Access Control Enforcement for Conversation-based Web Services

Massimo Mecella *
Univ. Roma LA SAPIENZA, Italy

Mourad Ouzzani
Purdue University, USA

Federica Paci
Univ. Milano, Italy

Elisa Bertino
Purdue University, USA

* while a visiting researcher (fall 2005) in the Department of Computer Science and CERIAS, Purdue University, USA
Overview

• The conversational model of Web services
• Security concerns
• Access control based on conversations
  – K-trustworthiness
• The technique
• The architecture
• Conclusions
Web Services

- A Web service is characterized by the set of (atomic) operations that it exports ...

- ... and possibly by constraints on the possible conversations
  - Using a service typically involves performing sequences of operations in a particular order (conversations)
  - During a conversation, the client typically chooses the next operation to invoke on the basis of previous results, among the ones that the service allows at that point
Web Services

- A service is characterized by the set of (atomic) operations that it exports ...

- ... and possibly by constraints on the possible conversations
  - Using a service typically involves performing sequences of operations in a particular order (conversations)
  - During a conversation, the client typically chooses the next operation to invoke on the basis of previous results, among the ones that the service allows at that point
Transition Systems

- A transition system (TS) is a tuple $T = \langle A, S, S^0, \delta, F \rangle$ where:
  - $A$ is the set of actions
  - $S$ is the set of states
  - $S^0 \subseteq S$ is the set of initial states
  - $\delta : S \times A \times S$ is the transition relation
  - $F \subseteq S$ is the set of final states

- Initial state: the client starts the interaction
- Final state(s): the client can terminate the interaction (it has reached its own goal and the service is not "dangling")
**The Conversational Model**

**Abstract Behavior of the Service:**

Do until Client selects “end”

1. Give Client a choice of actions to be performed
2. Wait for Client choice
3. Perform action chosen by Client

Conversations supported by the service as a TS
Security Concerns

- **Access Control**
  - **Credentials**
    - signed assertions describing properties of a subject that are used to establish trust between two unknown communicating parties before allowing access to information or services
  - **Access control policies**
    - rules stating that only subjects with certain credentials satisfying specific conditions can invoke a given operation of the Web service
Current Approaches (1)

- Single operation model
  - operations are not related to ("independent" from) each other

- Access control is enforced
  - at the level of the entire Web service
    - the Web service could ask the client, in advance, to provide all the credentials associated with all operations of that Web Service
    - A subject will always arrive at the end of whichever conversation
    - The subject will become aware of all policies on the basis of which access control is enforced
    - The client may have to submit more credentials than needed
Current Approaches (2)

- at the level of single operations
  - to require only the credentials associated with the next operation that the client wants to perform
    - Asking from the subject only the credentials necessary to gain access to the requested operation
    - The subject is continuously solicited to provide credentials for each transition
    - After several steps, the client may reach a state in which it cannot progress because the lack of credentials (and the service provider has wasted resources)
Challenges

- Access control not only at the level of single operation
- Should consider conversations
  • Willingness of the client to reach a “goal”
  • Willingness of the service provider not to waste resources
  • Willingness of the service provider to limit disclosure of access control policies
The Idea

- Considering access control mainly at the level of conversations (sequences of operations leading to a final state of the TS)
- The service provider gives a k-trustworthiness level $k$ to a client in a given state
- On the basis of such a $k$, asks the client to provide credentials for the conversations of length less/equal $k$ (starting from the current state and with operations not yet “controlled”)
The Rationale (1)

• The approach maximizes the likelihood that a client reaches a final state and doesn’t drop off due to lack of authorization
  - Likelihood and not guarantee as the client is free, and can take different conversations

• The approach maximizes also the likelihood that the service provider doesn’t waste resources, even without disclosing the access policies
Example

Conversations from $S_0$:
- $\text{chooseItem} \rightarrow \text{addToCart} \rightarrow \text{saveForLater} \rightarrow \text{chooseItem} \rightarrow \text{addToCart} \rightarrow \text{checkOut} \rightarrow \text{completeTransaction}$

Hence the $k$-levels for $S_0$ are $\{3,4\}$

$k$-levels for $S_2$ are $\{1,2\}$
**Interaction Model**

---

**Client**

1. **Bind**
2. **Invoke Operation**: \( op \)
3. **Execute Operation**: \( \text{Is an Authorized Operation (} op \in \text{conversations of} \ k \text{)}? \)
   - **Yes**: **Assign New K-Level**
   - **No**: **Access Denied**

**Web Service**

1. **requireCredentials()**
2. **submitCredentials()**
3. **Calculate Required Credentials**
4. **Evaluate Credentials Against Policies**
   - **Policies Not Satisfied**
   - **Policies Satisfied**: **Access Denied**

---

*On the basis of previously provided credentials, it may be.*
Basic Concepts (1)

- **Credential**
  - Attribute (pair <name, value>)
- **Attribute condition**
- **A credential satisfies an attribute condition if one among its attributes makes true the condition**
- **Operation access control policy**
  - Rule specifying credentials and attribute conditions to grant access to the operation
  - Can be checked by a reasoning service that verifies if the access request is a logical consequence of the policy and the credentials
Basic Concepts (2)

