End to End Security in Service Oriented Architecture

Mehdi Azarmi, Bharat Bhargava
Department of Computer Science and CERIAS, Purdue University

Problem Statement

- Most of the modern services outsource a subset of their functionalities to external services (subcontracting).
- Outsourcing leads to chain of service invocations (service consumers may not be aware of it).
- These services are controlled under different administrations.
- Service consumers have no visibility or control on the execution of these services (no accountability).
- Service consumers may have different policies (sec requirements) that they expect to be monitored or enforced.
- Modern services are continuously changing to meet the business requirements.
- Static security solutions do not work.

We are proposing a dynamic security framework for SOA, which is able to monitor/enforce the user-defined policies and maintain the trustworthiness of services based on their execution history.

Proposed Solution

The following solutions are proposed:

- **End to End security policy monitoring and enforcement**
  - Inter-service information flow control
  - Intercepting external service invocations
  - Using Aspect-Oriented Programming
  - Intra-service information flow control
  - Using taint analysis to detect the propagation of sensitive data to external services
  - Implemented in AOP
  - Service clients are able to define policies in XACML language
  - The proposed system monitors the policies and report back to Trust Broker
  - The proposed system enforces the policies and applies the defined action
  - Actions upon violation of policies: Termination of service execution; delay (throttling); replacing the service with a new service (redirection)

- **Secure and adaptive service composition**
  The objective is to select a subset of services from different service categories in order to maximize the overall security of the SOA application.
  - Leveraging the collected/maintained metrics at the Trust Broker subsystem
  - Formulating the problem as a variation of the Knapsack problem
  - Solving the problem using Dynamic Programming (Pseudo-polynomial solution)

Proposed System Architecture

- Implemented in both SOAP and REST web service technologies (Java)
- Implemented a set of representative scenarios
  - Ticket Reservation Scenario
  - Complex Service chains
- Evaluated the effectiveness of the proposed solutions for these scenarios
- Created a GUI to design new scenarios and interact will different subsystems of the framework

Implementation

Acknowledgement

This project is supported by Northrop Grumman Consortium (NGC program).