Enforcing Purpose of Use via Workflows

Mohammad Jafari
Dept of Computer Science,
University of Calgary
Calgary AB T2N 1N4, Canada
jafarm@ucalgary.ca

Reihaneh Safavi-Naini
Dept of Computer Science,
University of Calgary
Calgary AB T2N 1N4, Canada
rei@ucalgary.ca

Nicholas Paul Sheppard
Dept of Computer Science,
University of Calgary
Calgary AB T2N 1N4, Canada
nsheppar@ucalgary.ca

ABSTRACT

One of the main privacy concerns of users when submitting their data to an organization is that their data will be used only for the specified purposes. Although privacy policies can specify the purpose, enforcing such policies remains a challenge. In this paper we propose an approach to enforcing purpose in access control systems that uses *workflows*. The intuition behind this approach is that purpose of access can be inferred, and hence associated with, the workflow in which the access takes place. We thus propose to encode purposes as properties of workflows used by organizations and show how this can be implemented. The approach is more general than other known approaches to purpose-based enforcement, and can be used to implement them. We argue the advantages of the new approach in terms of accuracy and expressiveness.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Security

Keywords

privacy, purpose, workflow, access control

1. INTRODUCTION

The OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data [15] contain what is probably the most influential articulation of the principle of purpose of use of private data, by stipulating that use of data should comply with the purpose specified at or before collection time. "Purpose" has thus been included in a number of privacy-oriented access control models [3, 8, 14, 21] and in policy specification languages such as P3P [23], EPAL [20] and XACML [17] as a decision factor in the access control policy. The method by which the purpose of an access control request is tested, however, has been a challenging problem that is addressed only by a fewer number of authors. We will review these methods in Section 2.

In this paper, we propose a new approach to specifying and enforcing purpose-based policies that identifies purpose of access based on the *workflow context* where the access is requested. A *workflow* is a set of tasks that must be carried out in some particular order to achieve a specific goal. The approach and its advantages compared to previous approaches is further discussed in Section 3 and a purpose-based access control model is then proposed in Section 3.1. Section 4 describes a prototype implementation of a purpose-based access control system that provides access in the context of workflows and prevents workflows that would violate the policy from being instantiated. Section 5 gives some observations and possibilities for future work.

2. RELATED WORK

A number of authors proposed to trust the requester to declare the purpose for which the access is requested [10, 12]. This approach however does not prevent a user from claiming a false purpose.

Another widely-used approach is to assign purposes to particular users [25], or to roles in a role-based access control system [7, 13, 18, 19, 24]. For example, only users who are members of the *marketing* role, are permitted to request access to data for *marketing* purposes. This approach assumes a correspondence between purposes and users or roles which could be very limiting. Roles, as collections of permissions or sets of users, are designed with criteria such as organizational structure or job functions in mind and so access request in one role may be for different purposes. For example, a physician in an organization may read a patient's file once for the purpose of treatment and another time with the purpose of some research, two obviously different purposes practiced by same user and role. These two purposes are not distinguishable if purpose is associated with the role.

Finally, a number of authors have noted that the purpose of access lies in the context within which the access takes place, such as the function, or the *task* [21, 9, 11]. This paper employs a similar intuition but generalizes the approach by considering workflows that consist of several tasks in a particular order.

3. WORKFLOW APPROACH

The main intuition is that the purpose of access is usually visible at a higher level unit of work where the access takes place. For example, if Alice is currently calculating tax-

WPES'09, November 9, 2009, Chicago, Illinois, USA.

returns, we can say her purpose of reading the incomes is *tax-return calculation*, whereas when she reads them while looking for potential new customers, her aim is *marketing*. Thus, tasks or workflows can be used as an indication of purpose of access.

A task is a unit of work in a system; for example *Reception* is a unit of work in a hospital. A workflow is a larger work unit consisting of several tasks that should be carried out in some particular order. Figure 1 shows an example of a very simple workflow with three tasks. Compared to tasks, workflows can more accurately specify a purpose. For example, using the workflow approach, one can distinguish between a *medical test* that is part of a treatment process in a hospital, or part of a testing process in a health research project, and hence, differentiate between *treatment* and *research* purposes.

An assumed purpose may lead to bindings of future actions. For instance, if Alice withdraws money from an account for the purpose of buying books, this implies that at a later time, she should buy some books and should not spend the money otherwise. These future obligations are in perfect match with the workflow model wherein execution of a task can only lead to specific future tasks. The importance of the relationship between actions in realizing a purpose has also been pointed out in [2] and [5].

