Adapting the Model Driven Security strategy to generate contextual security policy for multi-cloud systems

Authors: W. F. Ouedraogo\textsuperscript{1}, F. Biennier\textsuperscript{2}, P. Ghodous\textsuperscript{3}

\textsuperscript{1,2}Université de Lyon, CNRS INSA-Lyon. LIRIS. UMR5205. F-69621. France,
\textsuperscript{3}Université de Lyon, CNRS Université Claude Bernard Lyon 1. LIRIS. UMR5205. F-69621, France

E-mail: wendpanga-francis.ouedraogo@liris.cnrs.fr\textsuperscript{1}, frederique.biennier@liris.cnrs.fr\textsuperscript{2}, ghodous@liris.cnrs.fr\textsuperscript{3}
Plan

- Context
- State of the art
- Model-Driven Security approach
- Conclusion and further works
Globalized economic environment involve for companies to:
- focus on their core business
- develop new collaborative strategies

→ build their IS (Information System) around on the Business Process (BP).

SOA (Service Oriented Architecture) provides companies a new model [1]:
- Build activities functionalities as business services and combine them dynamically with the partner companies service.
  → Interoperable, and agile services;
  → Open system mean security threats

Collaborative IS involve to share data, service and BP (Business Process) coming from different companies.
  → companies assets, which required to be protected
  → each has its own security policies
Context

- To protect IS: EBIOS, MEHARI, OCTAVE approach [8]
  - Approach based on the vulnerabilities and threats analysis,
  - Use knowledge bases
  - Not adapted to the dynamic environment imposed by process and SOA
  - Difficult and so long to implement
  - Not end users oriented (security expert is required)

- Cloud computing [3] emerge thank to:
  - Web 2.0
  - Development of broadband and network,
  - Virtualization
  - New solution to consume services and deploy collaborative IS (BP)
  - Allow to have on demand “unlimited” capacity for storage and processing
  - Involve a externalization strategy and new challenges to secure the SI
Context

Challenge

Our approach based on a Model-Driven Engineering (MDE).

- identify BP security requirements of each company,
- define an adapted Quality of Protection,
- generate adapted security policies, paying attention on the deployment platforms.
State of the art

Business process modeling

- Various types of modeling tools and languages: EPC, BPEL, WS-CDL, XPDL, BPMN, ...

- BPMN is mostly used to describe flows between the different activities as well as “launching” conditions of a particular part of the process.
# State of the art

## Secure BP

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstractions levels</td>
<td>PIM-PSM-code</td>
<td>PIM-PSM-Code</td>
<td>CIM-UML Use case (PIM)</td>
<td>PIM-PSM</td>
</tr>
<tr>
<td>Approach used</td>
<td>UML</td>
<td>Annotation based+ UML</td>
<td>UML</td>
<td>Annotation</td>
</tr>
<tr>
<td>Oriented end user</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Automatic Policy generation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Modification language and transformation</td>
<td>UML+DSL</td>
<td>UML2+SECTET-DSL</td>
<td>UML +QVT</td>
<td>Ad-hoc</td>
</tr>
<tr>
<td>Take account infrastructure</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Take account execution context</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy monitoring</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SecaaS (security as a Service)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Security Standard</td>
<td>XACML</td>
<td>SAML, WS-policy, XACML</td>
<td>XACML</td>
<td>XACML</td>
</tr>
</tbody>
</table>
State of the art

Cloud security

  - Define cloud security cube model that allows companies to choose the type of cloud that is adapted to their business needs
**State of the art**

**Conclusion**

**Business and application level**

- Do not pay more attention on vulnerabilities of infrastructure
- Not end user oriented.

**Infrastructure level**

- Customers don’t trust providers
- Difficult for providers to enforce each company policies.

![Diagram showing process 2.0](image-url)
Model-Driven Security approach
Model-Driven Security approach

Multidimensional model to secure BP
Multidimensional model

Weaving BP/Security: Business Security context Model

Business security context model
Multidimensional model

Weaving Deployment/Security: Platform Security context Model
Model-Driven Security approach

MDS Approach

Resources → Process Questions Pattern

Define

Business Context

CIM

Identify

Security Requirement

PIM

Platform Question Pattern

Security Pattern

Platform

Define

Platform Context

PSM

Abstract Security Policy

Generate

Security Policy

Generate

Runtime policy code

Model@Runtime
Model-Driven Security approach

CIM specification

R = (N, T, L, {R})
- N: Resource Name
- T: Resource Type
- L: Resource Layer
- U: the Resource URI (reference)
- R: Related Resources

All the resources as:
Res = \{R_i\} where 0 < k < N_k; (1)
Where “i” is the resource number and N_i the total of the all resources.

