(s) and Matching result of Selected Policy against the selcted security requirement: Image: Security Security Biased Permit Originated Not App Sequence No Subject Reviewer Review_1 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App Rule(s) and Matching result of Selected Policy against the selcted security requirement: Image: Security Value IsSubmitted = False Permit Originated Not App 2 Reviewer = pomember_3 Reviewer = Review_2 ContinAction = Update Environment = Any Value IsSubmitted = False									Г	Search		
Test Case 3 Reviewer = pcmember_2 Review = Review_4 ContinAction = Update Environment = Anv Value IsSubmitted = False Permit FALS Policy(s) and Matching result against the selcted security requirement: Image: 2 rows out of 2 Search Search Sequence No Policy Name Rule Combination Algorithm Policy Enforcement Algorithm Combined Result 1 ABAC : PCmember Policy Denv-overrides Denv Biased Denv 2 ABAC : Reviewer Policy Denv-overrides Denv Biased Permit Search Sequence No Selected Policy against the selcted security requirement: Image: Proving Security Securities Denv Biased Denv Sequence No Subject Reviewer Policy Denv-overrides Denv Biased Denv Sequence No Subject Reviewer Review_1 ContinAction = Update Environment = Anv Value IsSubmitted = False Permit Originated Not App 2 Reviewer = pcmember_1 Review = Review_3 ContinAction = Update Environment = Anv Value IsSubmitted = False Permit Originated Not App 3	Result(s) with s	elected verification (P	olicy Verification (June	21, 2018 11:32:45))			I rows out of 1			Search		
Policy(s) and Matching result against the selcted security requirement: Image: Sequence No Policy Name Rule Combination Algorithm Policy Enforcement Algorithm Combined Result 1 ABAC : PCmember Policy Deny-overrides Deny Biased Deny 2 ABAC : PCmember Policy Deny-overrides Deny Biased Permit 2 ABAC : Reviewer Policy Deny-overrides Deny Biased Permit Rule(s) and Matching result of Selected Policy against the selcted security requirement: Image: Search Search Sequence No Subject Resource Action Environment = Any Value IsSubmitted = False Permit Originated Not App 2 Reviewer = promember_3 Reviewer = Review_3 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 3 Reviewer = promember_3 Review = Review_3 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 4 Reviewer = promember_1 Paper = Paper_3 ContinAction = MakeReview Environment = Any Value IsSubmitted = False Permit O	Requirement Schema		Subject	Resour	ce A	ction	Environment	Condition	Decision	Verification Result		
Sequence No Policy Name Rule Combination Algorithm Policy Enforcement Algorithm Combined Result 1 ABAC : PCmember Policy Deny-overrides Deny Biased Deny 2 ABAC : Reviewer Policy Deny-overrides Deny Biased Deny Rule(s) and Matching result of Selected Policy against the selected security requirement: Image: Sequence No Subject Resource Action Environment Condition Decision Inheritance Relation Matching 1 Reviewer = pcmember_1 Reviewer = pcmember_2 Reviewer = pcmember_2 Reviewer = Review_3 Continaction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 3 Reviewer = pcmember_3 Review = Review_3 Continaction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 4 Reviewer = pcmember_1 Paper = Paper_1 Continaction = MakeReview Environment = Any Value IsSubmitted = False Permit Originated Not App 5 Reviewer = pcmember_3 Review = Review_3 Continaction = MakeReview Environment = Any Value IsSubmitted = False Permit	Test Case 3 Reviewer		eviewer = pcmember_2	Review = Re	view_4 ContinAct	ion = Update	nvironment = Anv Value	isSubmitted = False	Permit	FALSE		
Sequence No Subject Resource Action Environment Condition Decision Inheritance Relation Match 1 Reviewer = pcmember_1 Review = Review_1 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 2 Reviewer = pcmember_2 Review = Review_2 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 3 Reviewer = pcmember_3 Review = Review_3 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 4 Reviewer = pcmember_1 Paper = Paper_1 ContinAction = MakeReview Environment = Any Value IsSubmitted = False Permit Originated Not App 5 Reviewer = pcmember_3 Paper = Paper_2 ContinAction = MakeReview Environment = Any Value IsSubmitted = False Permit Originated Not App 6 Reviewer = pcmember_3 Paper = Paper_3 ContinAction = Update Environment = Any Value IsSubmitted = False Permit				member Policy				1				
1 Reviewer = pcmember_1 Review = Review_1 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not App 2 Reviewer = pcmember_2 Review = Review_2 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not App 3 Reviewer = pcmember_3 Review = Review_3 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not App 4 Reviewer = pcmember_1 Paper = Paper_1 ContinAction = MakeReview Environment = Any Value isSubmitted = False Permit Originated Not App 5 Reviewer = pcmember_2 Paper = Paper_2 ContinAction = MakeReview Environment = Any Value isSubmitted = False Permit Originated Not App 6 Reviewer = pcmember_3 Paper = Paper_2 ContinAction = MakeReview Environment = Any Value isSubmitted = False Permit Originated Not App 7 Reviewer = pcmember_3 Review_4 ContinAction = Update Environment = Any Value isSubmitted = Fal		-				Fovironmer		n Decision	Ļ			