The role of the access requester and the type of data being accessed can sometimes be suggestive about the purpose of access, as observed by the role-based models discussed in Section 2. Since this information is included in the definition of the workflow, as noted in Section 3.1.2, the workflow model can encompass role- or category-based purpose-enforcement models.

Finally, realizing a purposes by assigning it to a workflow can provide a clear interpretation of the purpose name and prevent any ambiguities. The workflow definitions could be made available to auditors, or even published to customers.

3.1 Model

We assume organizations provide all their data accesses and processings within the context of workflows that are designed and maintained by a policy officer. The purpose of each access request is then determined by the the workflow in which it occurs.

3.1.1 Purpose Model

Purpose-based access control policies can be expressed in one of the following two different ways; in practice, an organization will use one of the two:

- *data-centric* in which data items are associated with the purposes for which they can (or cannot) be used. The OECD Guidelines [15], XACML Privacy Profile [17], and some purpose-based models (such as [9], and [6]) are examples of this approach; and

- *rule-centric* where access control rules are composed of a tuple of (at least) subject, action, object and purpose. EPAL [20] and purpose-based models such as [11] are examples of this approach.

Many authors, beginning with Bonatti, et al. [4], organize purposes into a hierarchy or lattice [22]. From this point of view, a workflow can be thought of as being a very specific purpose, lying at a leaf of the purposes hierarchy.

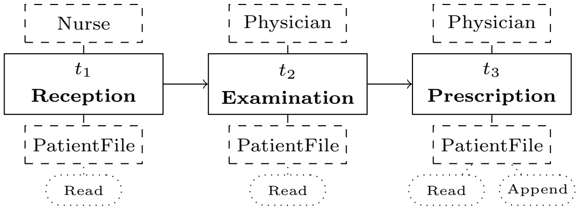

Figure 1: A very simple example of a workflow.

For simplicity of our model, we suppose that the organization supports a flat set of purposes $\mathcal{P} = \{P_1, \ldots, P_n\}$. The implementation described in Section 4, however, could be readily adapted to the use of hierarchical purposes, as discussed in Section 5.

3.1.2 Workflow Model

Let $\mathcal{U}$ denote the set of users, $\mathcal{R}$ the set of roles, $\mathcal{X}$ the set of items of data held by the organization, $\mathcal{C}$ the set of data categories into which items of data are classified, and $\mathcal{A}$ the set of basic actions in the system, such as read or write.

A workflow management system contains a set of *workflow descriptions* $\mathcal{W} = \{W_1, \ldots, W_k\}$ each of which consists of a set of tasks T, and a set of arcs $E \subseteq T \times T$ that denote a precedence relationship between the tasks. Every task is carried out by some *actor* on some array of *resources*. The authorized actors of a task are specified by the mapping $\mathit{TASK\_ROLES} : T \mapsto 2^{\mathcal{R}}$ that maps each task to a set of authorized roles. The *input resources* to the task are defined as a variable-size array of the form $\langle I_1, \ldots I_{n(t)} \rangle$ in which $n(t)$ is the number of inputs to task $t \in T$. Each I_i is a set of categories that denotes the authorized data types for a particular input, i.e. $I_i \in 2^{\mathcal{C}}$. We denote the resource types array of a specific task t by $\mathit{IN\_CATS}(t)$, and the set of categories of its i'th input by $\mathit{IN\_CATS}_i(t)$, where $1 \leq i \leq n(t)$. Each input resource may be subject to some *actions* in the task. A similar variable-size array of the form $\langle A_1, \ldots A_{n(t)} \rangle$ is defined in which each $A_i \in 2^{\mathcal{A}}$ and shows the actions that are performed on the i'th input of the task. The actions array of a specific task t is denoted by $\mathit{IN\_ACTS}(t)$, and the set of actions of its i'th input by $\mathit{IN\_ACTS}_i(t)$. In summary, the workflow description is defined as a quintuple containing T, E, $\mathit{TASK\_ROLES}$, $\mathit{IN\_CATS}$, and $\mathit{IN\_ACTS}$ as defined above. The purpose corresponding to each workflow is denoted by the mapping $\mathit{PURP\_OF} : \mathcal{W} \mapsto \mathcal{P}$.