Res(R_k) = \{ r / r \in Res \land r.N = R_k \} (2)
Where R_k is the resource Name

R1(A1 "Activity" "Business", "http://com.insa.bp/connecteur/A1",[S1])
R11("S1" "Service" "Business", "http://com.insa.bp/connecteur/A1",[D11,S1])
R3(A3 "Activity" "Business", "http://com.insa.bp/connecteur/A3",[S3])
R8("A8" "Activity" "Business", "http://com.insa.bp/connecteur/A8",[S8])
R81(S81 "Service" "Service", "http://com.insa.bp/connecteur/A8/S81",[D811,D812])
R82(S82 "Service" "Service", "http://com.insa.bp/connecteur/A8/S82",[D821,D822])
R811(D811 "Data" "Service", "http://com.insa.bp/connecteur/A8/S81/D811",[])
R812("D821","Data","Service","http://com.insa.bp/connecteur/A8/S81/D821",[])
....
## Model-Driven Security approach

### CIM specification

![Diagram of Model-Driven Security approach]

### Questions and answers

<table>
<thead>
<tr>
<th>№</th>
<th>Questions</th>
<th>answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Which services or activity manipulate personal data?</td>
<td>Any services and process</td>
</tr>
<tr>
<td>2</td>
<td>Which services or process manipulate financial data?</td>
<td>Any services and process</td>
</tr>
<tr>
<td>3</td>
<td>Are there some activities in the process that handled data Strategic order (ie giving a strategic advantage to your business or associated with knowledge / expertise giving you a strategic advantage)?</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>Are there activities that involve external stakeholders (partners, customers, ...)?</td>
<td>[A9, A10, A13]</td>
</tr>
<tr>
<td>7</td>
<td>Are there activities for which you wish to restrict access to specific time slots (eg access between 7 and 19h on working days)</td>
<td>Yes, ALL</td>
</tr>
<tr>
<td>9</td>
<td>Which means can you use to access to the resources (data or applications): - A public network (public Wifi, 3G network, home network of personal ...) - the company network (LAN, VPN) -Any Network</td>
<td>Any network</td>
</tr>
</tbody>
</table>
Model-Driven Security approach

CIM To PIM : define security requirement

\[ M_{\text{CIM2PIM}}(\text{Res}_1, \text{Res}_2, \ldots, \text{Res}_n) \rightarrow \mathcal{M}_{\text{CIM2PIM}}(\text{Res}_1, \text{Res}_2, \ldots, \text{Res}_n) \] (8)

\[ \text{Reqi} = (\text{RR}, (\text{RT}, \text{RM}), \text{RG}, \{\text{RCtx}\}) \]

- RR (Requirement Resource)
- RT (Requirement Type)
- RM (Requirement Metrics) = [0-1]
- RG (Requirement Goal)
- RCtx (Requirement context)

All the requirements for all the resources as:
\[
\text{Reqs} = \{\text{Reqi}\} \text{ where } 0 < i < N; \text{ (3)}
\]
Where “i” is the requirement number and N the total of the all requirements of all resources.

The requirements associated to the resource \( R_k \) is:
\[
\text{Reqs}(R_k) = \{ r \mid r \in \text{Reqs} \land r.PR=R_k \} \text{ (4)}
\]
Model-Driven Security approach

CIM To PIM: define security requirement

Algorithm 2: Extrait du fichier ATL de transformation CIM TO PIM

```
context Requirement

let secReq : Resource = secReq

let maxMetric = getMaxMetric()

let level = getState().getString()

let autho : SecReq = autho

let authoContext : AuthContext = authoContext

let authorization : Boolean = true

if (secReq.hasWho())
    authorization = false

if (secReq.hasWhen())
    authorization = false

if (secReq.hasFromWhere())
    authorization = false

if (secReq.hasHow())
    authorization = false

if (secReq.hasWhoContext())
    authorization = false

if (secReq.hasWhenContext())
    authorization = false

if (secReq.hasFromWhereContext())
    authorization = false

<securityreq:Requirement>
    <context type="Who"/>
    <context type="How"/>
    <context type="When"/>
    <context type="Where"/>
    <condition key="AccessMode" value="[role]"/>
    <condition key="Shared" value="true"/>
    <condition key="users" value="[Production]"/>
    <condition key="Device" value="[PC]"/>
    <condition key="NetWork" value="[Public, private]"/>
    <condition key="Temporal" value="true"/>
    <condition key="Location" value="[IPDomaine]"/>
</securityreq:Requirement>
```
Model-Driven Security approach

**PIM To PSM : security pattern**

Patj= (PatN, PatG, PatTech, {PatR}, {PatM} {PatL}, {PatCtx}, {PatCol}, {PatParm})

- **PatN** : pattern’s name;
- **PatG** : pattern’s goal;
- **PatType** : Abstract or technical pattern
- **{PatL}** : pattern’s layers;
- **{PatCtx}**: Pattern context (set of conditions and obligations)
- **{PatR}** : related patterns (sub-patterns);
- **{PatM}**: set of level of protection offer by the pattern
- **{PatCol}** : pattern collaboration;
- **{PatParm}** : setting elements;
Model-Driven Security approach

