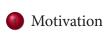
### Extending Access Control Models with Break-glass

Achim D. Brucker Helmut Petritsch {achim.brucker, helmut.petritsch}@sap.com



ACM Symposium on Access Control Models and Technologies (SACMAT 2009) Stresa, Italy, 5th June 2009

#### Outline



- A Generic Architecture Supporting Break-glass
- Conclusion and Future Work

#### **Our** Vision

## **Our** Vision

Assume,

- we are a nurse
- trying to access the patient record of Peter Meier ...



Break-glass or Overriding Access Control

#### Our Vision

# Override Access Control Access Denied - Your are not assigned to Peter Meier Peter Meier is a patient of Dr. Smith. You can contact Dr.<br/>Smith by phone (+49 761 203 6498) or send him a<br/>notification. Vou need to be assigned to the patient "Peter Meier" to be<br/>allowed to access his patient record. In case of<br/>emergency, you may override this restriction. All your actions will be logged for later audit! I agree that my actions are logged for later audit. Image: Cancel Notify Dr. Smith

Access Control Models with Break-glass

Motivation

## The Situation Today

Brucker and H. Petritsch (SAP Research)

Mostly implemented using pre-staged accounts that are

- either stored in sealed covers or
- electronically issued on request.

Break-glass solutions should cover

- the creation of break-glass accounts,
- the distribution pre-staged accounts,
- the monitoring of the use of break-glass accounts, and
- the cleanup after an break-glass situation.

This solution is

- quite coarse-grained and
- not integrated into regular access control.

SACMAT 2009 5

SACMAT 200

AD. Brucker and H. Petritsch (SAP Research) Access Control Models with Break-glass
Outline
Motivation
Break-glass: The Main Idea
A Generic Architecture Supporting Break-glass
Extending Model-driven Security
Conclusion and Future Work

While often motivated with

- health care or
- public security

scenarios, also enterprises demand break-glass solutions:

- for preventing stagnation on the system administration level and
- for preventing stagnation on the business process level.

In fact, state of the art enterprise systems support break-glass, e.g.,

- Virsa Firefighter for SAP,
- Oracle's Role Manager.

Break-glass: The Main Idea

#### Observations and Goals

- During discussions with end users, we observed:
  - depending on the situation, different overrides can be justified
  - some restrictions can never be overridden
- The two main design goals are:
  - access-control decisions should be overrideable on a per permission basis and
  - fine-grained configuration of the restrictions that can be overridden.

## Emergeny Levels

#### Definition

A policy *p* refines a policy p' (written  $p \equiv p'$ ) if and only if the set of system traces that are allowed under *p* is a subset of the system traces that are allowed under p'.

- A policy *p* refines a policy *p'* iff *p* is at least as restrictive as *p'*.
- $p^{\mathsf{T}}$  is the policy that allows all actions and
- $p^{\perp}$  is the policy that denies all actions.
- *p*<sup>⊥</sup> refines all policies and every policy is a refinement of *p*<sup>⊤</sup>.
- $P_A$  be the set of all policies of the access control model A.
- $(P_{\mathcal{A}}, \subseteq, p^{\perp}, p^{\top})$  is a lattice.

A.D. Brucker and H. Petritsch (SAP Research) Access Control Models with Break-glass SACMAT 2009 7 A.D. Brucker and H. Petritsch (SAP Research) Access Control Models with Break-glass

Break-glass: The Main Idea

## **Regular Policies and Emergeny Policies**

#### Definition

We refer to the *regular policy*, i. e., the policy that should be obeyed in normal operations, as  $p^{reg}$  and we refer to the set of policies that are refined by the regular policy, i. e.,

$$L_{\mathcal{A}} = \left\{ p \mid p \in P_{\mathcal{A}} \land p^{\text{reg}} \sqsubseteq p \land p \neq p^{\text{reg}} \right\}$$

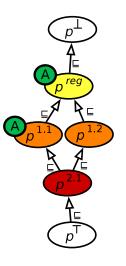
as *emergency levels* or *emergency policies* of the policy  $p^{\text{reg}}$ . We require that  $(P_A \smallsetminus p^{\perp}, \subseteq, p^{\text{reg}}, p^{\top})$  is a lattice, i. e.,  $\inf(P_A \smallsetminus p^{\perp}) = p^{\text{reg}}$ .

