Guarding Sensitive Information Streams through the Jungle of Composite Web Services

Jinpeng Wei, Lenin Singaravelu, Calton Pu
Georgia Institute of Technology
Atlanta, Georgia, USA
Agenda

- Problem statement
- WS-sensFlow - Security policy specification
- Concrete solution: SF-Guard
  - Security policy enforcement
  - Prototype implementation and its evaluation
- Related work and conclusion
Running Example: Travel Agent

Travel Agent
- reserve (name, freqflyer, creditcard)

Car Rental
- rent (name, creditcard)

Hotel
- book (name, creditcard)

Airline
- book (name, freqflyer, creditcard)

Third Party Airline
- charge (name, creditcard)
- rebook (name, freqflyer, creditcard)

Credit Card Co.
- charge (name, creditcard)
Running Example: Travel Agent

Travel Agent

- reserve (name, freqflyer, creditcard)
- book (name, freqflyer, creditcard)

Car Rental

- rent (name, creditcard)
- charge (name, creditcard)

Hotel

- book (name, creditcard)
- charge (name, creditcard)

Airline

- book (name, freqflyer, creditcard)
- charge (name, creditcard)
- rebook (name, freqflyer, creditcard)

Credit Card Co.

- charge (name, creditcard)

Third Party Airline

- freqflyer ?

Web Service invocation
Non Web Service invocation
Agenda

- Problem statement
- **WS-sensFlow - Security policy specification**
- Concrete solution: SF-Guard
  - Security policy enforcement
  - Prototype implementation and its evaluation
- Related work and conclusion
WS-sensFlow

- Policy-based: specification and attachment of security policies to the web service invocation requests
- Fine-grain
  - Spatially, different data items can have different security policies
  - Temporally, the security policy for the same data item can change from one invocation to another
Security Policy Envelopes

- **Formal Definition**

  \[ L = \langle \text{white list} \rangle; \langle \text{black list} \rangle \]
  \[ \langle \text{white list} \rangle = \text{allow} \langle \text{node list} \rangle \]
  \[ \langle \text{black list} \rangle = \text{deny} \langle \text{node list} \rangle \]
  \[ \langle \text{node list} \rangle = * | \langle \text{node id} \rangle | \langle \text{node id}, \langle \text{node list} \rangle \rangle \]

- **Example**

  ```
  reserve (  
    name <allow *>,  
    freqflyer <allow Travel Agent, Airline, Hotel, Car Rental;  
      deny Third Party Airline , Credit Card Co>,  
    creditcard <…>)
  ```
Secure Policy Specification (1)

- Composite Service Topology Discovery
  - Leverage on meta-information exchanged dynamically among component web services
  - Leverage on ontology to infer information streams

URI: A
- Provides: \{f(n)\}
- Calls: \{B: g(m,n)\}

URI: B
- Provides: \{g(m,n)\}
- Calls: \{C: h(n,p)\}

URI: C
- Provides: \{h(n,p)\}
- Calls: {}
Secure Policy Specification (2)

• Generation of SPEs
  - Known nodes: based on the trust on them
  - Unfamiliar nodes: leverage on reputation and trust systems

```xml
reserve (
  name <allow *>,
  freqflyer <allow Travel Agent, Airline, Hotel, Car Rental;
    deny Third Party Airline , Credit Card Co>,
  creditcard <…>)
```
Agenda

- Problem statement
- WS-sensFlow - Security policy specification
- Concrete solution: SF-Guard
  - Security policy enforcement
  - Prototype implementation and its evaluation
- Related work and conclusion
Secure Policy Enforcement and Propagation: SF-Guard (1)

- Threat model: There is a minimal TCB (Trusted Computing Base) on each web service node, but the web service application (business logic) is not trusted.
- SF-Guard is added as part of the TCB on each web service node to enforce the SPEs.
- SF-Guard checks the security policy envelopes before invoking a target web service.

freqflyer <allow …; deny Third Party Airline, …>
Secure Policy Enforcement and Propagation: SF-Guard (2)

- Using *capabilities* to hide sensitive information from the business logic.
- Operate on the sensitive information on behalf of the business logic

**Feasibility**
- Security-sensitive information is read only. E.g., Social security number
- Security-sensitive information is atomic. E.g., Credit card number

**Conclusion:** Only a few pre-defined simple interfaces are required.
Incoming Message Sanitization
Normal Operation on the Sensitive Information

Business Logic (Untrusted)

... capability ...

Interface Call

SG-Wrapper

Secure Object Repository

secret

Mapping Table
Outgoing Message Processing

Business Logic (Untrusted)

Mapping Table

SG-Wrapper

Secure Object Repository

secret

Output SOAP message

Output SOAP message

Web Service Framework (Trusted)
Agenda

- Problem statement
- WS-sensFlow - Security policy specification
- Concrete solution: SF-Guard
  - Security policy enforcement
  - Prototype implementation and its evaluation
- Related work and conclusion
SF-Guard Prototype Implementation

- Based on Apache Axis2
- As a module inserted into the message processing stack, between web service framework and the business logic
- Works by checking and manipulating attributes of the XML elements in a SOAP message. e.g., whitelist, blacklist, capability
- A wrapper object for the sensitive information is passed on to the business logic through the Axis2 message context
Evaluation

- Protection of SF-Guard
  - Reducing the size of WSF that has to be trusted
- Reasonable overhead

<table>
<thead>
<tr>
<th></th>
<th>Original (ms)</th>
<th>SF-guard (ms)</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>793</td>
<td>413</td>
<td>305</td>
</tr>
<tr>
<td></td>
<td>819</td>
<td>422</td>
<td>310</td>
</tr>
<tr>
<td></td>
<td>3.3%</td>
<td>2.2%</td>
<td>1.6%</td>
</tr>
</tbody>
</table>
Agenda

- Problem statement
- WS-sensFlow - Security policy specification
- Concrete solution: SF-Guard
  - Security policy enforcement
  - Prototype implementation and its evaluation
- Related work and conclusion
Related Work

- Information privacy in web applications
  - P3P (Platform for Privacy Preferences)
- Access control in composite web services [Elisa Bertino, ICWS’06]
- Compliance checking of privacy policies [Xu, ICWS’06]
Conclusion

- WS-senFlow specification to support fine-grain, policy-based access control of security-sensitive data in composite web services
- The SF-Guard architecture to enforce the security policy specifications
- Using a wrapper style design with capability-based protection
- Prototype implementation shows strong protection properties and low overhead
Questions?
Thank you!