Analyzing Access Control Overrides

Achim D. Brucker   Helmut Petritsch

{achim.brucker, helmut.petritsch}@sap.com

SAP Research Karlsruhe
Germany

International Workshop on Policies for the Future Internet
(PoFI 2011)
Pisa, Italy, 9th June 2011

Aniketos in a Nutshell

Aniketos: Make composite services able to establish and maintain security and trustworthiness

Goals of the Aniketos platform:

- Design-time discovery, composition and evaluation, threat awareness
- Runtime adaptation or change in service configuration
- Runtime monitoring, detection, notification

Two related dimensions:

- **Trustworthiness:** Reputation, perception, centralized vs. distributed
- **Security properties:** Behavior, contracts, interfaces, formal verification

ANIKETOS

http://www.aniketos.eu

Aniketos Fact-Sheet:

- EU Integrated Project (IP), FP7 Call 5
- Budget: € 13.9 Mio (€ 9.6 Mio funding)
- 42 month (Aug. 2010 – Feb. 2014)
- Coordinator: Sintef (Norway)
- Consortium: ATC, ATOS, DAEM, DBL, ELSAG, CNR, ITALTEL, LJMU, SAP, SEARCH-LAB, TECNALIA, THALES, TSSG, UNITN, PLUS, WIND

SAP applies and develops *formal methods* for *ensuring* the *security* and *technical trustworthiness* of services.

Outline

1 Motivation

2 An Architecture Supporting for Analyzing Access Control

3 Analyzing Access Control Overrides

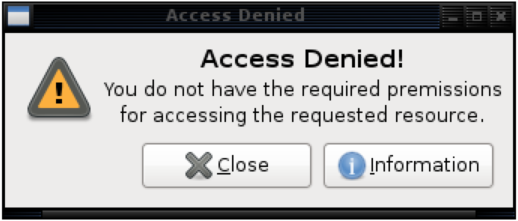
4 Conclusion

Our Vision

Assume,

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

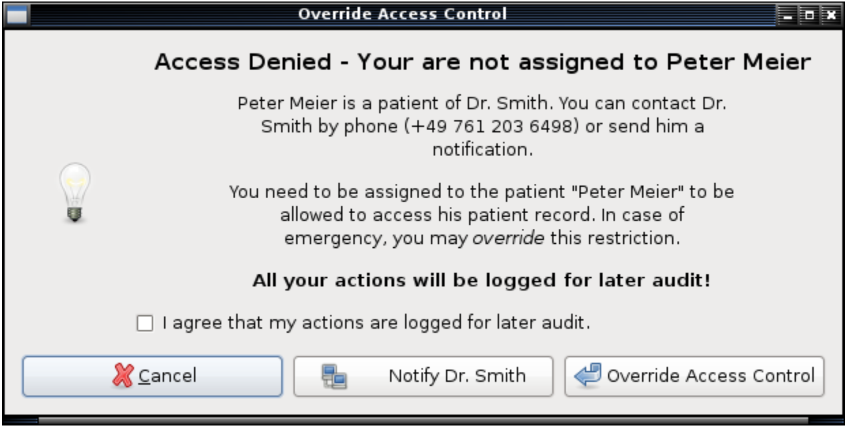
Our Vision



Access Denied!

You do not have the required premissions for accessing the requested resource.

Our Vision



Access Denied - Your are not assigned to Peter Meier

Peter Meier is a patient of Dr. Smith. You can contact Dr. Smith by phone (+49 761 203 6498) or send him a notification.

You need to be assigned to the patient "Peter Meier" to be allowed to access his patient record. In case of 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.