The workflow management system may create a *workflow instance* by instantiating a workflow description with a set of concrete actors and resources. In order to perform a particular treatment, for example, a workflow instance must be supplied with a particular physician and a particular patient's file. This can be defined by two mappings: one that maps each task to a user, denoted by $\mathit{TASK\_USER}(t)$, and another one mapping each task to an array of input data, denoted by $\mathit{TASK\_INPUT}(t)$. The latter array is of the form $\langle x_1, \ldots, x_{n(t)} \rangle$ where $x_i \in \mathcal{X}$, and its ith element is denoted by $\mathit{TASK\_INPUT}_i(t)$. An instance of the workflow W is thus a triple $\langle W, \mathit{TASK\_USER}, \mathit{TASK\_INPUT} \rangle$.

As an example, a very simple workflow is depicted in Figure 1. In this workflow, $T = \{t_1, t_2, t_3\}$, $E = \{(t_1, t_2), (t_2, t_3)\}$, and the mappings are as follows:

For all i, $\mathit{TASK\_ROLES}(t_i) = \{\text{Physician}\}$, and $\mathit{IN\_CATS}(t_i) = \{\text{PatientFile}\}$. $\mathit{IN\_ACTS}(t_i) = \langle\{\text{Read}\}\rangle$, for i=1,2, and $\mathit{IN\_ACTS}(t_3) = \langle\{\text{Read, Append}\}\rangle$.

3.1.3 Access Control Model

Every user $u \in \mathcal{U}$ is assigned to a set of roles denoted by $ROLES\_OF(u)$ and every item of data $x \in \mathcal{X}$ is assigned a set of data categories, denoted by $CATS\_OF(x)$. The organization's access control policy also includes a data-centric or rule-centric purpose policy described below. The purpose policies may be supplied by the data subject e.g. a patients' consent directive, or by the organization as part of some privacy policy.

In a data-centric policy, every item of data $x \in \mathcal{X}$ submitted to the organization must be associated with a set of intended purposes $INT\_PURP(x) \subseteq \mathcal{P}$.

Rule-centric policies have a more complex form. Let $\mathcal{B} \subseteq \mathcal{R} \times \mathcal{A} \times \mathcal{C} \times \mathcal{P}$ be a policy base consisting of quadruples $\langle R, AC, C, P \rangle$ of role R, action AC, data category C and purpose P for which access is permitted. For simplicity, we do not formalize other features such as subject or object attributes that exist in some policy languages such as EPAL [20] and XACML [16].

An access control request represents a request to instantiate a workflow and thus takes the form of a triple containing W, $TASK\_USER$, and $TASK\_INPUT$ as defined in Section 3.1.2.

The workflow system should comply with three access control requirements as follows:

Role and Category Authorization: Every proposed actor for a task is a member of one of its authorized roles, and every proposed resource for a task falls into one of the authorized categories, that is, for all $t \in T$, the following sets contain at least one member:

$ROLES\_OF(TASK\_USER(t)) \cap TASK\_ROLES(t)$

$CATS\_OF(TASK\_INPUT_i(t)) \cap IN\_CATS_i(t)$ (for each index i in the resource array)

Purpose Authorization: In the case of data-centric policy, the purpose of the workflow W matches with one of the purposes for which each of its resources were collected, that is, for each index i of all of the tasks $t \in T$ of the workflow W:

$PURP\_OF(W) \in INT\_PURP(TASK\_INPUT_i(t))$

In the case of rule-centric policy, every access in the workflow is consistent with the policy-base, that is, for each index i in the resource array of every task t, and for all $AC \in IN\_ACTS_i(t)$, there exists a quadruple

$\langle R, AC, C, PURP\_OF(W) \rangle \in \mathcal{B}$,

in which $R \in ROLES\_OF(TASK\_USER(t))$ and

$C \in CATS\_OF(TASK\_INPUT_i(t))$.

4. IMPLEMENTATION

We have implemented a prototype of a *workflow reference monitor* ("WfRM"), capable of enforcing purpose-based policies at workflow instantiation time, that ensures the access events that may happen in the course of the workflow instance comply with the policy of the system (Figure 2). An alternative design is discussed in Section 5.

We have used XACML version 2 [16] as the format for access control requests, responses and policies and have made use of Enterprise XACML Library [1] as the XACML policy decision point(PDP). The attributes of the entities, such as subject roles, are stored in an *attribute authority* ("AA"), which is currently implemented as a program module. Using a standard attribute authority is left as future work.

