Challenges of Security Risks in Service-Oriented Architectures

Youakim Badr\textsuperscript{1}, Frederique Biennier\textsuperscript{1}, Pascal Bou Nassar\textsuperscript{3}, Soumya Banerjee\textsuperscript{2}

\textsuperscript{1} LIRIS Lab, INSA-Lyon, France
\textsuperscript{2} Agence Universitaire de la Francophonie (AUF)
\textsuperscript{3} Birla Institute of Technology, Mesra, India
Outline

- Motivation Example

- Challenges:
  - Managing security in opened, dynamic, and distributed environments
  - Handling unforeseen threats and deciding on security treatment strategies

- Contributions:
  - Security aware SOA design method
  - Dependency Model and Security Service Reference Model
  - **Design time**: Security Support-decision system
  - **Runtime**: Security Monitor system

- Conclusion and perspectives
Motivating Example:
SOA and information security in opened and dynamic environments

Information security: Confidentiality, Integrity, Availability, Accountability, Assurance, Non-repudiation, ...
Web Service Security

Web service Security Standards
- Application layer: SAML, ebXML, XACML, XML Firewall, …
- Messaging layer: SOAP, WS-Security, XML-Signature, XML Encryption, …
- Transport layer: TLS/SSL, HTTP, FTP, SMTP, TCP/IP, …

XML specific attacks
- oversize payload, coercive parsing, XML injection, WSDL scanning, indirect flooding, SOAPAction spoofing, BPEL state deviation, middleware hijacking, …

Security aware SOA infrastructures?
Challenges

- **Existing SOA design methods**
  - provide little attention to integrate security concerns in reference models, guiding each stage of the *service lifecycle* (i.e., design and runtime)
    - **Reference Models**: (OASIS) reference architecture, (Open Group) SOA Ontology, …
    - **SOA Design Methods**: SOMA, SOAD, CBM, SOAF, SODM, …

- **SOA security solutions**
  - often limited to services, composition mechanisms and technical implementation
  - underestimate the *(opened & dynamic) environment* by which SOA-based applications collaborate and exchange information (=>end-to-end security)

- **Need for security risk management**
  - **Security Management**: define global and coherent security policies
  - **Risk Management**: OCTAVE, EBIOS, CORAS, SNA,…
The Security Risk-driven SOA Design Method addresses information security in the SOA from a risk management perspective (...) at design time and runtime.

Cycle de vie
- The Preparatory Stage
- The Design Stage
- The Execution Stage

Outcome:
- key models, tools and deliverables in each step to progressively identify business goals, essential assets, and services
**Dependency Model**

- **Essential Assets for the SOA design context**
  - **Business Assets**
    - business processes, documents, partners, actors, roles, …
  - **Service Assets**
    - atomic & composite services, operations, messages, …
  - **Infrastructure Assets**
    - hardware, software, network protocols, …

- **Building the Dependency Graph**
  - Bayesian Networks learned from surveys
1- The Service Identification and Specification Phase

2- The Risk Management Phase

3- The Annotation Phase
The Service Identification and Specification Phase

- **1: Business Domain Identification**
  - 2A: Business Process Modeling
  - 2B: Business Document Modeling
- **3: Security Objectives Identification**
- **4: Service Identification**
- **5: Service Specification**

<table>
<thead>
<tr>
<th>Steps</th>
<th>Tasks</th>
<th>Deliverables</th>
</tr>
</thead>
</table>
| 1     | Identify domain business assets:  
- what (missions), how (activities)  
- who (actors), why (motivations) |  
- Business goals (OMG Business motivation Model)  
- Business objects and activities  
- Actors-system interactions |
| 2A/2B | Establish use cases and business processes |  
- UML use cases  
- BPMN Business process |
| 3     | Identify business needs and security goals |  
- Business security goals (EBIOS, OCTAVE)  
(Confidentiality, Integrity, Availability, ...), |
| 4     | Apply an outside-in approach to identify services based on business objects and processes and use cases |  
**Top-down approach**: manual activities, automated activities (atomic services, composite services, ...)  
**Bottom-up approach**: legacy and technical services, |
| 5     | Specify service profiles:  
- Business capabilities,  
- Functional / non-functional properties |  
- Service specifications |
# The Risk Management Phase

- **6: Context Establishment**
  - Identify essential assets at business, service and infrastructure
  - Essential Assets
  - Asset contexts: Dependency Graph

- **7A: Security Requirements**
  - Identify security requirements for each asset based on business security goals
  - Identify risks related to assets
  - Vulnerability list (CERT/MITRE)
  - Threats list (EBIOS/OCTAVE)
  - Security Policy Model

- **7B: Risk Identification**
  - Evaluate risks
    - Severity of impact
    - Rate of occurrences
  - Risk list
  - Risk Model

- **8: Risk Assessment**
  - Prioritize risks
    - Evaluate security costs
    - Choose a risk treatment strategy
  - Security Policy Model
  - Treatment strategies:
    - Avoidance, reduction, sharing, retention

- **9: Risk Treatment**
  - Annotate asset security levels with weighted values
  - Secure Design ontology
  - Security annotations (confidentiality, availability, ...)