Motivation

Overriding Access Control

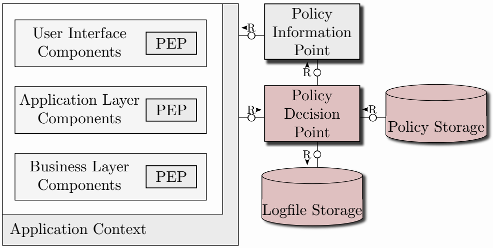
- Allows the user to temporarily extend his permissions
- Also known as Break-Glass or Break the Glass (BTG)

Relies on a post-hoc audit to evaluate the override

- Effort for auditing overrides increases costs
- Support auditor to reduce time and effort

Standard Architecture

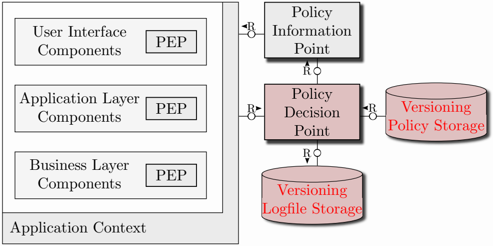
- Multiple PEPs accessing a central PDP
- Policies are loaded from a Policy Storage
- Policy Information Point (PIP) to resolve information from the application context
- Access control requests and results are stored in a Logfile Storage



Versioning

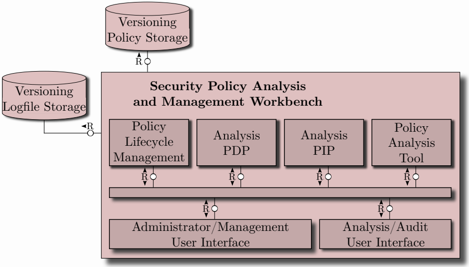
Based on XACML

- Store policies in a Versioning Policy Storage
- Save all PIP-resolved data in a Versioning Logfile Storage
  - XACML: resolved attributes
  - Save the current “state” of the system as seen by the PDP
- Interface for clients and PIP remains the same


User Interface Components
PEP
Application Layer Components
PEP
Business Layer Components
PEP
Application Context
R
Policy Information Point
R
R
Policy Decision Point
R
Versioning Policy Storage
R
Versioning Logfile Storage

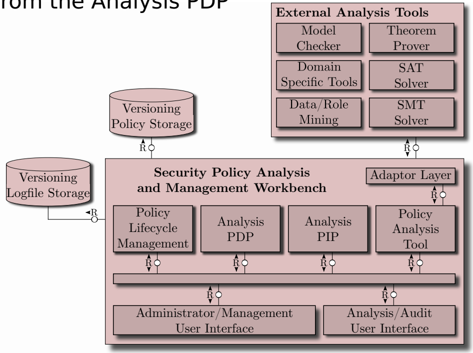
Analysis Workbench

- Analysis PDPs load any policy version from the policy store
- Analysis Policy Information Point (PIP) as context provider
  - Analysis PIP retrieves attributes from log store
  - Simulated runtime environment for analysis
- Replay (re-evaluate) recorded (or new) access control requests
- XACML engine analysis enhancements allow for advanced analysis methods
  - Debugging of Policies
  - Abstract evaluation
  - Policy animation



External Analysis Tools

- Integrate existing and new developed tools
- Provide interfaces to access policy and log store
- Load and use Analysis PDPs
  - Define or modify the simulated runtime environment
  - Retrieve evaluation events from the Analysis PDP
  - Browse the evaluation state


External Analysis Tools
Model Checker
Theorem Prover
Domain Specific Tools
SAT Solver
Data/Role Mining
SMT Solver
Versioning Policy Storage
Versioning Logfile Storage
R
Security Policy Analysis and Management Workbench
Adaptor Layer
Policy Lifecycle Management
Analysis PDP
Analysis PIP
Policy Analysis Tool
Administrator/Management User Interface
Analysis/Audit User Interface

Replay Access Control Requests

To replay an access control request

- Select log entry from the log store
- Instantiate an Analysis PDP with a policy version
- Replay request on Analysis PDP
- Analysis PDP retrieves attributes as recorded for this request via Analysis PIP from the log store

Support for understanding policies changes, e. g.,

- Replaying incidents or suspicious requests with different policy versions
  - Does a change in the policy lead to a different result?