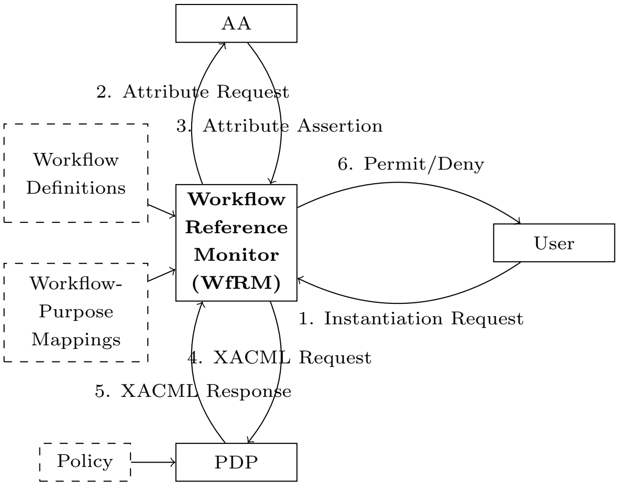

Figure 2: Architecture of the access control system.

For defining workflows, we have designed an XML-based workflow description language in accordance with the model of Section 3.1.2, and similar to those used by well-known workflow engines. The mapping of workflows to purposes is also implemented as a simple XML-based document that assigns purpose names to workflow identifiers. Finally, the instantiation request is also designed in the form of an XML document according to the model of Section 3.1.3.

Upon receiving an instantiation request, WfRM generates a XACML request corresponding to each of the impending access events in the workflow instance. The subject and resource part come from the instantiation request, the actions are part of the workflow definition, and the purpose of access is determined by checking the workflow-purpose mapping, and included as an attribute of action, as recommended by the privacy policy profile of XACML [17]. The AA is also queried to augment the request by adding the attributes of the entities involved in the access event. If the intended purpose is specified for the data item, it appears as an attribute of the resource. Eventually, a set of XACML requests are generated each of which corresponds to one possible access in the workflow instance. The requests are sent to the PDP where they are checked against the access control policy. The workflow instantiation is permitted by the WfRM, only if all access requests are permitted according to the PDP decisions. XACML is capable of supporting both data-centric and rule-based privacy policies (as defined in Section 3.1.1). Figure 3 shows a general data-centric policy stipulating that all purposes of the action should match at least one of the intended purposes the accessed resource.

5. DISCUSSION

Early vs. Late Authorization.

The current implementation considers the execution of a workflow instance as an uninterruptible unit of work: even if a single access event is likely to violate the policy, the workflow instantiation is rejected, although since the actual sequence of tasks is usually dynamic and based on run-time factors, the violating access may in fact never occur. This is important from the purpose-enforcement point of view, since the purpose of workflow lies in its entirety and a halfway-broken instance can no more be assumed to serves the same purpose. A more dynamic approach is to make authorization

```
<Rule RuleId="matching-purpose" Effect="Permit">
 <Condition>
  <Apply FunctionId="function:all-of-any">
   <Function FunctionId="function:string-equal"/>
   <ActionAttributeDesignator AttributeId="action:purpose"/>
   <ResourceAttributeDesignator AttributeId="resource:purpose"/>
  </Apply>
 </Condition>
</Rule>
```

Figure 3: A general data-centric rule.

decisions upon the commencement of each task. In that case, a workflow instance that *may*, but not necessarily *will* have a violating access event is allowed to be instantiated.

Abstract vs. Concrete Purposes.

Suppose Alice purchases a book from an on-line bookstore, and therefore submits her address to the bookstore under the condition that it only be used for the purpose of delivering goods. Alice presumably means that her address is only to be used in the delivery of her book, and not just any goods to anyone. Our model of purpose, along with those used in all previous works of which we are aware, does not really capture this distinction, but Alice's intent does have a natural interpretation in terms of workflows: she means that her address should only used by the particular workflow instance that is created in order to deliver her book, and not just any workflow for delivering goods. Our model could be extended to support abstract and concrete purposes, though the notation becomes somewhat more complicated, and we will leave investigation of this kind of model as a future work.

Purpose Hierarchy.

The current implementation does not support purpose hierarchies. One simple way of implementing this is to use a regular expression string matching functions in the policy, together with an appropriate prefix naming scheme in the purpose tree.

