## **University 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 University 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>. 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 university example contains two policies (GradePolicy & TAPolicy). 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.

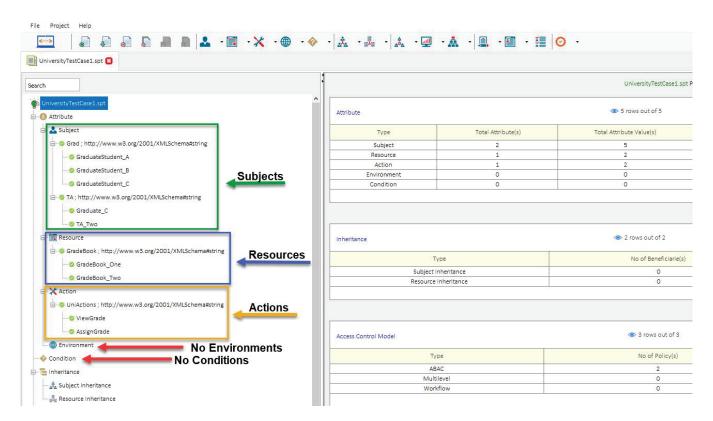


Fig. 1. Test Case 1

## 3 MODELING YOUR POLICY – TEST CASE 1 (RULE CONFLICT)

Now that we have entered our attributes we can model our two policies (GradePolicy & TAPolicy). 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:

#### GradePolicy:

```
(Subject = Any Value & Grad = GraduateStudent_A, ViewGrade, GradeBook_One) → Permit
(Subject = Any Value & Grad = GraduateStudent_A, AssignGrade, GradeBook_One) → Deny
(Subject = Any Value & Grad = GraduateStudent_A, Any Action, GradeBook_Two) → Deny
(Subject = Any Value & Grad = GraduateStudent_B, ViewGrade, GradeBook_One) → Permit
(Subject = Any Value & Grad = GraduateStudent_B, AssignGrade, GradeBook_One) → Deny
(Subject = Any Value & Grad = GraduateStudent_B, Any Action, GradeBook_One) → Deny
(Subject = Any Value & Grad = GraduateStudent_B, Any Action, GradeBook_Two) → Deny
(Subject = Any Value & Grad = GraduateStudent_C, ViewGrade, GradeBook_Two) → Permit
(Subject = Any Value & Grad = GraduateStudent_C, AssignGrade, GradeBook_Two) → Deny
(Subject = Any Value & Grad = GraduateStudent_C, Any Action, GradeBook_Two) → Deny
(Subject = Any Value & Grad = GraduateStudent_C, Any Action, GradeBook_One) → Deny
```

#### **TAPolicy**:

(Subject = Any Value & TA = GraduateStudent\_C, ViewGrade, GradeBook\_One)  $\rightarrow$ Permit (Subject = Any Value & TA = GraduateStudent\_C, AssignGrade, GradeBook\_One)  $\rightarrow$ Deny (Subject = Any Value & TA = GraduateStudent\_C, Any Action, GradeBook\_Two)  $\rightarrow$ Deny (Subject = Any Value & TA = TA\_Two, ViewGrade, GradeBook\_Two)  $\rightarrow$ Permit (Subject = Any Value & TA = TA\_Two, AssignGrade, GradeBook\_Two)  $\rightarrow$ Permit

(Subject = Any Value & TA = TA\_Two, Any Action, GradeBook\_One) →Deny

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: UniversityTestCase1 and these policies will have been already created for you.

GradPolicy Po	licy(s) Summary		1 rows out of 1			Searc	h 関 🕻	
Model	Policy Name Rule Combinatio	Algorithm Policy Enfor	cement Algorithm	No. of Rule(s)	Time Created		Last Modified	C,
ABAC	GradPolicy Deny-over	ides Der	ny Biased	ut e	ne 14, 2018 11:54:11	June	14, 2018 11:54:11	3
	d with selected policy (GradPolicy):	- to be set of	9 rows out of 9		Condition	Search	h	•
Sequence No	Subject	Resource	Action	Environment				
2								
1	Subject = Any Value & Grad = GraduateStudent_		UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	
1 2	Subject = Any Value & Grad = GraduateStudent_ Subject = Any Value & Grad = GraduateStudent_		UniActions = ViewGrade UniActions = AssignGrade	Environment = Any Value Environment = Any Value				
		GradeBook = GradeBook_One			Condition = Any Value	Permit	Originated	
2	Subject = Any Value & Grad = GraduateStudent	A GradeBook = GradeBook_One A GradeBook = GradeBook_Two	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value Condition = Any Value Condition = Any Value	Permit Deny	Originated Originated	
2 3	Subject = Any Value & Grad = GraduateStudent Subject = Any Value & Grad = GraduateStudent	GradeBook = GradeBook_One GradeBook = GradeBook_Two GradeBook = GradeBook_One GradeBook = GradeBook_One	UniActions = AssignGrade Action = Any Value	Environment = Any Value Environment = Any Value	Condition = Any Value Condition = Any Value Condition = Any Value	Permit Deny Deny	Originated Originated Originated	
2 3 4	Subject = Any Value & Grad = GraduateStudent Subject = Any Value & Grad = GraduateStudent Subject = Any Value & Grad = GraduateStudent	GradeBook = GradeBook_One GradeBook = GradeBook_Two GradeBook = GradeBook_Two GradeBook = GradeBook_One	UniActions = AssignGrade Action = Any Value UniActions = ViewGrade	Environment = Any Value Environment = Any Value Environment = Any Value	Condition = Any Value Condition = Any Value Condition = Any Value Condition = Any Value Condition = Any Value	Permit Deny Deny Permit	Originated Originated Originated Originated	
2 3 4 5	Subject = Any Value & Grad = GraduateStudent Subject = Any Value & Grad = GraduateStudent Subject = Any Value & Grad = GraduateStudent Subject = Any Value & Grad = GraduateStudent	A GradeBook = GradeBook_One GradeBook = GradeBook_Two GradeBook = GradeBook_Two GradeBook = GradeBook_One GradeBook = GradeBook_Two	UniActions = AssignGrade Action = Any Value UniActions = ViewGrade UniActions = AssignGrade	Environment = Any Value Environment = Any Value Environment = Any Value Environment = Any Value	Condition = Any Value Condition = Any Value	Permit Deny Deny Permit Deny	Originated Originated Originated Originated Originated	
2 3 4 5 6	Subject = Any Value & Grad = GraduateStudent Subject = Any Value & Grad = GraduateStudent	A GradeBook = GradeBook_One GradeBook = GradeBook_Two GradeBook = GradeBook_Two GradeBook = GradeBook_One GradeBook = GradeBook_Two GradeBook = GradeBook_Two	UniActions = AssignGrade Action = Any Value UniActions = ViewGrade UniActions = AssignGrade Action = Any Value	Environment = Any Value Environment = Any Value Environment = Any Value Environment = Any Value Environment = Any Value	Condition = Any Value Condition = Any Value	Permit Deny Deny Permit Deny Deny	Originated Originated Originated Originated Originated Originated	

#### Fig. 2. GradePolicy

TAPolicy Policy(s	s) Summary		I rows out of 1			Search	<b>.</b>
Model	Policy Name	Rule Combination Algorithm	Policy Enforcement Algorithm	No. of Rule(s)	Time Created	Last Modified	ſĘ
ABAC	TAPolicy	Deny-overrides	Deny Biased	6	June 14, 2018 11:57:50	June 14, 2018 11:57:50	Ĵ

Rule (s) defined	with selected policy (TAPolicy):		6 rows out a	of 6		Sear	ch 🚺 🦸	-
Sequence No	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Relation	0
1	Subject = Any Value & TA = Graduate C	GradeBook = GradeBook_One	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	-
2	Subject = Any Value & TA = Graduate C	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	
3	Subject = Any Value & TA = Graduate C	GradeBook = GradeBook_Two	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated	
4	Subject = Any Value & TA = TA Two	GradeBook = GradeBook_Two	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	
5	Subject = Any Value & TA = TA Two	GradeBook = GradeBook_Two	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	
6	Subject = Any Value & TA = TA Two	GradeBook = GradeBook_One	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated	



## 4 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 security requirements as follows.

#### **Individual Security Requirements:**

 $(Grad = GraduateStudent_C \& TA = GraduateStudent_C) \& (Action = ViewGrade) \& (GradeBook = GradeBook_One) \rightarrow decision = Permit$ 

 $(Grad = GraduateStudent_C \& TA = GraduateStudent_C) \& (Action = ViewGrade) \& (GradeBook = GradeBook_Two) \rightarrow decision = Permit$ 

After entering the rules above your individual security requirements should look like the screenshots below. If you did not create your own Project File you can simply open Security Policy Tool – Project File: UniversityTestCase1 and these requirements will have been already created for you.

fest Case 1(s) Sum	mary	I rows o	ut of 1		Search	
	Access Control Security Requirement	Require	ment Schema	No. of	Security Requirement(s)	
	Individual	Tes	t Case 1		2	
ecurity Requirem	ent (s) defined under selected Requirement Schema (T	Test Case 1):	2 rows out	of 2	Search	<b>8</b> 🖶
Security Requirem	ent (s) defined under selected Requirement Schema (T Subject	fest Case 1): Resource	2 rows out Action	of 2 Environment	Search	Decision

Fig. 4. Individual Security Requirements

## 5 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 at this university has a role of both TA and Graduate Student at the facility. Due to this, the individual is assigned both (TA: GraduateStudent\_C and Grad: GraduateStudent\_C) attribute values by the system during access evaluation. In the GradePolicy it defines that graduate students can view grades but cannot assign grades. However, in the TAPolicy graduate students can assign grades resulting in a Rule Conflict (e.g., Permitted to AssignGrade in TAPolicy, Denied to AssignGrade in GradePolicy).

