You live in a certified house, you drive a certified car, why would you use an uncertified service?

Ernesto Damiani
SESAR Lab, Università degli Studi di Milano
Advanced Security Service cERTificate for SOA (ASSERT4SOA) Factsheet

- Started October 1st 2010
- [http://www.assert4soa.eu](http://www.assert4soa.eu)
- Partners:
  - SAP AG, Germany
  - Università degli Studi di Milano, Italy
  - Universidad De Malaga, Spain
  - Fraunhofer Gesellschaft zur Förderung der Angewandten Forschung e.V., Germany
  - Engineering Ingegneria Informatica, Italy
  - City University of London, UK
  - Fondazione Ugo Bordoni, Italy
Why ASSERTs? Trust in SOA

- Service-Oriented Computing (SOC) is being used internally by companies. Slowly, the benefits of this paradigm are fostering their expansion to open systems.
- In an open SOC model, **trust** is an essential element.
  - However, we have no tools to support trust in open settings.
- **Certification** is a well-known approach for establishing trust on security properties of products, systems, services...
  - Software or people can rely on the asserted properties, provided the process of certification produces sufficient evidence.
  - Today’s certification processes include examination by experts following pre-defined and publicly accepted criteria.
  - But, the evidence itself is not typically part of the awarded certificate and **no machine-processable element** is produced.
Why ASSERTs? Trust in SOA

- Thus, currently the relying party needs
  - To trust the **certificate**
  - The **experts**
  - The **certification scheme** and
  - Needs to know in advance what are all the details and meaning of each certification scheme used

- Trust is established by
  - The scheme being run by accredited authorities
  - The accreditation of the experts themselves and
  - The certificate being officially approved
Why ASSERTs? Certification limitations

- Certification–based assurance of security does not scale well to service–based systems in clouds because of:
  - Dynamic service discovery and binding
  - High degree of distribution
  - Lack of control over services and
  - Ever–changing environments

- Current schemes produce certificates intended for human users (natural language, high abstraction level)
- The asserted properties are not explicitly mentioned in certificates. They are either:
  - Part of the scheme (like in BSI Grundschutz) or
  - Expressed in a separate document (like the Security Target in a CC evaluation)

- Current certificates refer to a particular version of the product or system. In general, changes in the system structure require re–certification
Why ASSERTs? Certification limitations

- The system that is subject of certification is considered to be monolithic
  - In CC, the system boundaries are explicitly defined, and security assumptions on the environment can be expressed
  - CC v3.1 allows to deal with composite systems (i.e., derive a system certification from certificates of its components), but requires a perfect match between assumptions and component guarantees
- Existing certification schemes do not provide support for dynamic changes of components (i.e., at run–time)
  - Even in CC v3.1, changing components would require new evaluator/expert interaction and repetition of (at least parts of) the evaluation and certification.
ASSERT4SOA Results

- **Techniques and tools** for expressing and certifying security properties of complex service-oriented applications
  - Enable a **multi-party trust model** suitable for open service ecosystems

- Integrating security certification in the SOA service lifecycle, e.g. on clouds
  - Enable new business models for software certification

- Extend SOA infrastructure to achieve **certificate-based selection and comparison** of services
  - Increase service users’ confidence and enable assurance-driven service composition
ASSERT4SOA Certification

Service Consumer

Certificate Administration Point

Requested Assurance Properties

Service Discovery (ASSERT Aware)

Requested Assurance Properties

CERTificate Decision Point

ASSERT Accredited Authority

Test Suites

Models

Ontologies

Model Reasoner

Evidence Reasoner

Ontology Reasoner

Certificate Evaluation Point
Evidence/Test-based certificates provide evidence-based proofs that a test carried out on the software has given a certain result, which in turn shows (perhaps with a certain level of uncertainty) that a given property holds for that software.

Model-based certificates provide formal proofs that an abstract model (e.g., a set of logic formulas, or a formal computational model such as a finite state automaton) representing a software system holds a given property.

Ontology-based certificates provide a solution to issue an ASSERT4SOA certificate starting from existing certificates (e.g., Common Criteria).
A closer look: ASSERT-E Certificate

- XML incarnation that Includes:
  - Concrete security property
  - Test model
  - Test evidences
  - Aggregate metrics

- Allows:
  - Automatic Discovery and Matching against security-oriented queries
  - Comparison between certificates
  - Security aware run-time composition
Hierarchy of Security Properties
The matching process is called “Triple Matching”, since it deals with security properties (property-match), service models (model-match), and evidence (evidence-match) in the certificate.

Every matching process that constitutes the triple matching is managed by a sub-matching engine.

Triple matching is successful if both property-match, model-match, and evidence-match succeed.

Each sub-matching process is triggered by the client’s preferences.

- For instance, a client can express requirements on security properties and evidence only.
- In this case, the matching process involved in the final decision (match, no-match) considers the property- and evidence-match only.
Comparison (1)

- The comparison process receives in input the set of services’ certificates that satisfy the client’s requirements (matching process) and returns in output a ranked list of these certificates
  - Different approach can be used like cumulative metrics or rule-based metrics
- The properties in the certificates of two different services can be compared using the security property hierarchy
- The models can be compared by evaluating their level of detail (i.e., WSDL, WSCL, implementation)
  - The set of indexes (e.g., number of nodes, edges, linear independent paths) can be used to compare services with the same level of detail
- The evidence can be compared using the hierarchies of test types and the specified test attributes
  - The set of coverage metrics can be used to evaluate the certification quality and used to rank and compare services
Different approaches can produce different ranked lists of services that match the client’s requirements

Clients’ profiles can be defined to make the ranked list closer to their expectations and will drive the service selection

As an example, if a user expresses a profile with trust on property definition only, a rule-based aggregation considering the property first should be used
Composition of Services

- The goal is to evaluate security property for a composition of services starting from the certificates of each component.
- We identified two approaches:
- A-priori certification of composed services
  - Issuing a Virtual certificate starting from each component’s certificate
  - It is called Virtual since the evidence is not generated by testing the entire composition but from combining the test-based evidence of each service component
  - A virtual certificate can be substituted by a real ASSERT-E, if the real test cases are executed on the service
- A-posteriori, the composition is calculated iff there is no single service available at matching time
  - The virtual certificate can be of type ASSERT-O only
  - Having virtual evidence computed runtime can be difficult to achieve
Three readings

Thank you for the attention