2 Reviewer = pcmember_2 Review = Review_2 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not App 3 Reviewer = pcmember_3 Review = Review_3 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not App 4 Reviewer = pcmember_1 Paper = Paper_1 ContinAction = MakeReview Environment = Any Value isSubmitted = False Permit Originated Not App 5 Reviewer = pcmember_2 Paper = Paper_2 ContinAction = MakeReview Environment = Any Value isSubmitted = False Permit Originated Not App 6 Reviewer = pcmember_3 Paper = Paper_3 ContinAction = MakeReview Environment = Any Value isSubmitted = False Permit Originated Not App 7 Reviewer = pcmember_3 Review_4 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not App 7 Reviewer = pcmember_3 Review_4 ContinAction = Update Environment = Any Value isSubmitted = False	100 A * 2 200			w = Peview 1	ContinAction = Lodate		isSubmitted =	False Dermit	Originated	N		
3 Reviewer = pcmember_3 Review = Review_3 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 4 Reviewer = pcmember_1 Paper = Paper_1 ContinAction = MakeReview Environment = Any Value IsSubmitted = False Permit Originated Not App 5 Reviewer = pcmember_2 Paper = Paper_2 ContinAction = MakeReview Environment = Any Value IsSubmitted = False Permit Originated Not App 6 Reviewer = pcmember_3 Paper = Paper_3 ContinAction = MakeReview Environment = Any Value IsSubmitted = False Permit Originated Not App 7 Reviewer = pcmember_3 Review_4 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App 7 Reviewer = pcmember_3 Review_4 ContinAction = Update Environment = Any Value IsSubmitted = False Permit Originated Not App	1						ly value			Not Applicable Not Applicable		
4 Reviewer = pcmember_1 Paper = Paper_1 ContinAction = MakeReview Environment = Any Value isSubmitted = False Permit Originated Not App 5 Reviewer = pcmember_2 Paper = Paper_2 ContinAction = MakeReview Environment = Any Value isSubmitted = False Permit Originated Not App 6 Reviewer = pcmember_3 Paper = Paper_3 ContinAction = MakeReview Environment = Any Value isSubmitted = False Permit Originated Not App 7 Reviewer = pcmember_3 Review = Review_4 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not App									ly value			Not Applicable
6 Reviewer = pcmember_3 Paper = Paper_3 ContinAction = MakeReview Environment = Any Value IsSubmitted = False Permit Originated Not App 7 Reviewer = pcmember_3 Review = Review_4 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not App	2		nember_3 Review	w = Review_3								
7 Reviewer = pcmember_3 Review = Review_4 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not Applicated	2 3	Reviewer = pcn		-			v Value isSubmitted =	False Permit	Originated	Not Applicable		
	2 3 4	Reviewer = pcn Reviewer = pcn	member_1 Pape	er = Paper_1 (ContinAction = MakeReview	Environment = An	ly value	i elee	-	Not Applicable Not Applicable		
	2 3 4 5 6	Reviewer = pcn Reviewer = pcn Reviewer = pcn Reviewer = pcn	nember_1 Pape nember_2 Pape nember_3 Pape	er = Paper_1 (er = Paper_2 (er = Paper_3 (ContinAction = MakeReview ContinAction = MakeReview ContinAction = MakeReview	Environment = Ar Environment = Ar	y Value isSubmitted = y Value isSubmitted =	False Permit False Permit	Originated Originated			
8 Reviewer = pcmember_1 Review = Review_4 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Not App 9 Reviewer = pcmember_2 Review = Review_4 ContinAction = Update Environment = Any Value isSubmitted = False Permit Originated Permit	2 3 4 5 6 7	Reviewer = pcn Reviewer = pcn Reviewer = pcn Reviewer = pcn Reviewer = pcn	nember_1 Pape nember_2 Pape nember_3 Pape nember_3 Review	er = Paper_1 (er = Paper_2 (er = Paper_2 (w = Review_4	ContinAction = MakeReview ContinAction = MakeReview ContinAction = MakeReview ContinAction = Update	Environment = Ar Environment = Ar Environment = Ar Environment = Ar	y Value isSubmitted = y Value isSubmitted = y Value isSubmitted = y Value isSubmitted =	False Permit False Permit False Permit	Originated Originated Originated	Not Applicable Not Applicable Not Applicable		

Fig. 32. Updated Results: No Undecided Rule

20 CONCLUSION

Now you should have a better understanding of what to look for as you go onto verify your access control policies with Security Policy Tool. In addition to this document there are other resources located in the Learning Center in your My account page that will help you start leveraging Security Policy Tool to prevent access control leaks, today!

If you have not yet, download Security Policy Tool – Lite Version for FREE now! Close the door the Access Control Leaks and save time and cost creating, modeling, testing, and verifying your access control policies, today.

Click here to begin securing your policies now \rightarrow <u>Lite Version</u>.



InfoBeyond Technology, LLC is an innovative company specializing in Network, Machine Learning and Data Security within the Information Technology industry. The mission of InfoBeyond is to research, develop, and deliver viable software products for network communication and security. Some of our research is sponsored by Department of Defense, Department of Energy, Missile Defense Agency, Department of Transportation, NIST (National Institute of Standards and Technology), etc. Security Policy Tool is Awarded the 2017 Innovative Security Solution Award at the 2017 Big Data and SDN/NFV Summit. NXdrive is a fragment-based cybersecurity storage system and more information can be found at www.NXdrive.com. The company is featured as one of 50 fast growth IT small businesses in 2017 by The Silicon Review.