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 GradePolicy and Test Case 1 (security requirement) as a Single Policy Verification and also choose TAPolicy and Test Case 1 (security requirement) as a Single Policy Verification and analyze our verification results. Again, this will have already been done for you if you open Project File: UniversityTestCase1.

Policy Verification I	June 14, 2018 14:58:09)(s) Summary			1 rows out of 1			Sear	rch 🚺	-
Status	Name	Verification Type	Verification Techni	ique Number of Pol	icy(s) Combination Alg	orithm Enforcement	t Algorithm	Policy List	6
UpToDate	Policy Verification (June 14, 2018 14:58:09)	Standard	Single Policy	1	Deny-overrid	les Deny B	iased	ABAC:GradePolic	v :
Result(s) with sele	cted verification (Policy Verification (June 14, 201		source	O 2 m Action	ows out of 2 Environment	Condition	Searc	th	t C
		Res				Condition Condition = Any Value		1	1

Fig. 5. GradePolicy x Test Case 1

olicy Verification (Jur	e 14, 2018 14:58:16)(s) Summary		<	1 rows out of 1			Sear	ch 🗵	-
Status	Name	Verification Type Ve	erification Technique	Number of Policy	(s) Combination A	lgorithm Enforcem	ent Algorithm	Policy List	
UpToDate Po	licy Verification (June 14, 2018 14:58:16)	Standard	Single Policy	1	Deny-over	rides Den	y Biased	ABAC:TAPolic	γ
esult(s) with selecter	I verification (Policy Verification (June 14, 2018 1	14:58:16))		2 rows (	out of 2		Searc	h 🚺	-
		(4:58:16)) Resource		2 rows of Action	out of 2 Environment	Condition	Search	h Verification Resu	IT
Result(s) with selecter Requirement Schemi Test Case 1		Resource		Action		Condition			F

Fig. 6. TAPolicy x Test Case 1

As you can see from our verification results our policies are both Permitting and Denying the individual (Grad = GraduateStudent\_C/TA = GraduateStudent\_C) from viewing GradeBook\_One and GradeBook\_Two which is known as a Rule Conflict error.

## 6 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 all (4) of our specific Results (GradePolicyxTestCase1: False;True & TAPolicyxTestCase1: True;False) and see which Rules have "Match Results".

See the screenshots below of our two Policies Match Results to discover which specific rules are related to our Verification Results (e.g., False, True).

Result(s) with select	ted verification (Policy Verification (June 14, 2018	14:58:09))			rows out of 2			Search	X
Requirement Scher	ma Subject	Resource		Action	Envir	onment	Condition	Decision	Verification Result
Test Case 1	TA = Graduate_C & Grad = GraduateStude	nt_C GradeBook = GradeB	ook_One Uni	Actions = ViewGrade	Environmen	t = Any Value C	Condition = Any Va	lue Permit	FALSE
Test Case 1	TA = Graduate_C & Grad = GraduateStude	nt_C GradeBook = GradeB	ook_Two Uni	Actions = ViewGrade			Condition = Any Va		TRUE
Policy(s) and Match	hing result against the selcted security requirement	é.		🐵 1 rows c	out of 1			Search	×
	Policy Name	Rule Combination Algorithm		Poli	cy Enforcement	Algorithm		Combined R	esult
ABA	AC : GradePolicy	Deny-overrides			Deny Biase	d	1	Deny	
Rule(s) and Matchir	ng result of Selected Policy against the selcted secu	urity requirement:		۲	9 rows out of 9			Search	X
		rity requirement:	Action		9 rows out of 9	Condition	Decision	Search	
equence No	Subject	Resource		Envi	ronment	Condition		Inheritance Relation	Match Result
equence No	Subject bject = Any Value & Grad = GraduateStudent_A	Resource GradeBook = GradeBook_One	UniActions = Vie	Envi wGrade Environme	ronment ent = Any Value	Condition Condition = Any Va	alue Permit	Inheritance Relation Originated	Match Result
equence No 1 Sul 2 Sul	Subject         Image: Subject and the second s	Resource	UniActions = Vie UniActions = Assi	wGrade Environme gnGrade Environme	ronment ent = Any Value ent = Any Value	Condition Condition = Any Va Condition = Any Va	alue Permit alue Deny	Inheritance Relation	Match Result Not Applicab Not Applicab
equence No 1 Sul 2 Sul 3 Sul	Subject           bject = Any Value & Grad = GraduateStudent_A           bject = Any Value & Grad = GraduateStudent_A           bject = Any Value & Grad = GraduateStudent_A	Resource GradeBook = GradeBook_One GradeBook = GradeBook_One	UniActions = Vie	wGrade Environme gnGrade Environme Value Environme	ronment ent = Any Value	Condition Condition = Any Va	alue Permit alue Deny alue Deny	Inheritance Relation Originated Originated	Match Result Not Applicab Not Applicab Not Applicab
equence No 1 Sul 2 Sul 3 Sul 4 Sul	Subject         Subject           bject = Any Value & Grad = GraduateStudent_A         C           bject = Any Value & Grad = GraduateStudent_A         C           bject = Any Value & Grad = GraduateStudent_A         C           bject = Any Value & Grad = GraduateStudent_A         C	Resource GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_Two	UniActions = View UniActions = Assi Action = Any V	Environme gnGrade Environme /alue Environme wGrade Environme	ronment ent = Any Value ent = Any Value ent = Any Value	Condition Condition = Any Va Condition = Any Va Condition = Any Va	alue Permit alue Deny alue Deny alue Permit	Inheritance Relation Originated Originated Originated	Match Result Not Applicab Not Applicab Not Applicab Not Applicab
equence No 1 Sul 2 Sul 3 Sul 4 Sul 5 Sul 6 Sul	Subject         Subject           bject = Any Value & Grad = GraduateStudent A         GraduateStudent A           bject = Any Value & Grad = GraduateStudent A         GraduateStudent A           bject = Any Value & Grad = GraduateStudent A         GraduateStudent B           bject = Any Value & Grad = GraduateStudent B         GraduateStudent B           bject = Any Value & Grad = GraduateStudent B         Grad = GraduateStudent B	Resource GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_Two	UniActions = Viev UniActions = Assig Action = Any V UniActions = Viev UniActions = Assig Action = Any V	Environme gnGrade Environme Value Environme wGrade Environme gnGrade Environme	ronment Int = Any Value Int = Any Value Int = Any Value Int = Any Value	Condition Condition = Any Va Condition = Any Va Condition = Any Va Condition = Any Va	alue Permit alue Deny alue Deny alue Permit alue Deny	Inheritance Relation Originated Originated Originated Originated Originated Originated	Match Result Not Applicab Not Applicab Not Applicab Not Applicab Not Applicab
equence No 1 Sul 2 Sul 3 Sul 4 Sul 5 Sul 6 Sul 7 Sul	Subject           bject = Any Value & Grad = GraduateStudent A           (bject = Any Value & Grad = GraduateStudent A           (bject = Any Value & Grad = GraduateStudent A           (bject = Any Value & Grad = GraduateStudent A           (bject = Any Value & Grad = GraduateStudent B           (bject = Any Value & Grad = GraduateStudent B           (bject = Any Value & Grad = GraduateStudent B           (bject = Any Value & Grad = GraduateStudent B           (bject = Any Value & Grad = GraduateStudent C	Resource GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_Two GradeBook = GradeBook_Two GradeBook = GradeBook_Two	UniActions = View UniActions = Assignation Action = Any V UniActions = View UniActions = Assignation Action = Any V UniActions = View	wGrade Environme gnGrade Environme Value Environme gnGrade Environme gnGrade Environme Value Environme wGrade Environme	ronment Int = Any Value Int = Any Value Int = Any Value Int = Any Value Int = Any Value	Condition Condition = Any Va Condition = Any Va Condition = Any Va Condition = Any Va Condition = Any Va	alue Permit alue Deny alue Deny alue Permit alue Deny alue Deny alue Permit	Inheritance Relation Originated Originated Originated Originated Originated Originated	Match Result Not Applicabl Not Applicabl Not Applicabl Not Applicabl Not Applicabl
Surface       1     Surface       2     Surface       3     Surface       4     Surface       5     Surface       6     Surface       7     Surface	Subject         Subject           bject = Any Value & Grad = GraduateStudent A         C           bject = Any Value & Grad = GraduateStudent A         C           bject = Any Value & Grad = GraduateStudent A         C           bject = Any Value & Grad = GraduateStudent A         C           bject = Any Value & Grad = GraduateStudent B         C           bject = Any Value & Grad = GraduateStudent B         C           bject = Any Value & Grad = GraduateStudent B         C           bject = Any Value & Grad = GraduateStudent C         C           bject = Any Value & Grad = GraduateStudent C         C	Resource GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_Two	UniActions = Viev UniActions = Assig Action = Any V UniActions = Viev UniActions = Assig Action = Any V	Environme griGrade Environme wGrade Environme wGrade Environme yalue Environme value Environme wGrade Environme wGrade Environme	ronment Int = Any Value Int = Any Value	Condition Condition = Any Va Condition = Any Va Condition = Any Va Condition = Any Va Condition = Any Va	alue Permit alue Deny alue Deny alue Permit alue Deny alue Deny alue Permit alue Deny	Inheritance Relation Originated Originated Originated Originated Originated Originated	Match Resul Not Applicat Not Applicat Not Applicat Not Applicat Not Applicat

Fig. 7. GradePolicy: Match Results (GradeBook\_One)