PIM To PSM : define abstract policy

Polx = (PR, PT, PG, PL, PM, {PC}, PP)

- PR : policy resource;
- PT : Policy type
- PG : Policy goal
- PL : the layer of this policy
- PM : the metric of this policy
- {PR} : the policy rules
- PP : the pattern to use

All the policy rules of all resources as:
Pols = {Polj} where 0 < j < Nj ; (6)

The policies rules associated to the resource Rk is:
Pols(Rk) = {p / p ∈ Pols ∧ p.PR=Rk} ; (7)
Model-Driven Security approach

**PDM specification**

Plat= (PlaN, PlaT, PlaTst,{PlaSM})

- PlaN : platform provider;
- PlaT : platform type (public, comminatory, private,..)

PlaTst: the level of client Trust to the platform

{PlaSM} : Security mechanisms implemented

**Questions**

**Deployment platform specification**

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who manages the Cloud infrastructure? You (the company) or the service provider?</td>
<td>The service provider</td>
</tr>
<tr>
<td>Where are data stored? Inside your company boundaries or outside.</td>
<td>Outside</td>
</tr>
<tr>
<td>Who owns the data? You (The company) or service provider?</td>
<td>The company</td>
</tr>
<tr>
<td>Is Cloud infrastructure shared to another’s companies?</td>
<td>yes</td>
</tr>
<tr>
<td>Do infrastructure provides backup and versioning systems to restore the system in case of an incident?</td>
<td>No</td>
</tr>
<tr>
<td>Does Infrastructure provide services and protocols to secure communications (VPN, HTTPS, …)?</td>
<td>Yes</td>
</tr>
<tr>
<td>Does Infrastructure provide security services and APIs to control access to business services and data?</td>
<td>No</td>
</tr>
<tr>
<td>Does infrastructure is certified (ISO 27001 certification, SAS 07, FISMA,)?</td>
<td>No</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>
Model-Driven Security approach

PSM To PSM: risk analysis and assessment

\[ \text{Risque} = \text{NEP} \times \text{NPVP} \times \text{NI} = (\text{NEP} \times (1-\text{trust+e}) \times \text{NI} \] (17)

\[ R(A08) = (\text{NEP}=0.75) \times (\text{NPVP}=1-0.36) \times (\text{NI}=1) = 0.48 \]

Protection level assessment grid
Model-Driven Security approach

PSM To PSM : Security policy generation

Model-Driven Security approach

PSM To PSM : Security policy generation
Model-Driven Security approach

Model@Runtime: Security architecture
Model-Driven Security approach

**Model@Runtime**: execution context
Model-Driven Security approach

Security components implemented as SecaaS
## Conclusion

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstractions levels</td>
<td>CIM- PIM- PSM-code</td>
<td>PIM-PSM-code</td>
<td>PIM-PSM-Code</td>
<td>CIM-UML Use case (PIM)</td>
<td>PIM-PSM</td>
<td></td>
</tr>
<tr>
<td>Approach used</td>
<td>Annotation based + UML</td>
<td>UML</td>
<td>Annotation based+ UML</td>
<td>UML</td>
<td>Annotation</td>
<td></td>
</tr>
<tr>
<td>Oriented end user</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Automatic Policy generation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Modelisation language and transformation</td>
<td>EMF+ATL+ Ad-hoc transformation</td>
<td>UML+DSL</td>
<td>UML2+SECTET-DL</td>
<td>UML +QVT</td>
<td>Ad-hoc</td>
<td></td>
</tr>
<tr>
<td>Take account infrastructure</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Take account execution context</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Policy monitoring</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>SecaaS (security as a Service)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Security Standard</td>
<td>XACML, SAML, WS-Security</td>
<td>XACML</td>
<td>XACML</td>
<td>SAML, WS-policy,</td>
<td>XACML</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion and further works

Conclusion

- Use model driven approach to:
  - Identity, for each enterprise, their business process security requirements.
  - Define an adapted Quality of Protection
  - Generate contextual security policies
- Define security architecture to take account the execution context
- Define standardized security mechanisms as SecaaS which are invoked according to the runtime context and allow end to end security

Further works

- Extend security pattern for privacy
- Monitoring security policies
References


Thank you for your attention
Model-Driven Security approach

Authentication sequence diagram

GetPolicyInServiceWSDL()
SendServiceCall & Policies()
GetUserContext()
BuildUserContext
SendUserContext()
If UserToken=null & Exist AuthenticationPolicy()
CallSecurityService
GetUserInfo()
SendUserInfo()
GetUserByID()
Match User Info Sended By User and Registry
If Authentication is Ok
CreateUserToken
UpdateSSORegistry()
SendAuthenticationResponse()
If UserToken!=null
Call businessService
SendBusinessServiceResponse
SendToken and Service response
StoreToken
Send BusinessResponse
CallService()
SendUserInfo
Model-Driven Security approach

Authorization sequence diagram