- **10**
Example: Risk Levels

<table>
<thead>
<tr>
<th>Rate of Occurrences</th>
<th>Insignificant [0 minute, 30 seconds]</th>
<th>Minor [30 seconds, 5 minutes]</th>
<th>Major [5 minutes, 2 hours]</th>
<th>Catastrophic [2 hours, $\infty$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rare [2, 9] :10 hours</td>
<td>Low Risk</td>
<td>Low Risk</td>
<td>Low Risk</td>
<td>Medium Risk</td>
</tr>
<tr>
<td>Possible [10, 19] : 10 hours</td>
<td>Low Risk</td>
<td>Medium Risk</td>
<td>Medium Risk</td>
<td>Medium Risk</td>
</tr>
<tr>
<td>Probable [20, 49] : 10 hours</td>
<td>Low Risk</td>
<td>Medium Risk</td>
<td>High Risk</td>
<td>High Risk</td>
</tr>
<tr>
<td>Certain [50, $\infty$] : 10 hours</td>
<td>Medium Risk</td>
<td>Medium Risk</td>
<td>High Risk</td>
<td>High Risk</td>
</tr>
</tbody>
</table>
Example: Availability Threat Scenario

Web Portal Availability

**Threat Scenario (1): Web Container Crash**
- Incident: Hard Disk Crash
- Rate of Occurrence: Rare: [0, 1]: 5 years
- Scenario Probability: 0.9
- Combine Value: [0, 1]: 5 x 0.9 = [0, 0.9]:5
- Global Occurrence Probability: [0, 0.9]:5 + [1.8, 4.5]:5 = [1.8, 5.4]:5
  
  **Threat Scenario (2): Router Crash**
- Incident: Ethernet Card Failure
- Rate of Occurrence: Possible: [2, 5]: 5 years
- Scenario Probability: 0.9
- Combine Value: [2, 5]: 5 x 0.9 = [1.8, 4.5]:5
- Global Occurrence Probability: [1.8, 5.4]:5 = Rare
A Continuous Security Improvement Process

1) From risk management phase to service specification phase
   - Risk high => choose a risk treatment strategy

2) From runtime to risk management phase
   - Context changes => establish the context

- Security Decision-Making System
- Service Monitoring System
Problem: Deciding on the best risk treatment strategy to deal with threats often relies on *rules of thumb* and often incorporates security analyst’s *intuition* and judgment.

Risk Treatment Decision Process:
[Threats] **cause** [Risks] **handled by** [Security Objectives] **resulting in** [Security Treatment]

Fuzzy Logic:
- Simulating analogy and approximation
- Handling imprecision measures conveyed by the natural language
The Decision-making System for Security Risk Treatments
Fuzzy Variables and Memberships

1- Fuzzy Linguistic Variables

T(Essential Assets) = \{Service, Operation, Message, Business Process\}
T(Vulnerability) = \{Low, Medium, High\}
T(Incident) = \{Random, Regular, Intentional\}
T(Threat) = \{Malicious, Accidental, Failure, Natural\}
T(Security Objective) = \{Confidentiality, Integrity, Availability, Accountability, Assurance\}
T(Security Measure) = \{Encryption, Authentication, Secure Transmission\}
T(Rate of Occurrence) = \{Certain, Possible, Probable, Rare\}
T(Severity of Impact) = \{Insignificant, Major Impact, Loss\}
T(Risk) = \{Low, Medium, High\}
T(Risk Treatment) = \{Reduction, Sharing, Avoidance, Retention\}

2- Membership Functions

\[
T(u) = \begin{cases} 
0 & u \leq a \\
(u-a)/(b-a) & a < u \leq b \\
1 & b < u \leq c \\
(d-u)/(d-c) & c < u \leq d \\
0 & d < u 
\end{cases}
\]

\[0 \leq a \leq b \leq c \leq d \leq 1\]
3- Fuzzy rules

R₄  IF [Security Measure] THEN [Risk Treatment]

Examples of rules in stage R₁, R₂, R₃ and R₄:

R₁₁  IF Essential Assets is Service AND Vulnerability is High AND Incident is Intentional THEN Threat is Malicious
R₂₁  IF Threat is Malicious AND Rate of Occurrence is Possible AND Severity of Impact is Loss THEN Risk is High
R₃₁  IF Risk is AND Security Objective is Confidentiality THEN Security Measure is Encryption
R₄₁  IF Security Measure is Encryption THEN Risk Treatment is Reduction
4 - Fuzzy evaluation method to propagate multi-stage analysis
Problem: Revealing security profiles disclose service weaknesses to potential threats by providing critical information about essential assets

Security Annotations: obfuscate security information and enrich service descriptions with a global security level

Annotation value: For a service $s$ that depends on $n$ assets, $x_1, \ldots, x_n$

$$V_C = \frac{\sum_{i=1}^{n} x_i \times w_i}{|A_s|}$$

$$x_i = \begin{cases} 
0 & \text{if } x_i \text{ is vulnerable} \\
1 & \text{if } x_i \text{ is invulnerable}
\end{cases}$$

Examples: Confidentiality, Availability, Supervision, …

Supervision $\subseteq$

$(\forall \text{ hasPertinentEssentialAsset.Message}) \land$

$(\forall \text{ hasPertinentEssentialAsset.BusinessObject}) \land$

$(\forall \text{ hasPertinentEssentialAsset.HostingServer}) \land$

$(\forall \text{ hasPertinentEssentialAsset.OperatingSystem})$
A Service Monitoring System for Vulnerability Detection

Public Vulnerability Databases
- National Vulnerability Database (NVD)
- Open Source Vulnerability DataBase (OSVDB)
- United States Computer Emergency Readiness Team (US-CERT)

The Common Platform Enumeration (CPE)
cpe://{part}:{vendor}:{product}:{version}:{update}:{edition}:{language}
Thank you

Questions?