Result(s) with s	selected verif	fication (Policy Verification (June 14, 20)	18 14:58:09))		2 rov	ws out of 2				Search	
Requirement S	Schema	Subject	Resource		Action	Enviro	onment	Co	ondition	Decision	Verification Result
Test Case	ie 1	TA = Graduate_C & Grad = GraduateStu	dent_C GradeBook = GradeB	Book_One UniActio	ons = ViewGrade	Environmen	t = Any Value	Conditio	n = Any Valu	Je Permit	FALSE
Test Case	ie 1	TA = Graduate_C & Grad = GraduateStu	dent_C GradeBook = GradeB	Book_Two UniActio	ons = ViewGrade	Environmen	t = Any Value	Conditio	m = Any Valu	Je Permit	TRUE
Policy(s) and N	Matching resu	ult against the selcted security requireme	nt:		👁 1 rows ou	t of 1				Search	X
	Policy N	Name	Rule Combination Algorithm	n	Policy	Enforcement	Algorithm			Combined Re	esult
Rule(s) and Ma	ABAC : Grad	idePolicy	Deny-overrides		<b>@</b> 9	Deny Biaser	d			Permit	× I
		it of Selected Policy against the selcted se	curity requirement:	Artion		rows out of 9			Decision	Search	
equence No	latching result	it of Selected Policy against the selcted se Subject	curity requirement: Resource	Action	Enviro	rows out of 9 nment	Condition		Decision	Search	Match Result
Sequence No	latching result	It of Selected Policy against the selcted se Subject Any Value & Grad = GraduateStudent_A	curity requirement: Resource GradeBook = GradeBook_One	UniActions = ViewGra	Enviro de Environment	rows out of 9 nment t = Any Value	Condition Condition = Any	/ Value	Permit	Search Inheritance Relation Originated	Match Result Not Applicabl
equence No 1 2	latching result Subject = A Subject = A	t of Selected Policy against the selcted se Subject Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_A	curity requirement: Resource GradeBook = GradeBook_One GradeBook = GradeBook_One	UniActions = ViewGra UniActions = AssignGr	Enviro de Environment ade Environment	rows out of 9 nment t = Any Value t = Any Value	Condition Condition = Any Condition = Any	/ Value / Value	Permit Deny	Search Inheritance Relation Originated Originated	Match Result Not Applicabl Not Applicabl
Sequence No 1 2 3	latching result Subject = A Subject = A Subject = A	t of Selected Policy against the selcted se Subject Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_A	curity requirement: Resource GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_Two	UniActions = ViewGra	Environment de Environment ade Environment	rows out of 9 mment t = Any Value t = Any Value t = Any Value	Condition Condition = Any Condition = Any Condition = Any	/ Value / Value / Value	Permit Deny Deny	Search Inheritance Relation Originated Originated Originated	Match Result Not Applicabl Not Applicabl Not Applicabl
Sequence No 1 2	latching result Subject = A Subject = A Subject = A Subject = A	t of Selected Policy against the selcted se Subject Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_B	curity requirement: Resource GradeBook = GradeBook_One GradeBook = GradeBook_Two GradeBook = GradeBook_Two GradeBook = GradeBook_One	UniActions = ViewGra UniActions = AssignGr Action = Any Value UniActions = ViewGra	Environment ade Environment ade Environment Environment de Environment	rows out of 9 mment t = Any Value t = Any Value t = Any Value t = Any Value	Condition Condition = Any Condition = Any Condition = Any Condition = Any	/ Value / Value / Value / Value	Permit Deny	Search Inheritance Relation Originated Originated Originated Originated	Match Result Not Applicabl Not Applicabl Not Applicabl Not Applicabl
Sequence No 1 2 3 4	latching result Subject = A Subject = A Subject = A	t of Selected Policy against the selcted se Subject Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_B Any Value & Grad = GraduateStudent_B	curity requirement: Resource GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_Two	UniActions = ViewGra UniActions = AssignGr Action = Any Value	de Environment ade Environment e Environment de Environment ade Environment	rows out of 9 mment t = Any Value t = Any Value t = Any Value	Condition Condition = Any Condition = Any Condition = Any	/ Value / Value / Value / Value / Value	Permit Deny Deny Permit	Search Inheritance Relation Originated Originated Originated	Match Result Not Applicable Not Applicable Not Applicable Not Applicable Not Applicable
Sequence No 1 2 3 4 5	latching result Subject = A Subject = A Subject = A Subject = A Subject = A	t of Selected Policy against the selcted se Subject Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_A Any Value & Grad = GraduateStudent_B Any Value & Grad = GraduateStudent_B Any Value & Grad = GraduateStudent_B	Curity requirement: Resource GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One	UniActions = ViewGra UniActions = AssignGr Action = Any Value UniActions = ViewGra UniActions = AssignGr	Environment de Environment de Environment de Environment ade Environment e Environment	rows out of 9 nment t = Any Value t = Any Value t = Any Value t = Any Value t = Any Value	Condition Condition = Any Condition = Any Condition = Any Condition = Any Condition = Any	/ Value / Value / Value / Value / Value / Value	Permit Deny Deny Permit Deny	Search Inheritance Relation Originated Originated Originated Originated	Match Result Not Applicable Not Applicable Not Applicable Not Applicable Not Applicable
Sequence No 1 2 3 4 5 6	Subject = A Subject = A Subject = A Subject = A Subject = A Subject = A Subject = A	t of Selected Policy against the selcted se Subject Any Value & Grad = GraduateStudent A Any Value & Grad = GraduateStudent A Any Value & Grad = GraduateStudent A Any Value & Grad = GraduateStudent B Any Value & Grad = GraduateStudent B Any Value & Grad = GraduateStudent B Any Value & Grad = GraduateStudent B	Curity requirement: Resource GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_Two	UniActions = ViewGra UniActions = AssignGr Action = Any Value UniActions = ViewGra UniActions = AssignGr Action = Any Value	Environment de Environment de Environment de Environment ade Environment de Environment de Environment	rows out of 9 mment t = Any Value t = Any Value t = Any Value t = Any Value t = Any Value	Condition Condition = Any Condition = Any Condition = Any Condition = Any Condition = Any	/ Value / Value / Value / Value / Value / Value / Value	Permit Deny Deny Permit Deny Deny	Search Inheritance Relation Originated Originated Originated Originated Originated	Match Result Not Applicable Not Applicable Not Applicable Not Applicable Not Applicable

Fig. 8. GradePolicy: Match Results (GradeBook\_Two)

#### WWW.SECURITYPOLICYTOOL.COM

Result(s) with se	elected verification (Policy Verification (June	e 14, 2018 14:58:16	i))		<b>@</b> 2	rows out of	2		Searc	h 🚺
Requirement Se	chema Subject		Res	ource	Action	E	nvironment	Condition	n Decision	Verification Resul
Test Case	1 TA = Graduate_C & Grad = Grad	luateStudent_C	GradeBook = G	GradeBook_One	UniActions = ViewGrade	Enviror	ment = Any Value	Condition = An	Value Permit	TRUE
Test Case	1 TA = Graduate_C & Grad = Grad	luateStudent_C	GradeBook = G	aradeBook_Two	UniActions = ViewGrade		nment = Anv Value	Condition = An	Value Permit	FALSE
Policy(s) and Mi	Policy Name		nbination Algori	thm	1 rows		ent Algorithm		Search Combined F	
	ABAC : TAPolicy	D	eny-overrides			Deny Bi	ased		Permit	
tule(s) and Mat	ching result of Selected Policy against the s	elcted security requi	rement:		۲	6 rows out	of 6		Searc	h
Sequence No	Subject	Resour	rce	Action	Environn	nent	Condition	Decision	Inheritance Relation	Match Result
1	Subject = Any Value & TA = Graduate_C	GradeBook = Grad	deBook_One	UniActions = View	Grade Environment =	Any Value	Condition = Any Va	ue Permit	Originated	Permit
2	Subject = Any Value & TA = Graduate_C	GradeBook = Grad	deBook_One	UniActions = Assign	nGrade Environment =	Any Value	Condition = Any Va	ue Permit	Originated	Not Applicabl
3	Subject = Any Value & TA = Graduate_C	GradeBook = Grad	deBook_Two	Action = Any Va	alue Environment =	Any Value	Condition = Any Va	ue Deny	Originated	Not Applicabl
4	Subject = Any Value & TA = TA_Two	GradeBook = Grad		UniActions = View	Grade Environment =	Any Value	Condition = Any Va	ue Permit	Originated	Not Applicabl
5	Subject = Any Value & TA = TA_Two	GradeBook = Grad	-	UniActions = Assign	nGrade Environment =	Any Value	Condition = Any Va	ue Permit	Originated	Not Applicab
		GradeBook = Grad							Originated	

Fig. 9. TAPolicy: Match Results (GradeBook\_One)