- An emergency level can be *active* or *inactive*.
- Only active emergency levels contribute to the access control decision.
- The regular policy is always active.

# Hierarchical Break-glass Access Control

Break-glass: The Main Ide

- An access that is only granted by an emergency policy *ℓ* ∈ *L<sub>A</sub>* is called *override access*.
- Override accesses are only granted if there is an active policy granting access.
- *Obligations* can be attached to an (emergency) policy, i.e., requiring user confirmations or for activating monitoring.
- By evaluating the policies in topological order, the refinement relation holds **by construction**!



SACMAT 2009

#### Outline

- A Generic Architecture Supporting Break-glass

#### Break-glass Architecture: Main Idea

The break-glass policy combination strategy can be *implemented by a meta PDP.* 

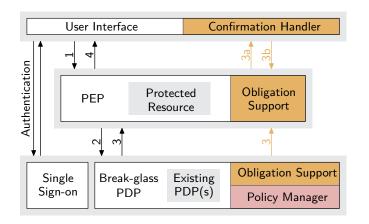
- The **Break-glass PDP** implements the break-glass policy combination strategy on top of existing PDPs
- User confirmations can be implemented using obligations:
  - the various PDPs need to support obligations
  - the various PEPs need to support obligations
  - the user interface needs to support confirmation requests

Break-glass does not impose restrictions on the underlying access control model!

SACMAT 2009

12

#### A Generic Architecture Supporting Break-glass A Generic Break-glass Architecture



# Outline

A.D. Brucker and H. Petritsch (SAP Research) Access Control Models with Break-glass

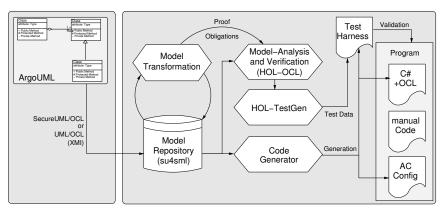
- Extending Model-driven Security

#### The Model-driven Security Vision

A Tool-supported and Security-aware Formal Model-driven Engineering Process

## The Model-driven Security Vision

A Tool-supported and Security-aware Formal Model-driven Engineering Process

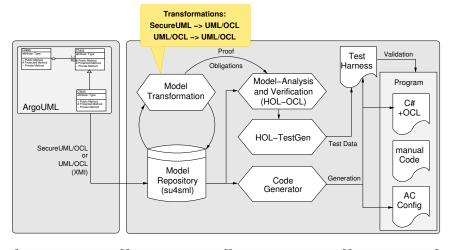


Design	Model Transformation	Verification and	Testing and	
Phase	Phase	Code-generation Phase	Deployment Phase	
A.D. Brucker and H. Petritsch (SAP Research)	Access Control Models	with Break-glass	SACMAT 2009	1

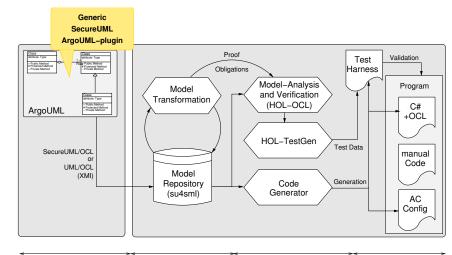
extending Model-driven Security

#### The Model-driven Security Vision

A Tool-supported and Security-aware Formal Model-driven Engineering Process



Design Phase	Model Transformation Phase	Verification a Code–generation	Testing and Deployment Phase	
A.D. Brucker and H. Petritsch (SAP Research)	Access Control Models	s with Break-glass	SACMAT 2009	15



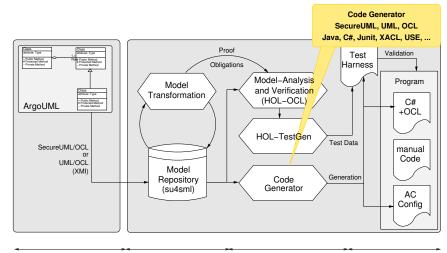
Design	Model Transformation	Verification and	Testing and	
Phase	Phase	Code-generation Phase	Deployment Phase	
A.D. Brucker and H. Petritsch (SAP Research)	Access Control Models	with Break-glass	SACMAT 2009	15

