Attribute Transformation for Attribute-Based Access Control

Prosunjit Biswas, Ravi Sandhu and Ram Krishnan
Department of Computer Science
Department of Electrical and Computer Engineering
University of Texas, San Antonio

ABAC’17, March 24, 2017, Scottsdale, AZ, USA
Outline

- Summary
- Motivation
- Attribute Transformation
- Attribute Reduction
- Attribute Expansion
- Conclusion
- Q/A

World-Leading Research with Real-World Impact!
We have presented a concept of attribute transformation and specify two types of transformation---attribute reduction and attribute expansion.
Attribute explosion!

Figure 1: Attributes defined for OpenStack Virtual Machines
Attribute Explosion incurs difficulties in managing authorization policies and attribute-value assignments.
We cannot get rid of attributes we need.

But we can manage

with

Attribute Transformation
Attribute Transformation (assumptions)

Attribute types

Non-policy Attributes

Policy Attributes

Assumptions:

Non-policy Attributes $\cap$ Policy Attributes $= \emptyset$
Non-policy Attributes $\gg$ Policy Attributes

Examples:

Object attributes (Non-policy):
size, created_by, shared, location

Object attributes (Policy):
sensitivity, security-label
Attribute Transformation is the process of transforming one set of attribute-value assignments into another set of assignments.

Types of attribute transformation

Reduction
(Non-policy Attr → Policy Attr)

Expansion
(Policy Attr → Policy Attr)
Attribute Reduction

The process of transforming non-policy attribute-value assignments into policy attributes-value assignments.

Non-policy attributes

- size(f1)=100MB
- created-by(f1) = system-d
- shared(f1) = false
- location(f1) = /log/system-log

Policy attributes

- security-label(f) = sensitive

Effective assignments

Derived assignments

Derived assignments

security-label(f) = sensitive
Motivation from literature:

1. Attribute-Based User-Role Assignment [1]
Attribute Reduction (usefulness)

Useful for

Abstraction  Modular design  Hierarchical policy
Authorization policy with Policy attributes:

\[ \text{Can-read} \equiv \text{security-label}(o) = \text{sensitive} \land \text{role}(u) = \text{manager} \]

Mapping rules with Non-policy Attributes:

\[ \text{VM-mapping} \equiv \text{resource-type}(o) = \text{VM} \land \text{image-type}(o) = \text{corporate} \rightarrow \text{security-label}(o) = \text{sensitive} \]

\[ \text{Firewall-mapping} \equiv \text{resource-type}(o) = \text{firewall} \land \text{protocol}(o) = \text{UDP} \land \text{network}(o) = \text{internal} \rightarrow \text{security-label}(o) = \text{sensitive} \]
Example of mapping rule:

\[
\text{file-length}(f) = 100 \text{ MB} \land \text{created-by}(f) = \text{system-d} \land \text{is-shared}(f) = \text{false} \rightarrow \text{security-label}(f) = \text{sensitive}
\]
Conflicts resulting from multiple mappings

resource-type(o) = VM
encryption(o) = plain
security-label(o) = regular

resource-type(o) = VM
image-type(o) = corporate
security-label(o) = sensitive

mapping1
mapping2
Conflicts resulting from assigned and derived values

resource-type(o) = VM

encryption(o) = plain

security-label(o) = regular

security-label(o) = sensitive

mapping1
The process of transforming policy-attribute-value assignments into a different set of policy-attributes-value assignments.

**Attribute Expansion**

- **Policy attributes**
  - **is-a-veteran(u) = True**
  - **benefits(u) = \{b1, b2\}**
  - **skills(u) = \{skill1, skill2\}**

**Derived assignments**

**Resulting assignments**

- **is-a-veteran(u) = True**
- **benefits(u) = \{b1, b2\}**
- **skills(u) = \{skill1, skill2\}**
Motivation from literature:

1. Hierarchical Group and Attribute-Based Access Control (HGABAC) [3]
Conclusion

What next?

- Other forms of Attribute Transformation
- Chain of Attribute Transformation
- Fitting Attribute Transformation in ABAC models