Result(s) with se	elected verification (Policy Verification (Jun	e 14, 2018 14:58:16))		2	rows out of	2		Search	
Requirement S	ichema Subject		Resource	Action	E	Invironment	Condition	Decision	Verification Result
Test Case	1 TA = Graduate_C & Grad = Grad	duateStudent_C GradeBoo	ok = GradeBook_One	UniActions = ViewGrade	Enviror	nment = Any Value	Condition = Any	Value Permit	TRUE
Test Case	1 TA = Graduate_C & Grad = Grad	duateStudent_C GradeBoo	ok = GradeBook_Two	UniActions = ViewGrade	Enviror	nment = Any Value		Value Permit	FALSE
Policy(s) and M	latching result against the selcted security re	equirement:		1 rows	out of 1			Search	
	Policy Name	Rule Combination A	Algorithm	Poli	cy Enforcem	ent Algorithm		Combined Re	sult
	ABAC : TAPolicy	Deny-overric	les		Deny Bi	ased		Deny	
Rule(s) and Mat	tching result of Selected Policy against the s	elcted security requirement:			▶ 6 rows out	of 6		Search	
Sequence No	Subject	Resource	Action	Environr	ment	Condition	Decision	Inheritance Relation	Match Result
1	Subject = Any Value & TA = Graduate C	GradeBook = GradeBook_Or	ne UniActions = Vie	wGrade Environment =	Any Value	Condition = Any Valu	e Permit	Originated	Not Applicable
2	Subject = Any Value & TA = Graduate_C	GradeBook = GradeBook_Or	ne UniActions = Ass	ignGrade Environment =	Any Value	Condition = Any Valu	e Permit	Originated	Not Applicable
3	Subject = Any Value & TA = Graduate_C	GradeBook = GradeBook_Tw	Action = Any	Value Environment =		Condition = Any Valu	e Deny	Originated	Deny
4	Subject = Any Value & TA = TA_Two	GradeBook = GradeBook_Tw	vo UniActions = Vie	wGrade Environment =	Any Value	Condition = Any Valu	e Permit	Originated	Not Applicable
5	Subject = Any Value & TA = TA_Two	GradeBook = GradeBook_Tw		ignGrade Environment =	Any Value	Condition = Any Valu	Permit	Originated	Not Applicable
6	Subject = Any Value & TA = TA Two	GradeBook = GradeBook Or	Action = Any	Value Environment =		Condition = Any Valu	Denv	Originated	Not Applicable

Fig. 10. TAPolicy: Match Results (GradeBook\_Two)

Now that we have pinpointed our (4) 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 suitable based on your organizational needs.

For our example, let's assume this individual within both of our policies (GraduateStudent\_C) should be allowed to View GradeBook\_Two & GradeBook\_One, however, the individual should NOT be able to AssignGrade to GradeBook\_Two because within GradeBook\_Two is his own grade. He can still AssignGrade to GradeBook\_One because it is required for his TA duties. To resolve this, we will add 2 rules and delete 1 rule in both policies which will in turn resolve the Rule Conflict.

#### GradePolicy: Delete (1) Current Rule:

 $(Rule No. = 9) \rightarrow (Subject = Any Value \& Grad = GraduateStudent_C) \rightarrow (Action = Any Value) \rightarrow (Resource = GradeBook_One) \rightarrow decision = Deny$ 

#### GradePolicy: Add (2) New Rules:

 $\begin{array}{l} (\text{Rule No. = 9}) \rightarrow (\text{Subject = Any Value \& Grad = GraduateStudent_C}) \rightarrow (\text{Action = ViewGrade}) \rightarrow (\text{Resource = GradeBook_One}) \\ \rightarrow \text{ decision = Permit} \\ (\text{Rule No. = 10}) \rightarrow (\text{Subject = Any Value \& Grad = GraduateStudent_C}) \rightarrow (\text{Action = AssignGrade}) \rightarrow (\text{Resource = GradeBook_One}) \\ \rightarrow \text{ decision = Permit} \end{array}$ 

#### **TAPolicy: Delete (1) Current Rule:**

 $(Rule No. = 3) \rightarrow (Subject = Any Value \& TA = GraduateStudent_C) \rightarrow (Action = Any Value) \rightarrow (Resource = GradeBook_Two) \rightarrow decision = Deny$ 

#### TAPolicy: Add (2) New Rules:

 $(Rule No. = 6) \rightarrow (Subject = Any Value \& TA = GraduateStudent_C) \rightarrow (Action = ViewGrade) \rightarrow (Resource = GradeBook_Two) \rightarrow decision = Permit$  $(Rule No. = 7) \rightarrow (Subject = Any Value \& TA = GraduateStudent_C) \rightarrow (Action = AssignGrade) \rightarrow (Resource = GradeBook_Two) \rightarrow decision = Permit$ 

9 Subject=AnyValue & Grad = GraduateStudent\_C GradeBook = GradeBook\_One Action = AnyValue Environment = AnyValue Condition = AnyValue Deny Originated

Fig. 11. GradePolicy: Delete Rule (9)



Fig. 12. GradePolicy: Add New Rules

3 ISubject = Any Value & TA = Graduate\_C GradeBook = GradeBook\_Two Action = Any Value Environment = Any Value Condition = Any Value Deny Originated

Fig. 13. TAPolicy: Delete Rule (3)

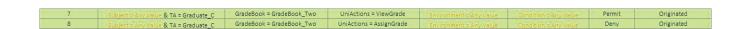


Fig. 14. TAPolicy: Add New Rules

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

olicy Verification (.	une 19, 2018 18:30:28)(s) Summary			1 rows out of 1	1		Sear	rch	
Status	Name	Verification Type	Verification Tech	nique Number of Po	licy(s) Combination Alg	orithm Enforcement	t Algorithm	Policy List	
UpToDate	Policy Verification (June 19, 2018 18:30:28)	Standard	Single Policy	γ 1	Deny-overrid	les Deny Bi	iased	ABAC:GradePe	licy
	ted verification (Policy Verification (June 19, 201				ows out of 2		Searc		l.
			esource	@ 2 r Action	ows out of 2 Environment	Condition	Searc	ch Verification R	IT
Result(s) with selec Requirement Sche Test Case 1		Ri	esource = GradeBook_One			Condition	1		l.

Fig. 15. Updated Results: GradePolicy (No Rule Conflict)

olicy Verification (.	une 19, 2018 18:29:43)(s) Summary			1 rows out of 1			Sear	ch 🚺	-
Status	Name	Verification Type	Verification Techni	ique Number of P	olicy(s) Combination	Algorithm Enforcem	ent Algorithm	Policy List	Ī
UpToDate	Policy Verification (June 19, 2018 18:29:43)	Standard	Single Policy	1	Deny-ove	rrides Den	y Biased	ABAC:TAPoli	y
	ted verification (Policy Verification (June 19; 2018	( 18:29:43))		<b>∞</b> 2 rc	ows out of 2		Search		•
			ource	∞ 2 ro Action	ows out of 2 Environment	Condition	Search	h Verification Res	100
Result(s) with selec Requirement Sche Test Case 1		Reso				Condition Condition = Any Value			100

Fig. 16. Updated Results: TAPolicy (No Rule Conflict)

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

This university example contains two policies (GradePolicy & TAPolicy). The attributes in this example have been changed slightly from previous Test Case 1. TA's attribute value has been changed from "GraduateStudent\_C" to "TA\_One" and also GradeBook has gained a new attribute value called "GradeBook\_Three." The Attribute/Attribute Values included in these policies are as shown in Figure 17.

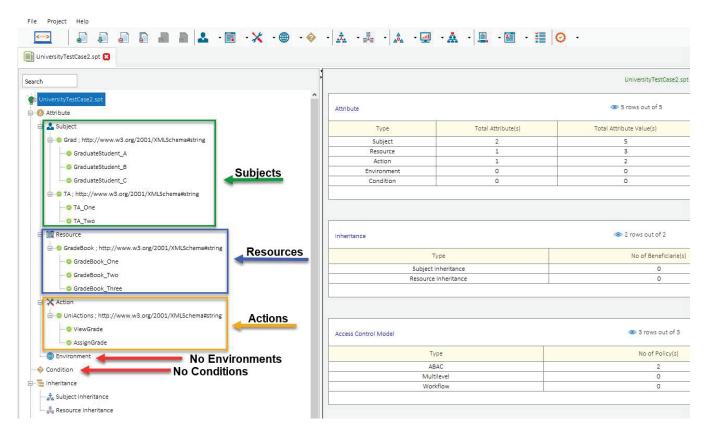


Fig. 17. Test Case 2

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

Now that we have entered our attributes we can model our two policies (GradePolicy & TAPolicy). 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:

#### GradePolicy:

- $(Grad = GraduateStudent_A, ViewGrade, GradeBook_One) \rightarrow Permit$
- $(Grad = GraduateStudent_A, AssignGrade, GradeBook_One) \rightarrow Deny$
- (Grad = GraduateStudent\_A, Any Action, GradeBook\_Two)  $\rightarrow$  Deny
- $(Grad = GraduateStudent_B, ViewGrade, GradeBook_One) \rightarrow Permit$
- $(Grad = GraduateStudent_B, AssignGrade, GradeBook_One) \rightarrow Deny$
- (Grad = GraduateStudent\_B, Any Action, GradeBook\_Two)  $\rightarrow$  Deny
- (Grad = GraduateStudent\_C, ViewGrade, GradeBook\_Two) →Permit
- $(Grad = GraduateStudent_C, AssignGrade, GradeBook_Two) \rightarrow Deny$
- (Grad = GraduateStudent\_C, Any Action, GradeBook\_One) →Deny

#### **TAPolicy**:

- (TA = TA\_One, ViewGrade, GradeBook\_One) →Permit
- (TA = TA\_One, AssignGrade, GradeBook\_One) →Permit
- (TA = TA\_One, Any Action, GradeBook\_Two) →Deny
- (TA = TA\_Two, ViewGrade, GradeBook\_Two) →Permit
- (TA = TA\_Two, AssignGrade, GradeBook\_Two) →Permit
- $(TA = TA_Two, Any Action, GradeBook_One) \rightarrow Deny$

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: UniversityTestCase2 and these policies will have been already created for you.