Extending Model-driven Security

## The Model-driven Security Vision

A.D

A Tool-supported and Security-aware Formal Model-driven Engineering Process



Design	Model Transformation	Verification and	Testing and	
Phase	Phase	Code-generation Phase	Deployment Phase	
D. Brucker and H. Petritsch (SAP Research)	Access Control Models	s with Break-glass	SACMAT 2009	15

#### The Model-driven Security Vision

A Tool-supported and Security-aware Formal Model-driven Engineering Process

#### HOL-OCL formal analysis formal verification Class wwbute: Type Proof Test Validation + Public Method # Protected Method Release Method Public Method Protected Method Obligations Harness Model-Analysis Program Model and Verification Class atribute: Type Transformation (HOL-OCL) C# ic Method ected Metho ArgoUML +OCL HOL-TestGen Test Data SecureUML/OCL manual Code UML/OCL Model (XMI Generation Repository Code Generator (su4sml) AC Config

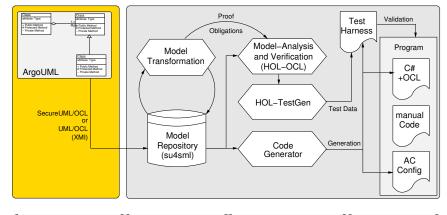
<del>، ، ، ، ، ، ، ، ، ، ، ، ، ، ، ، ، ، ، </del>	><	>*	<>	
Design	Model Transformation	Verification and	Testing and	
Phase	Phase	Code-generation Phase	Deployment Phase	
A.D. Brucker and H. Petritsch (SAP Research)	Access Control Models	with Break-glass	SACMAT 2009	15

Extending Model-driven Securit

A

## The Model-driven Security Vision

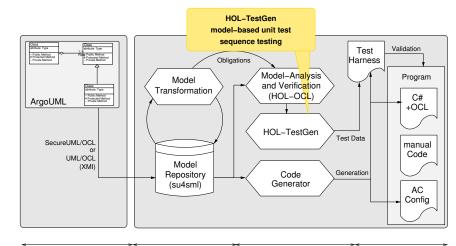
A Tool-supported and Security-aware Formal Model-driven Engineering Process



•	Design Phase	Model Transformation Phase	Verification and Code-generation Phase	Testing and Deployment Phase	
.D. Bruck	er and H. Petritsch (SAP Research)	Access Control Model	s with Break-glass	SACMAT 2009	15

### The Model-driven Security Vision

A Tool-supported and Security-aware Formal Model-driven Engineering Process



Design	Model Transformation	Verification and	Testing and	
Phase	Phase	Code-generation Phase	Deployment Phase	
A.D. Brucker and H. Petritsch (SAP Research)	Access Control Models v	vith Break-glass	SACMAT 2009	15

Extending Model-driven Security

## SecureUML

A.



#### SecureUML

- is a UML-based notation,
- provides abstract Syntax given by MOF compliant metamodel,
- is pluggable into arbitrary design modeling languages,
- is supported by an ArgoUML plugin.

16

Extending Model-driven Security

Modeling Access Control with SecureUML

#### SecureUML

#### Subject Role 0..\* Permission Action Resource ⚠ 0..\* $\supset 0.$ 0..\* 0... 1 \* ○0..\* 0 1 Policy Group User AuthorizationConstraint \* 0..\* Obligation AtomicAction CompositeAction 0..\*

SecureUML

- is a UML-based notation,
- provides abstract Syntax given by MOF compliant metamodel,

Access Control Models with Break-glass

SACMAT 2009

- is pluggable into arbitrary design modeling languages,
- is supported by an ArgoUML plugin.
- can easily be extended with support for **break-glass**.