- **Conversation access control policy**
  - Conjunction of the access control policies of the operations in the conversation

- **Trustworthiness level**
  - Length of “allowed” conversations

- **k-trust policies**
  - Given a state with different possible k-levels, defines which one to assign
The Technique (1)

- Given a TS, compute, for each state, all the possible k-levels
  - Requires computing all possible conversations
  - Are infinite for cyclic TSs!!
  - But for access control, once an operation has been checked, we do not have to check again

- We need to resort to the concept of
  - strongly connected component (SCC) of a TS
  - Graph of SCCs ($G^{SCC}$): acyclic, and can be computed by the Tarjan’s algorithm
The Technique (2)

• For any SCC, we need to determine all possible conversations that will lead from an in-going node, i.e., coming from outside the component, to an out-going node, i.e., going outside the component.

• These conversations should have the properties to cover all potential operations within the given strongly connected component.
  - Given a node in $G^{SCC}$, formal concepts of cardinality, coverage and rank.
The overall idea of the algorithm, which finds all potential k-trustworthiness levels for all states, is:

- for a given state, determine all subsequent SCCs, including the one to which the current state belongs to
- Traverse the transition system from that state and record all conversations leading to a final state
The Technique (4)
[An Example]

4 is the cardinality of $C_1$, as there are 4 different symbols: \{c,g,h,e\}
7 is the coverage, as you need a sequence of length 7 (c f e f h f c f g f c f e) to include all the four symbols going from the root to the end of the SCC

$C_1$ is the image (SCC) of the set of states \{S_1,S_3,S_5\}
Architecture

EXECUTION CONTROLLER SYSTEM

WEB SERVICE INFRASTRUCTURE

1. Access Request (Operation /Credentials)
   PEP - Policy Enforcement Point
   2. Request State + Requested Op
   3. Status + Table
   4. Credentials + K-Trust Levels + Conversations
   5. Request
   6. K-Trust Policies
   7. K-Trust Level + Conversations
   8. Request
   9. Access Policies
   10. Policies + K-Trust Level
   11. Request for Credentials
   12. Credentials
   13. Access Granted/Denied

K-TRUST POLICIES

ACCESS POLICIES

Policy Selection Module

PDP - Policy Decision Point

K-Trustworthiness Level Assignment Module

Transition System (TS) Table of K-Trustworthiness Levels + Conversations
Conclusions & Future Works

• A novel technique for access control enforcement taking into account the conversational nature of Web service
  - tradeoff between step-by-step (minimize the disclosure by maximizing the risk) and request-all (minimize the risk by maximizing the disclosure)
  - Good if k-level assignment is fine tuned (through client profiling)

• Conclude the ongoing implementation of the access control enforcement platform
  - Performance and scalability tests
• Apply the idea of k-trustworthiness to Web service choreographies
  - Compositions (i.e., orchestrators a-la Roman way) are already seamlessly included in the model
Backup
The Rationale (2)  
[A Simple Probability Model]

• Given an operation \( a \), we consider \( P_a \) as the probability that the client DOES NOT have the credential(s) satisfying the access control policy guarding the operation.

• **Damage** of having a client dropping off is the number of executed operations.

• **Leakage** in terms of disclosure of access control policies is proportional to the number of executed operations.

• Let’s consider a conversation \( \text{conv} = \{ a_1, \ldots, a_n \} \)
The Rationale (3)  
[A Simple Probability Model]

• Step-by-step
  - Risk faced before involving the i-th operation ($a_i$ is the next operation the client may not possess credentials)
    \[ R_i = P_{a_i} f(i - 1) \quad i = 1...n \]
  - Leakage after the i-th operation ($a_{i+1}$ is the next operation)
    \[ L_i = P_{a_{i+1}} f_i \quad i = 1...n \]

• Conversation-based
  - Risk faced after conv (being conv the conversation the service provider has requested the credentials)
    \[ R_i = \prod_{i=1}^{n} P_{a_i} f0 = 0 \quad i = 1...n \]
  - Leakage after the i-th operation ($a_{i+1}$ is the next operation)
    \[ L_i = P_{a_{i+1}} fn \quad i = 1...n \]

<table>
<thead>
<tr>
<th>Metric</th>
<th>Step-by-step</th>
<th>Conversation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk: $\sum_{i=1}^{n} R_i$</td>
<td>$P \frac{n(n-1)}{2}$</td>
<td>0</td>
</tr>
<tr>
<td>Leakage: $L_n$</td>
<td>$n$</td>
<td>$n$</td>
</tr>
</tbody>
</table>
The Rationale (4)
[A Simple Probability Model]

Conversation based is a tradeoff between step-by-step (minimize the disclosure by maximizing the risk) and request-all (minimize the risk by maximizing the disclosure)
Good if k-level assignment is fine tuned (through client profiling)

<table>
<thead>
<tr>
<th>Metric</th>
<th>step-by-step</th>
<th>k-level: 2</th>
<th>k-level: 4</th>
<th>request-all</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk</td>
<td>$2 \cdot \mathcal{P}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Leakage</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>acde</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk</td>
<td>$6 \cdot \mathcal{P}$</td>
<td>$0 + 3 \cdot \mathcal{P}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Leakage</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>