SradePolicy Policy(s	i) Summary		I rows out of 1	î.			Search
Model	Policy Name	Rule Combination Algorithm	Policy Enforcement Algorithm	No. of Rule(s)	Time Created		Last Modified
ABAC	GradePolicy	Deny-overrides	Deny Biased	9	June 14, 2018 11:54:1	1	June 14, 2018 11:54:11
	n selected policy (GradePolicy): Subject	Resource	Action	vs out of 9 Environment	Condition	Decision	Search
Sequence No							
1	Grad = GraduateStudent_A	GradeBook = GradeBook_One	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated
2	Grad = GraduateStudent_A	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated
3	Grad = GraduateStudent_A	GradeBook = GradeBook_Two	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated
	Grad = GraduateStudent_A Grad = GraduateStudent_B	GradeBook = GradeBook_Two GradeBook = GradeBook_One	Action = Any Value UniActions = ViewGrade	Environment = Any Value Environment = Any Value	Condition = Any Value Condition = Any Value	Deny Permit	Originated Originated
3							
3 4	Grad = GraduateStudent_B	GradeBook = GradeBook_One	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated
3 4 5	Grad = GraduateStudent_B Grad = GraduateStudent_B	GradeBook = GradeBook_One GradeBook = GradeBook_One	UniActions = ViewGrade UniActions = AssignGrade	Environment = Any Value Environment = Any Value Environment = Any Value	Condition = Any Value Condition = Any Value Condition = Any Value	Permit Deny	Originated Originated
3 4 5 6	Grad = GraduateStudent_B Grad = GraduateStudent_B Grad = GraduateStudent_B	GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_Two	UniActions = ViewGrade UniActions = AssignGrade Action = Any Value	Environment = Any Value Environment = Any Value	Condition = Any Value Condition = Any Value	Permit Deny Deny	Originated Originated Originated

## Fig. 18. GradePolicy

Policy Policy(s) Sur	mmary		1 rows or	ut of 1			Search	×
Model	Policy Name	Rule Combination Algorithm Policy Enforcement Algorithm No. of Rule(s)		Time Created		Last Modified		
ABAC	TAPolicy	Deny-overrides	Deny Biased	6	June 14, 2018 11:5	57:50	June 14, 2018 11:57:	50
e (s) defined with s	selected policy (TAPolicy	):	۲	6 rows out of 6			Search	×
						_		
	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Rela	tion
Sequence No	Subject TA = TA_One	Resource GradeBook = GradeBook_One	Action UniActions = ViewGrade	Environment Environment = Any Value	Condition = Any Value	Permit	Inheritance Rela Originated	tion
Sequence No	Subject TA = TA_One TA = TA_One	Resource GradeBook = GradeBook_One GradeBook = GradeBook_One	Action	Environment		Permit Permit	Inheritance Rela Originated Originated	tion
Sequence No	Subject TA = TA_One	Resource GradeBook = GradeBook_One	Action UniActions = ViewGrade	Environment Environment = Any Value	Condition = Any Value	Permit	Inheritance Rela Originated	ition
Sequence No 1 2	Subject TA = TA_One TA = TA_One	Resource GradeBook = GradeBook_One GradeBook = GradeBook_One	Action UniActions = ViewGrade UniActions = AssignGrade	Environment Environment = Any Value Environment = Any Value	Condition = Any Value Condition = Any Value	Permit Permit	Inheritance Rela Originated Originated	ition
Sequence No 1 2 3	Subject TA = TA_One TA = TA_One TA = TA_One	Resource GradeBook = GradeBook_One GradeBook = GradeBook_One GradeBook = GradeBook_Two	Action UniActions = ViewGrade UniActions = AssignGrade Action = Any Value	Environment Environment = Any Value Environment = Any Value Environment = Any Value	Condition = Any Value Condition = Any Value Condition = Any Value	Permit Permit Deny	Inheritance Rela Originated Originated Originated	ition

## Fig. 19. TAPolicy

## 9 INDIVIDUAL SECURITY REQUIREMENTS - 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:**

(TA = TA\_One) & (Action = Any) & (GradeBook = GradeBook\_Three)  $\rightarrow$  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: UniversityTestCase2 and this requirement will have been already created for you.

Test Case 2(s) Summary			I rows out of 1	Search			
A	Access Control Security Requirement		Requirement Schema		No. of Security Requirement(s)		
	Individual		Test Case 2		1		
Security Requirement (s) (	defined under selected Requin	ement Schema (Test Case 2):		I rows out of 1	Search	n 🖸 4	
Security Requirement (s) a	defined under selected Require	ement Schema (Test Case 2): Resource	Action	I rows out of 1 Environment	Search	n Eecision	

#### Fig. 20. Individual Security Requirement

# **10** 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 university policies; the author created a resource "GradeBook\_Three" 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 GradePolicy & TAPolicy as a Combined Policy Verification and analyze our verification result. Again, this will have already been done for you if you open Project File: UniversityTestCase2.

olicy Verifica	ation (June 14, 201	8 12:17:19)(s) Summary			I rows out of 1				Search	×	÷
Status		Name	Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement Algorith	m Policy List			_
JpToDate	Policy Verificati	ication (June 14, 2018 12:17:19) Standard		Combined Policy	2	Deny-overrides	Deny Biased	ABA	AC:GradePolicy, ABA	C:TAPolicy	y
lesult(s) with	h selected verificati	on (Policy Verification (June 14	, 2018 12:17:19))		@ 1r	ows out of 1			Search	×B (	
	h selected verificati	on (Policy Verification (June 14 Subject	, 2018 12:17:19)) Resource	Action	In Enviro		Condition	Decision	Search		

Fig. 21. 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 = GradeBook\_Three. 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 Verificatio	n (June 14, 2018 12:17:19)(s) Su	mmary		I rows out of 1			Sea	rch 🚺
Status	Name	Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement A	lgorithm	Policy List
JpToDate	Policy Verification (June 14, 2018	3 12:17:19) Standard	Combined Policy	2	Deny-overrides	Deny Bias	sed ABAC:Grad	ePolicy, ABAC:TAPolicy
Result(s) with se	elected verification (Policy Verifica	ition (June 14, 2018 12:17:19))		@ 1 ro	ows out of 1		Sear	ch 🚺 🕯
Requiremen	Schema Subject	Resource	Action	Environ	iment C	ondition	Decision	Verification Result
Test Ca	e 2 TA = TA One	GradeBook = GradeBook	Three Action = Any V	alue Environment:	- Any Maline Condition	on = Any Value	Permit	FALSE
Seq					2	10 <sup>-1</sup>	Com	
Seq	Jence No	Policy Name	Rule Combination	n Algorithm	Policy Enforcement	Algorithm	Comi	pined Result
	2	ABAC : GradePolicy ABAC : TAPolicy	Deny-over Denv-over	12	Deny Biase Deny Biase			Deny
Rule(s) and Mat	i da la companya da l	ainst the selcted security requirement	and a line of the		9 rows out of 9		Sear	ch 🛛
Sequence No	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Relation	Match Result
1	Grad = GraduateStudent A	GradeBook = GradeBook One	UniActions = ViewGrade	Environment = Any Valu	ue Condition = Any Value	Permit	Originated	Not Applicable
2	Grad = GraduateStudent_A	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Valu			Originated	Not Applicable
(1 m)	Grad = GraduateStudent_A	GradeBook = GradeBook_Two	Action = Any Value	Environment = Any Valu		Deny	Originated	Not Applicable
3		GradeBook = GradeBook One	UniActions = ViewGrade	Environment = Any Valu		1000 C	Originated	Not Applicable
3	Grad = GraduateStudent_B	Gradebook - Gradebook_one						
	Grad = GraduateStudent_B Grad = GraduateStudent_B	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Val			Originated	Not Applicable
4	-	GradeBook = GradeBook_One GradeBook = GradeBook_Two	UniActions = AssignGrade Action = Any Value		ue Condition = Any Value	Deny	Originated Originated	
4 5	Grad = GraduateStudent_B	GradeBook = GradeBook_One	the second se	Environment = Any Valu	ue Condition = Any Value ue Condition = Any Value	Deny Deny		Not Applicable