					medicantecord	
	ml.policy» encyLevel	«secureuml.role» UserRole			<ul> <li>disease:String</li> <li>medication:String</li> </ul>	
LowEmerg	encyLevel	UserNole		 	 ■ read():OclVoid	
				i i	update():OclVoid	
4	7	4	[	«secureuml.permission»	<pre>create():OclVoid</pre>	
«secureu	ml.policy»	«secureuml.role»		OwnerMedicalRecord	0*	
HighEmer	gencyLevel	AdministratorRole		MedicalRecord:read	owner 1	
				<ul> <li>MedicalRecord:update</li> </ul>	Patient	
			Ň.	MedicalRecord:delete	name:String	
		caller=self.owner.name			8	

«secureuml.permission»

EmergencyOwnerMedicalRecord

MedicalRecord:read

Extending Model-driven Secur

#### ArgoUML Support

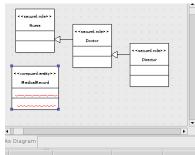
A.D. Brucker and H. Petritsch (SAP Research)

A.D. Bru

				nshot.zargo - Class D	iagram 1 - ArgoUML		_ D X
ile <u>E</u> dit ⊻iew <u>C</u> reate							
		4		i 🖬 🕽 🖨 🖓 🖬			
Package-centric	•	<u> </u>		+++	· ⊢ ↑   🖯 ↑   ↑	•   🗏 🗏 !	昱│ • ▼│ 🗅 … │ 🗆 ▼
rder By Type. Name   Profile Conflyance   Profile Conflyance   Cass Diagram   Lass	l ity ityy rtype raint ission r	<ul> <li>A</li> <li>A</li></ul>	Use	- · · · ·	amLentity>> saRecord	umt policy* * rergencyLavel	•••eeomf.solky>     •
Priority 🔻 86 Items	ToDo I	tem 🛥		Properties 🛦	Documentat	tion 🔺	Presentation A
📑 High	Source 🛓		raints 🛦	Stereotype 🛦	Tagged Values 🛓	Checklist	SecureUML Properties
E Medium Low				SecureUML Resource	ce: Entity MedicalRecord	🧐 New Role	
	DefaultPoli	cy ACT	TION		UserRole		AdministratorRole
	LowEmergency	/Level		create			2 8
	HighEmergenc	yLevel		read			. 9
	Create Poli	cy		delete			
				update			
				fullAccess			 _ &

#### ArgoUML Support

.D. Brucker and H. Petritsch (SAP Research) Access Control Models with Break-glass



 Properties
 Documentation
 Presentation
 Source

 Stereotype
 Tagged Values
 Checklist
 SecureUML Properties

ecureUML Resource: Entity MedicalRecord 🛛 😫 New Role

ACTION	Nurse	Doctor	Director
create			. 😫
read		. 8	
delete			
update			. 😫
fullAccess			- A

18

MedicalRecord

SACMAT 2009

17

#### Extending Model-driven Security

#### Code Generation (Java and XACML)

- In case of XACML, we can generate
  - the policies and
  - the PDP configuration.
- In particular, we
  - sort the policies topological,
  - use the "first-applicable" combining algorithm of XACML, and

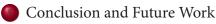
Access Control Models with Break-glass

- exploit the obligations support of XACML.
- With respect to the application, we generate
  - (stubs of) the business logic,
  - the calls to PDP, and
  - the PEP.

#### Outline



- Break-glass: The Main Idea
- A Generic Architecture Supporting Break-glass



Conclusion and Future Work

Conclusion and Future Work

#### We presented a

- a generic break-glass model that allows the fine-grained, overriding of access control decisions,
- an generic architecture for implementing break-glass,
- an extension of SecureUML supporting break-glass, and
- the mapping of break-glass to XACML

Future work includes the integration and development of

- analysis techniques for user providing feedback to the user,
- break-glass concepts for IT compliance, and
- techniques for a posteriori analysis of incidents.

# Thank you for your attention!

Any questions or remarks?

SACMAT 2009

19

# Bibliography

Achim D. Brucker and Helmut Petritsch.

Extending access control models with break-glass.

Bibliography

In ACM symposium on access control models and technologies (SACMAT), pages 197–206. ACM Press, 2009.

SACMAT 2009 23