NHS policies in XACML

Security policy for a NHS (National Health Service) electronic health record service (Becker, 2005):

- Permissions rely on relationships
- Policy how a relationship can be established

Modeling in XACML

- Relationships as attributes, e. g.,
  - Patient has a set of treating clinicians
- Saved as part of the policy in the policy store
- Resolved at runtime by an XACML attribute designator

Policy Administration

- Policy for management of relationships can be seen as administration policy
- Application to manage relationships can be seen as Policy Administration Point (PAP)
- PAP application has to enforce the administration policy, e. g.,
  - Who is permitted to add relationships
  - Implement obligations, e. g., “on delete cascade” for relationships (e. g., referral)

Versioning of policies

Versioning of XACML policies

- Subversion (svn) as versioning system for XML files
- Logging active policy version

Versioning of attributes: save to database

- Validity (i. e., from - to)
- Depending entity (or entities), e. g.,
  - Treating clinicians depend on patient id
- Type and further information of relationship, e. g.,
  - Patient assigning a treating clinician,
    - Patient can revoke relationship
  - Treating clinician referring patient and assigning referred clinical
    - Patient cannot revoke referral relationship
    - Save referring clinical

Break-Glass Scenario

In an emergency situation, there may be no valid patient - clinician relationship, e. g.,

- Patient is unconscious or not able to confirm a relationship
- No agent (i. e., trusted person of the patient) is available at time to confirm a relationship
- Due to an overwhelming emergency situation, a required referral is not entered to the IT system instantly

But, a clinician requires access to the patients health record

- Clinician uses Break-Glass to access the required data
- The access is marked as emergency access and has to be evaluated in a post-hoc audit phase

Post-hoc Analysis I

Analyzing Simple Access Control Overrides

After the emergency situation

- The patient is (hopefully) able to confirm the relationship
- The referral is entered to the system

Using our replay approach, an auditor can easily

- Load a PDP with a policy version from, e. g., twenty four hours later
- Replay the accesses in questions against this policy version
- Information not available at access time can be used to verify the Break-Glass access post-hoc in a semi-automated fashion

Post-hoc Analysis II

Process-based Compliance Checks

Observation:

- Many compliance regulations cannot be directly mapped to access control policies.

Problem:

- After overriding a single access control decision, it is unclear which compliance goals might be violated.

Idea:

- Use (process) mining techniques for re-constructing the actual
    - process executed and
    - data-flow

  that took place.
- Apply formal analysis techniques (e.g., using AVANTSSAR tools)
    - determine the set of high-level compliance and security requirements

  that were violated by the overridden access control decision.

Conclusion

Break-Glass allows to

- Write restrictive policies for the regular case, as, in emergency situations, access is still possible

Our framework allows to

- Use information at post-hoc time which was not available at access time
- Enable (partially) semi-automated evaluation of Break-Glass accesses and therefore reduce effort and costs

Thank you for your attention!

Any questions or remaks?

Related Publications

Achim D. Brucker and Helmut Petritsch.
Extending access control models with break-glass.
In Barbara Carminati and James Joshi, editors, *ACM SACMAT*, pages 197–206. acm Press, 2009.
ISBN 978-1-60558-537-6.
`http://www.brucker.ch/bibliography/abstract/brucker.ea-extending-2009`.

Achim D. Brucker and Helmut Petritsch.
Idea: Efficient evaluation of access control constraints.
In F. Massacci, D. Wallach, and N. Zannone, editors, *International Symposium on Engineering Secure Software and Systems (ESSoS)*, number 5965 in lncs, pages 157–165. Springer, 2010.
ISBN 978-3-642-11746-6.
`http://www.brucker.ch/bibliography/abstract/brucker.ea-efficient-2010`.

Achim D. Brucker and Helmut Petritsch.
A framework for managing and analyzing changes of security policies.
In *IEEE POLICY*, pages 105–112. ieee Computer Society, June 2011.
ISBN 978-0-7695-4330-7/11.
`http://www.brucker.ch/bibliography/abstract/brucker.ea-framework-2011`.