Condition = Any Value

Environment = Any Value

Deny

Denv

Originated

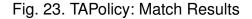
Originated

## Fig. 22. GradePolicy: Match Results

 Grad = GraduateStudent\_C
 GradeBook = GradeBook\_Two
 UniActions = AssignGrade

 Grad = GraduateStudent\_C
 GradeBook = GradeBook\_One
 Action = Any, Value

oncy vernication	(June 14, 2018 12:17	7:19)(s) Summary				1 rows out of 1	i			Search		×
Status	Nan	ne	Verification Type	Verification Tech	hnique Nur	mber of Policy(s)	Combination Algorith	m Enforceme	nt Algorithm	Po	licy List	
IpToDate Pe	olicy Verification (Jun	e 14, 2018 12:17:19)	Standard	Combined Po	olicy	2	Deny-overrides	Deny	Biased ABA	AC:GradePo	licy, ABAC:T	TAPolio
Result(s) with sele	ected verification (Pol	icy Verification (June 14, 2	2018 12:17:19))			@ 1r	rows out of 1			Search		×
Requirement S	chema S	Subject	Resource		Action	Enviro	nment	Condition	Decision	Ve	erification R	esult
Test Case	2 TA =	TA One Grade	Book = GradeBook_T	Three Action	n = Any Value	Environment	- Apy Malue	ndition = Any Value	Permit		FALSE	
	122						10,102 803-04				2009 DA	×
						2	100 (00 - 00 - <u>0</u> 0			100 110	705 10	-
Seque		Policy Na			mbination Algoriti	hm		ment Algorithm		Combine		
	1	ABAC : Grade	Policy	D	eny-overrides	ihm	Deny	Biased		Der	nγ	-
	1		Policy olicy	D			Deny				nγ	3
ule(s) and Match	1	ABAC : Grade ABAC : TAP	Policy olicy	D	eny-overrides eny-overrides		Deny Deny	Biased	Inheritance Relati	Der Der Search	nγ	×
ule(s) and Match	1 2 ning result of Selected	ABAC : Grade ABAC : TAP Policy against the selcted	Policy olicy	D D	eny-overrides ieny-overrides Env	vironment	Deny Deny 6 rows out of 6 Condition	Biased Biased Decision	Inheritance Relati	Der Der Search	ny ny Match Re	× sult
ule(s) and Match Sequence No	1 2 ning result of Selected Subject	ABAC : Grade ABAC : TAP Policy against the selcted Resource	Policy olicy I security requiremen pok_One Unit	D D It: Action	eny-overrides ieny-overrides Environm		Deny Deny 6 rows out of 6	Biased Biased Decision Permit		Der Der Search	יאי אי	x esuit cable
ule(s) and Match Sequence No 1	1 2 2 3 aning result of Selected Subject TA = TA_One	ABAC : Grade ABAC : TAP Policy against the selcted Resource GradeBook = GradeBo	Policy olicy I security requiremen pok_One UniA pok_One UniA	D D It: Action Actions = ViewGrade	eny-overrides eny-overrides Environm Environm	vironment tent = Any Value	6 rows out of 6 Condition	Biased Biased Decision Permit Permit	Originated	Der Der Search	ny ny Match Re Not Applie	sult able
ule(s) and Match Sequence No 1 2	1 2 2 3 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	ABAC : Grade ABAC : TAP Policy against the selcted Resource GradeBook = GradeBo GradeBook = GradeBo	Policy olicy I security requiremen pok_One UniA pok_One UniA	D D Action Actions = ViewGrade Actions = AssignGrade	eny-overrides eny-overrides Environm Environm Environm	vironment ment = Any Value ment = Any Value	Deny Deny 6 rows out of 6 Condition = Any Value Condition = Any Value	Blased Blased Decision Permit Permit Deny	Originated Originated	Search	Match Re Not Applic	sult able able
ule(s) and Match Sequence No 1 2 3	1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	ABAC : Grade ABAC : TAP ABAC : TAP Policy against the selcted Resource GradeBook = GradeBo GradeBook = GradeBook = GradeBook = GradeBook	Policy olicy d security requiremen bok_One UnitA bok_One UnitA bok_Two A bok_Two UnitA	D D Action Actions = ViewGrade Actions = AssignGrade Action = Any Value	eny-overrides eny-overrides Environm Environm Environm	vironment tent = Any Value tent = Any Value tent = Any Value	Deny Deny 6 rows out of 6 Condition = Any Value Condition = Any Value Condition = Any Value	Biased Biased Decision Permit Permit Perny Permit	Originated Originated Originated	Der Der Search	Match Re Not Applie Not Applie Not Applie	sult cable cable cable cable



#### **RESOLVING THIS ERROR - TEST CASE 2 (NOT PROTECTED RESOURCE)** 11

To eliminate a Not Protected Resource vulnerability the policy author would need to define a specific rule for the unprotected resource (GradeBook\_Three) and then test again to verify the intended access decision is being made based on this new rule design.

For example, if we're to add this rule below to the TAPolicy...

Not Applicable

Not Applicable

#### TAPolicy: Add (1) New Rule:

7 TA = TA\_One GradeBook = GradeBook\_Three

 $(Rule No. = 7) \rightarrow (TA = TA\_One) \rightarrow (Action = Any Value) \rightarrow (Resource = GradeBook\_Three) \rightarrow decision = Permit$ 

Fig. 24. TAPolicy: New Rule (7)

Then retest using the same Policy Verification selections as last time we will get the same False Verification result due to our Algorithm selections. However, we can see in the Match Results that we have provided a rule for the system to evaluate for TA\_One accessing this Resource.

Result(s) with sele	cted verification (Po	licy Verification (June 20, 2018 10:40:	(3))	1	rows out of 1			Search	×	
Requirement S	chema	Subject Resour	ce A	ction Envir	onment	Condition	Decision	Ver	ification Resu	ult
Test Case	2 TA	= TA_One GradeBook = Grad	eBook_Three Action	= Any Value Environme	nt = Any Value Con	dition = Any Value	Permit	-	FALSE	
Policy(r) and Mat		he selcted security requirement:		② 2 rows	out of 2			Search		
	Lining result against t	ne selcted security requirement.		2 10W3	000012			Search		
Seque	nce No	Policy Name	Rule Com	bination Algorithm	Policy Enforcem	ent Algorithm		Combined	Result	
	1	ABAC : GradePolicy	Der	ny-overrides	Deny B	iased		Deny	1	-
	2	ABAC : TAPolicy	Der	ny-overrides	Deny B	iased		Perm	it	
ule(s) and Match	ing result of Selected	d Policy against the selcted security req	uirement:		> 7 rows out of 7			Search	X	
Sequence No	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Relat	ion	Match Resu	lt
1	TA = TA_One	GradeBook = GradeBook_One	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	N	lot Applica	ble
2	TA = TA_One	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	N	lot Applica	Ы
3	TA = TA_One	GradeBook = GradeBook_Two	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated	N	lot Applica	ы
4	TA = TA_Two	GradeBook = GradeBook_Two	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	N	lot Applica	Ы
5	TA = TA_Two	GradeBook = GradeBook_Two	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	N	lot Applica	ble
	TA = TA Two	GradeBook = GradeBook One	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated	N	lot Applica	bl
6			Accord - Any value	anninene miny raide					or reprice	

Fig. 25. Updated Policy: Resource Now Protected

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

This university example contains two policies (GradePolicy & TAPolicy). The attributes in this example have not been changed from previous Test Case 2. The Attribute/Attribute Values included in these policies are as shown in Figure 26.

Permit Originated Permit

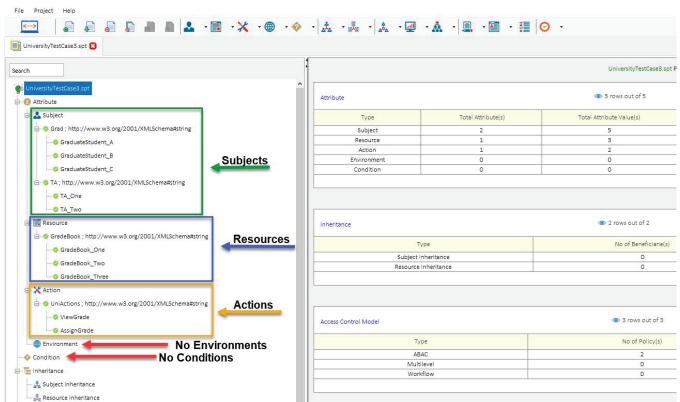


Fig. 26. Test Case 3

#### MODELING YOUR POLICY - TEST CASE 3 (UNDECIDED RULE) 13

Now that we have entered our attributes we can model our two policies (GradePolicy & TAPolicy). 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:

#### GradePolicy:

- (Grad = GraduateStudent\_A, ViewGrade, GradeBook\_One) → Permit (Grad = GraduateStudent\_A, AssignGrade, GradeBook\_One) →Deny (Grad = GraduateStudent\_A, Any Action, GradeBook\_Two) →Deny (Grad = GraduateStudent\_A, ViewGrade, GradeBook\_Three) →Permit (Grad = GraduateStudent\_A, AssignGrade, GradeBook\_Three) →Deny (Grad = GraduateStudent\_B, ViewGrade, GradeBook\_One) →Permit (Grad = GraduateStudent\_B, AssignGrade, GradeBook\_One) →Deny (Grad = GraduateStudent\_B, Any Action, GradeBook\_Two) →Deny  $(Grad = GraduateStudent_B, ViewGrade, GradeBook_Three) \rightarrow Permit$  $(Grad = GraduateStudent_B, AssignGrade, GradeBook_Three) \rightarrow Deny$ (Grad = GraduateStudent\_C, ViewGrade, GradeBook\_Two) →Permit (Grad = GraduateStudent\_C, AssignGrade, GradeBook\_Two) →Deny (Grad = GraduateStudent\_C, Any Action, GradeBook\_One) →Deny (Grad = GraduateStudent\_C, ViewGrade, GradeBook\_Three) →Permit (Grad = GraduateStudent\_C, AssignGrade, GradeBook\_Three) →Deny TAPolicy:  $(TA = TA\_One, ViewGrade, GradeBook\_One) \rightarrow Permit$
- $(TA = TA_One, AssignGrade, GradeBook_One) \rightarrow Permit$
- (TA = TA\_One, Any Action, GradeBook\_Two) →Deny
- (TA = TA\_One, ViewGrade, GradeBook\_Three) →Permit
- (TA = TA\_One, AssignGrade, GradeBook\_Three) →Permit
- (TA = TA\_Two, ViewGrade, GradeBook\_Two) →Permit
- $(TA = TA_Two, AssignGrade, GradeBook_Two) \rightarrow Permit$  $(TA = TA_Two, Any Action, GradeBook_One) \rightarrow Deny$

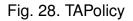
14

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: UniversityTestCase3 and these policies will have been already created for you.