6. CONCLUSION

Workflows provide a convenient and accurate way of associating specific actions with broader purposes. With the assistance of a workflow management system, an access control system can reliably determine the purpose of an action, and enforce a purpose-based access control policy in an automated fashion.

7. REFERENCES

[1] Enterprise Java XACML. `http://code.google.com/p/enterprise-java-xacml/`.

[2] S. S. Al-Fedaghi. Beyond purpose-based privacy access control. In *ADC'07*, pages 23–32, Ballarat, Australia.

[3] C. A. Ardagna, S. De Capitani di Vimercati, and P. Samarati. Enhancing user privacy through data handling policies. In *DBSEC'06*, pages 224–236, Sophia Antipolis, France.

[4] P. A. Bonatti, E. Damiani, S. de Capitani di Vimercati, and P. Samarati. A component-based architecture for secure data publication. In *ACSAC'01*, pages 309–318, New Orleans, USA.

[5] T. D. Breaux and A. I. Antón. Deriving semantic models from privacy policies. In *IEEE POLICY'05*, pages 67–76, Stockholm, Sweden.

[6] J.-W. Byun, E. Bertino, and N. Li. Purpose based access control of complex data for privacy protection. In *SACMAT'05*, pages 102–110.

[7] J.-W. Byun and N. Li. Purpose based access control for privacy protection in relational database systems. *The VLDB Journal*, 17:603–619, 2008.

[8] A. C. Duta and K. Barker. P4A: A new privacy model for XML. In *DBSEC'08*, pages 65–80, London, UK.

[9] S. Fischer-Hübner. *IT-Security and Privacy: Design and Use of Privacy-Enhancing Security Mechanisms.* Springer, Berlin, Germany, 2001.

[10] H. Haygood, Q. He, S. Smith, and J. Snare. A privacy-aware database interface. Technical Report TR-2003-05, North Carolina State University, 2003.

[11] Q. He. Privacy enforcement with an extended role-based access control model. Technical Report TR-2003-09, North Carolina State University, 2003.

[12] M. Jawad, P. S. Alvaredo, and P. Valduriez. Design of PriServ, a privacy service for DHTs. In *PAIS'08*, pages 21–26, Nantes, France.

[13] A. Masoumzadeh and J. B. D. Joshi. PuRBAC: Purpose-aware role-based access control. In *On the Move to Meaningful Internet Systems, Part II*, pages 1104–1121, Monterrey, Mexico, 2008.

[14] Q. Ni, A. Trombetta, E. Bertino, and J. Lobo. Privacy-aware role based access control. In *SACMAT'07*, pages 41–50, Sophia Antipolis, France.

[15] Organisation for Economic Co-operation and Development. OECD guidelines on the protection of privacy and transborder flows of personal data, 1980.

[16] Organisation for the Advancement of Structured Information Standards. OASIS eXtensible access control markup language, 2005.

[17] Organisation for the Advancement of Structured Information Standards. Privacy policy profile of XACML v2.0, 2005.

[18] H. Peng, J. Gu, and X. Ye. Dynamic purpose-based access control. In *ISPA'08*, pages 695–700, Sydney, Australia.

[19] F. Salim, N. P. Sheppard, and R. Safavi-Naini. Enforcing P3P policies using a digital rights management system. In *PETS'07*, pages 200–217, Ottawa, Canada.

[20] M. Schunter and C. Powers. The Enterprise Privacy Authorization Language (EPAL 1.1), 2003.

[21] T. Tachikawa, H. Higaki, and M. Takizawa. Purpose-oriented access control model in object-based systems. In *ACISP'97*, pages 38–49, Sydney, Australia.

[22] W. van Staden and M. S. Olivier. Using purpose lattices to facilitate customisation of privacy agreements. In *TrustBus'07*, pages 201–209, Regensburg, Germany.

[23] W3 Consortium. Platform for Privacy Preferences (P3P) project. `http://www.w3.org/P3P`, 2004.

[24] N. Yang, H. Barringer, and N. Zhang. A purpose-based access control model. In *IAS'07*, pages 143–148, Manchester, UK.

[25] G. Zhan, Z. Li, X. Ye, and J. Wang. Privacy preservation and protection by extending generalized partial indices. In *BNCOD'06*, pages 102–114, Belfast, UK.