radePolicy Policy(	s) Summary		I rows out of 1				Search
Model	Policy Name	Rule Combination Algorithm	Policy Enforcement Algorithm	No. of Rule(s)	Time Created		Last Modified
ABAC	GradePolicy	Deny-overrides	Deny Biased	15	June 14, 2018 11:54:1		une 14, 2018 11:54:11
ule (s) defined with	n selected policy (GradePolicy):		15 rows	s out of 15			Search 🚺
Sequence No	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Relation
1	Grad = GraduateStudent_A	GradeBook = GradeBook_One	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated
2	Grad = GraduateStudent_A	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated
3	Grad = GraduateStudent_A	GradeBook = GradeBook_Two	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated
4	Grad = GraduateStudent_B	GradeBook = GradeBook_One	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated
5	Grad = GraduateStudent_B	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated
6	Grad = GraduateStudent_B	GradeBook = GradeBook_Two	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated
7	Grad = GraduateStudent_C	GradeBook = GradeBook_Two	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated
8	Grad = GraduateStudent_C	GradeBook = GradeBook_Two	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated
9	Grad = GraduateStudent_C	GradeBook = GradeBook_One	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated
10	Grad = GraduateStudent_A	GradeBook = GradeBook_Three	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated
11	Grad = GraduateStudent_A	GradeBook = GradeBook_Three	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated
12	Grad = GraduateStudent_B	GradeBook = GradeBook_Three	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated
13	Grad = GraduateStudent_B	GradeBook = GradeBook_Three	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated
14	Grad = GraduateStudent_C	GradeBook = GradeBook_Three	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated
15	Grad = GraduateStudent C	GradeBook = GradeBook Three	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated

## Fig. 27. GradePolicy

TAPolicy Policy(s) Sur	nmary		1 rows out	t of 1			Search	×
Model	Policy Name	Rule Combination Algorithm	Policy Enforcement Algo	ithm No. of Rule(s)	Time Created		Last Modified	
ABAC	TAPolicy	Deny-overrides	Deny Biased	8	June 14, 2018 11:5	7:50	June 14, 2018 11:57:	50
Rule (s) defined with :	selected policy (TAPolicy	/): Besource	Action	8 rows out of 8 Environment	Condition	Decision	Search	
1	TA = TA One	GradeBook = GradeBook_One	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	
2	TA = TA_One	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	
3	TA = TA_One	GradeBook = GradeBook_Two	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated	1
4	TA = TA_Two	GradeBook = GradeBook_Two	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	
5	TA = TA_Two	GradeBook = GradeBook_Two	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	
	TA = TA Two	GradeBook = GradeBook_One	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated	
6	10-10-100							
6	TA = TA_One	GradeBook = GradeBook_Three	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	



## 14 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:

 $(TA = TA\_Two) \& (Action = ViewGrade) \& (GradeBook = GradeBook\_Three) \rightarrow decision = Permit$ 

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: UniversityTestCase3 and this requirement will have been already created for you.

est Case 3(s) Summary		I rows out of 1 Search					
A	ccess Control Security Require	ement	Requirement Schema		No. of Security Requirement(s)		
	Individual		Test Case 3		1		
curity Requirement (s)	defined under selected Requi	rement Schema (Test Case 3):	٩	1 rows out of 1	Searc	h 🚺	
curity Requirement (s) f	defined under selected Requi	rement Schema (Test Case 3): Resource	Action	> 1 rows out of 1 Environment	Searc	h Ecision	



## 15 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 university policies; the author created rules for all Subjects to access "GradeBook\_Three" but did not define access rules for TA = TA\_Two. In this situation, if TA\_Two were to attempt to take action on "GradeBook\_Three," the system would be forced to make a default decision instead of a defined 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 GradePolicy & TAPolicy as a Combined Policy Verification and analyze our verification result. Again, this will have already been done for you if you open Project File: UniversityTestCase3.

olicy Verifica	ation (June 14, 2018 12:28:55)(s) Summary			I rows out of 1	L			Search	
Status	Name	Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement Algorithm	m	Policy List	
JpToDate	Policy Verification (June 14, 2018 12:28:55)	Standard	Combined Policy	2	Deny-overrides	Deny Biased	ABAC:	GradePolicy, ABAC:TAPo	olicy
	n selected verification (Policy Verification (June 14		Artion		ows out of 1	Condition	L	Search	
Requireme	ent Schema Subject	4, 2018 12:28:55)) Resource =Book = GradeBook, Three	Action UniActions = View	Envi	ows out of 1	Condition	Decision	Search Verification Resul	

Fig. 30. 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 any Rules that are attached to Subject = TA\_Two taking action on Resource = GraduateBook\_Three. We can see this by noticing that every "Match Result" is "Not Applicable" whereas if there were Rules existing for TA\_Two and Resource = GraduateBook\_Three we would have at least seen one Rule with a (Permit or Deny) Match Result.

#### WWW.SECURITYPOLICYTOOL.COM

olicy Verific	ation (June 14, 20	018 12:28:55)(s) Summary			1 rows out of 1	i.			Search	×
Status		Name	Verification Type	Verification Technique	Number of Policy(s)	Combination Algorithm	Enforcement Algorith	m	Policy List	
JpToDate	Policy Verifica	ation (June 14, 2018 12:28:55)	Standard	Combined Policy	2	Deny-overrides	Deny Biased	ABAC	C:GradePolicy, ABAC	TAPolicy
	ient Schema	tion (Policy Verification (June 14	Resource	Action	Env	ironment	Condition	Decision	Search Verification	Result
	Case 3		Book = GradeBook_Three				lition = Any Value	Permit	FALSE	
olicy(s) and	d Matching result	against the selcted security requi	rement.		2 rows c	out of 2			Search	
	d Matching result Sequence No	against the selcted security requi		Rule Combination	2 rows of Algorithm	Policy Enforcemen	t Algorithm	1	Search Combined Result	
			lame	Rule Combination	Algorithm	10 Mar. 2010	5	9		

Rule(s) and Mate	thing result of Selected Policy aga	inst the selcted security requirement:		15 rows     15 rows	Search	×		
Sequence No	Subject	Resource	Action	Environment	Condition	Decision	Inheritance Relation	Match Result
1	Grad = GraduateStudent_A	GradeBook = GradeBook_One	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
2	Grad = GraduateStudent_A	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated	Not Applicable
3	Grad = GraduateStudent_A	GradeBook = GradeBook_Two	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated	Not Applicable
4	Grad = GraduateStudent_B	GradeBook = GradeBook_One	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
5	Grad = GraduateStudent_B	GradeBook = GradeBook_One	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated	Not Applicable
6	Grad = GraduateStudent_B	GradeBook = GradeBook_Two	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated	Not Applicable
7	Grad = GraduateStudent_C	GradeBook = GradeBook_Two	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
8	Grad = GraduateStudent_C	GradeBook = GradeBook_Two	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated	Not Applicable
9	Grad = GraduateStudent_C	GradeBook = GradeBook_One	Action = Any Value	Environment = Any Value	Condition = Any Value	Deny	Originated	Not Applicable
10	Grad = GraduateStudent_A	GradeBook = GradeBook_Three	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
11	Grad = GraduateStudent_A	GradeBook = GradeBook_Three	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated	Not Applicable
12	Grad = GraduateStudent_B	GradeBook = GradeBook_Three	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
13	Grad = GraduateStudent_B	GradeBook = GradeBook_Three	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated	Not Applicable
14	Grad = GraduateStudent_C	GradeBook = GradeBook_Three	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated	Not Applicable
15	Grad = GraduateStudent_C	GradeBook = GradeBook_Three	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Deny	Originated	Not Applicable

## Fig. 31. GradePolicy: Match Results

olicy Verification	(June 14, 2018 12:2)	8:55)(s) Summary			I rows out of	1			Search	×	
Status	Nar	me	Verification Type	Verification Techniq	ue Number of Policy(s)	Combination Algorith	m Enforcement	Algorithm	Policy List		
JpToDate P	ToDate Policy Verification (June 14, 2018 12:28:55) Standard				2	Deny-overrides	Deny Bi	ased ABA	C:GradePolicy, AE	BAC:TAPoli	ic
Result(s) with sele	ected verification (Po	licy Verification (June 14, 2	2018 12:28:55))		۰ 1	rows out of 1			Search	X	-
Requirement Schema		ubject	Resource	Act	ion En	vironment	Condition	Decision	Verificati	ion Result	
Test Case 3 TA =		TA Two GradeBook = GradeBook		Three UniActions = ViewGrade Environ		ent = Any Value	Condition = Any Value	Permit	Permit FALSE		-
		he selcted security requirer		Rule Combin	2 rows ation Algorithm		ement Algorithm		Search Combined Result		-
					10.5 10 2012						1
Seque	nce No	Policy Na ABAC : Grade	ime		ation Algorithm overrides	Policy Enforce	ement Algorithm r Biased		Search Combined Result Deny		
Seque	nce No	Policy Na	me Policy	Deny-	ation Algorithm	Policy Enforce Deny	-		Combined Resul		
Seque	ince No 1 2	Policy Na ABAC : Grade	ime Policy olicy	Deny- Deny-	ation Algorithm overrides	Policy Enforce Deny	Biased		Combined Resul	t	
Seque	ince No 1 2	Policy Na ABAC : Grade ABAC : TAPe	ime Policy olicy	Deny- Deny-	ation Algorithm overrides	Policy Enforc Deny Deny	Biased	Inheritance Relation	Combined Result Deny Deny Search	t	
Seque	ince No 1 2 hing result of Selected	Policy Na ABAC : Grade ABAC : TAPe ABAC : TAPe d Policy against the selcted	me Policy olicy d security requireme	Deny- Deny-	ation Algorithm overrides overrides	Policy Enforce Deny Deny 8 rows out of 8	Biased Biased	Inheritance Relati	Combined Resul Deny Deny Search	t	
Seque Rule(s) and Match Sequence No	hing result of Selected	Policy Na ABAC : Grade ABAC : TAP4 d Policy against the selcted Resource	me Policy olicy d security requireme pook_One U	Deny- Deny- nt: Action	etion Algorithm overrides overrides	Policy Enforce Deny Deny > 8 rows out of 8 Condition	Biased Biased Decision Je Permit	1410/04/05/05/05/05/05/05/05/05	Combined Resul Deny Deny Search On Mata	lt X	e
Seque ule(s) and Matcl Sequence No 1	ince No     1     2     ining result of Selected     Subject     TA = TA_One	Policy Na ABAC : Grade ABAC : TAPI d Policy against the selcted Resource GradeBook = GradeBo	me Policy olicy d security requireme pook_One U pook_One Ur	Deny- Deny- nt: Action niActions = ViewGrade	ation Algorithm overrides overrides Environment Environment = Any Value	Policy Enforce Deny Deny P 8 rows out of 8 Condition Condition = Any Value	Biased Biased Decision pe Permit pe Permit	Originated	Combined Resul Deny Deny Search On Mato Not Aj Not Aj	t Result	e
Seque ule(s) and Matcl Sequence No 1 2	Subject           TA = TA_One           TA = TA_One	Policy Na ABAC : Grade ABAC : TAPe ABAC : TAPe device against the selcted Resource GradeBook = GradeBc GradeBook = GradeBc	me Policy discurity requireme poly_One U poly_One Ur poly_One Ur	Deny- Deny- nt: Action niActions = ViewGrade iiActions = AssignGrade	ation Algorithm overrides overrides Environment Environment = Any Value Environment = Any Value	Policy Enforce Deny Deny > 8 rows out of 8 Condition Condition = Any Valu Condition = Any Valu	Biased Biased Decision re Permit re Permit re Deny	Originated Originated	Combined Resul Deny Deny Search Not Aj Not Aj Not Aj	t ch Result pplicable pplicable	e
Seque ule(s) and Matcl Sequence No 1 2 3	Subject           TA = TA_One           TA = TA_One           TA = TA_One	Policy Na ABAC : Grade ABAC : TAPr d Policy against the selcted Resource GradeBook = GradeBook = Grade	me Policy discurity requireme book_One U pook_One U pook_Two U	Deny- Deny- Deny- nt: Action = ViewGrade liActions = AssignGrade Action = Any Value	etion Algorithm overrides overrides Environment Environment = Any Value Environment = Any Value Environment = Any Value	Policy Enforce Deny Deny > 8 rows out of 8 Condition = Any Valu Condition = Any Valu Condition = Any Valu	Biased Biased Decision Permit Permit Permit Permit Permit	Originated Originated Originated	Combined Result Deny Deny Search Search Not A; Not A; Not A;	t ch Result pplicable pplicable pplicable	
Seque Rule(s) and Matcl Sequence No 1 2 3 4	Subject           TA = TA_One           TA = TA_One           TA = TA_One           TA = TA_One	Policy Na ABAC : Grade ABAC : TAPr d Policy against the selcted Resource GradeBook = GradeBo GradeBook = GradeBook	ime Policy d security requireme bok_One U bok_One Ur bok_Two bok_Two Ur bok_Two Ur	Action Action nlActions = ViewGrade ilActions = AssignGrade Action = Any Value nlActions = ViewGrade	etion Algorithm overrides overrides Environment = Any Value Environment = Any Value Environment = Any Value Environment = Any Value	Policy Enforce Deny Deny 28 rows out of 8 Condition = Any Valu Condition = Any Valu Condition = Any Valu Condition = Any Valu	Biased Biased Decision pe Permit pe Permit pe Permit pe Permit pe Permit	Originated Originated Originated Originated	Combined Result Deny Deny Search Not Aj Not Aj Not Aj Not Aj	t ch Result pplicable	
Seque Aule(s) and Matcl Sequence No 1 2 3 4 5	Subject           TA = TA_One           TA = TA_Tone           TA = TA_Tone	Policy Na ABAC : Grade ABAC : Grade ABAC : TAPI d Policy against the selcted Resource GradeBook = GradeBo GradeBook = GradeBo GradeBook = GradeBo GradeBook = GradeBo	me Policy disecurity requireme ook_One U ook_One Ur ook_Two ook_Two U ook_Two U ook_One U	Action Action Actions = ViewGrade ilActions = Any Value Action = Any Value IlActions = AssignGrade ilActions = AssignGrade	ation Algorithm overrides overrides Environment Environment = Any Value Enviro	Policy Enforce Deny Deny Deny Condition Condition = Any Valu Condition = Any Valu Condition = Any Valu Condition = Any Valu	Biased Biased Permit Permit Permit Permit Permit Permit Permit Permit Permit	Originated Originated Originated Originated Originated	Combined Result Deny Deny Search Not Aj Not Aj Not Aj Not Aj	t ch Result pplicable pplicable pplicable pplicable	

## Fig. 32. TAPolicy: Match Results

As you can see there has not been a rule defined for TA\_Two  $\rightarrow$  Action  $\rightarrow$  GraduateBook\_Three which is known as an Undecided Rule error.

## 16 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 (e.g., include TA\_Two) in any policies that determine TA access requests to GraduateBook\_Three.

For example, adding these rules below to the TAPolicy for our specific example...

#### TAPolicy: Add (2) New Rules:

 $(Rule No. = 9) \rightarrow (TA = TA_Two) \rightarrow (Action = ViewGrade) \rightarrow (Resource = GradeBook_Three) \rightarrow decision = Permit$  $(Rule No. = 10) \rightarrow (TA = TA_Two) \rightarrow (Action = AssignGrade) \rightarrow (Resource = GradeBook_Three) \rightarrow decision = Permit$ 

9	TA = TA_Two	GradeBook = GradeBook_Three	UniActions = ViewGrade	Environment = Any Value	Condition = Any Value	Permit	Originated
10	TA = TA_Two	GradeBook = GradeBook Three	UniActions = AssignGrade	Environment = Any Value	Condition = Any Value	Permit	Originated

#### Fig. 33. TAPolicy: New Rules (9,10)

Now, looking out 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.

For example, GradePolicy has no rules related to the security requirement (TA\_Two  $\rightarrow$  View-Grades  $\rightarrow$  GradeBook\_Three) 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 GradePolicy = Deny. However, in the case of the TAPolicy 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 (GradePolicy = Deny; TAPolicy = Permit). Finally, the Combination Algorithm = Deny-overrides makes a definitive answer for our Verification Results. The Deny-overrides selection overrules the Permit result from the TAPolicy in favor of the Deny result from the GradePolicy to make the final Verification Result = False.

olicy Verification	(June 20, 2018 11:1	14:24)(s) Summary			I rows out of	1			Search	×	
Status	Name		Verification Type	Verification Technique	Number of Policy(s)	Combination Algorith	m Enforcement	Algorithm	Policy List		
pToDate P	olicy Verification (Ju	ine 20, 2018 11:14:24)	Standard	Combined Policy	2	Deny-overrides	Deny B	iased AB	ABAC:GradePolicy, ABAC:TAP		
Result(s) with sele	ected verification (Pe	olicy Verification (June 20	. 2018 11:14:24))		۹ ۱	rows out of 1			Search	X	1
Requirement Schema		Subject	Resource	Actio	n Env	vironment	Condition	Decision	Verificatio	on Result	
Test Case	3 TA :	TA Two Grade	Book = GradeBook Thr	e UniActions = \	iewGrade Environm	ent = Any Value	Condition = Any Value	Permit	FAL	SF	
	nce No	Policy N		Rule Combina		1	ement Algorithm		Combined Result		
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		the selcted security requir							Search	×	
						1			1000		
	2	ABAC : Grad ABAC : TA		Deny-o Deny-o			Biased		Deny Permit		_
ule(s) and Matc	hing result of Selecte	ed Policy against the selcte	ed security requirement		•	10 rows out of 10			Search	×	1
Sequence No	Subject	Resource		Action	Environment	Condition	Decision	Inheritance Rela	ance Relation Match R		-
1	TA = TA_One	GradeBook = Grade	Book_One Uni	Actions = ViewGrade	Environment = Any Value	Condition = Any Valu	Permit	Originated Not An		plicable	
2	TA = TA_One	GradeBook = Grade	Book_One UniA	ctions = AssignGrade	Environment = Any Value	Condition = Any Valu		- Hot rep		plicable	
3	TA = TA_One	GradeBook = Grade	Book_Two	ction = Any Value	Environment = Any Value	Condition = Any Valu	Deny	Originated	Not Ap	plicable	
4	TA = TA_Two	GradeBook = Grade	Book_Two Uni	Actions = ViewGrade	Environment = Any Value	Condition = Any Valu	-			plicable	-
5	TA = TA_Two	GradeBook = Grade	Book_Two UniA	ctions = AssignGrade	Environment = Any Value	Condition = Any Valu	e Permit			plicable	
6	TA = TA_Two	GradeBook = Grade	Book_One	ction = Any Value	Environment = Any Value	Condition = Any Valu	Je Deny	Originated	Not Ap	plicable	
7	TA = TA_One	GradeBook = GradeB	look_Three Uni	Actions = ViewGrade	Environment = Any Value	Condition = Any Valu	Je Permit	Originated		plicable	
8	TA = TA_One	GradeBook = GradeB	look_Three UniA	ctions = AssignGrade	Environment = Any Value	Condition = Any Valu	e Permit	Originated	Not Ap	plicable	
			and the second se								
9	TA = TA_Two	GradeBook = GradeB	look_Three Uni	Actions = ViewGrade		Condition = Any Valu	le Permit	Originated	Per	rmit	

Fig. 34. Updated Results: No Undecided Rule

## 17 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.