Model Driven Security Architecture

Anthony Nadalin
Distinguished Engineer
IBM Corporation

Presented by Mike Perks
Why Model and Types of Model

- A model is some form of representation designed to aid in visualizing a thing that cannot be observed directly
  - Abstract
  - Has not yet been constructed

- Not just used in IT

- Levels of Model Abstraction
  1. Conceptual
  2. Logical (usually platform independent)
  3. Physical or Virtual (usually platform dependent)
Defining Modeling

OMG defines a hierarchical 4 layer model

<table>
<thead>
<tr>
<th>Instances</th>
<th>Example(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>MOF</td>
</tr>
<tr>
<td>Hundreds</td>
<td>UML, BPEL, XML, DDL</td>
</tr>
<tr>
<td>Millions</td>
<td>StockShare, askPrice, sellLimitOrder, StockQuoteServer</td>
</tr>
<tr>
<td>Trillion's</td>
<td>&lt;Acme_Software_Share_98789&gt;, 654.56, sell_limit_order, &lt;Stock_Quote_Svr_32123&gt;</td>
</tr>
</tbody>
</table>

- **Metametamodel**: The infrastructure for a metamodelling architecture. Defines the language for specifying metamodels.
- **Metamodel**: An instance of a meta-metamodel. Defines the language for specifying a model.
- **Model**: An instance of a metamodel. Defines a language to describe an information domain.
- **User Objects**: An instance of a model. Defines a specific information domain.
Extending Modeling Languages

- OMG defines 3 ways a model can be extended:
  - Profile to annotate model using stereotypes
    - e.g. <<Service>> stereotype on a UML Operation
    - J2EE, Web Services Profiles already defined
  - Extend UML metamodel with new entities
    - e.g. Create Service as a subclass of UML “Operation” class
  - Create new metamodel
    - e.g. OMG Common Warehouse Metamodel (CWM)
## Defining Security

<table>
<thead>
<tr>
<th><strong>Primitive</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>audit</td>
<td>Used to denote that the specified communication is to be audited. One can assume that the audit will contain both the identity of all parties as well as any data communicated between the parties.</td>
</tr>
<tr>
<td>authenticate</td>
<td>Used to denote a party that has to be authenticated in the scope of a given context.</td>
</tr>
<tr>
<td>authorize</td>
<td>Used to denote that a communication between two parties must ensure that the requester is authorized to perform the request.</td>
</tr>
<tr>
<td>confidentiality</td>
<td>Used to denote that the marked information should be treated as confidential and that all reasonable effort (considering technical implications) should be made to ensure the data remains protected against unauthorized viewing</td>
</tr>
<tr>
<td>integrity</td>
<td>Used to denote that the marked information is guaranteed to reach the recipient unaltered during transit (<em>e.g.</em> includes the digital signatures of parties related to the document)</td>
</tr>
</tbody>
</table>

- Products use various security primitives in different combinations with overlaps and gaps
- Application Developers need a mental model of security
Problem

• Security is often an afterthought in the development lifecycle, thus remediation is costly both for vendor and customers. Security enabled tooling for inclusion at earlier stages of the software lifecycle is needed.
Customer Findings

Key Observations

- Integrating applications into the company security infrastructure is treated by customers as projects that each have a lifecycle.
- Application groups are “customers” of the security group maintaining the security environment and the security groups want to automate that process as much as possible.
- Identity and Access are integrated in customer environments.
- Every customer either needs an automated way to move application security policy from test into production, or have written their own – they all want it out of the box.
- Each customer building up expertise in putting applications behind WebSEAL on their own. In several companies, the “best practices” for putting applications behind proxy/gateways are “in someone’s head”.
- The transfer of security policy from the Security office to application groups is usually via some form of company documentation, combined with meetings with the security office to evaluate the risk.
- Security products are used in large scale environments.
- Customers are trying to figure out how to deal with the large amount of audit and log data that their security products produce.
- Everyone is trying to figure out how to deal with compliance issues.
Why Application Security Lifecycle?

- **CIO Pain to be addressed**
  - IT downtime caused by security attacks results in loss of credibility/market opportunity and added cost of recovery.
  - Too expensive and difficult to develop and deploy secure applications.

- **Solution / Vision**
  - Make it easy for customers to produce secure applications.
  - Ease the integration with security-based products for the entire application lifecycle.
Targeted Security Personas

Security Developers

- Need an interface to enable easy development
  - Customized end user GUIs
  - Customized modules to plug into security interfaces
  - Develop schemas, workflows and scripts
- Security Developers require support for the development lifecycle – Design, Construct and Test

Security Policy Creators

- Need an interface to easily develop security policy for a customers Identity and Access environment
  - Who is allowed to access to what resources (systems and applications)
  - How users get provisioned to those same resources
  - How users authenticate to the infrastructure
- Security Policy creation is a development activity, and needs to be supported throughout the security policy lifecycle – Design, Construct, Test and Provision (Deploy)
Eclipse as Development Platform

Security Developers

➢ Allow developers to leverage existing tools and environments
  • JSF GUI builders
  • Open source language support – XSL, Javascript, XPATH, ...
➢ Enable easy use of APIs, easy access to API documentation, along with code samples, and best practices and how to documentation
➢ Enable import, run and testing of samples
  • End User Samples – Self-registration, Change my password, Approvals

Security Policy Creators

➢ Allows Security tools to integrate with application servers and portal application security specifications
➢ Allow developers to leverage existing development tools and environments
  • Open source language support – XSL, Javascript,...
  • Visual representation of relationships
➢ Allow developers to construct, test, and provision policy artifacts such as:
  • Policy for Web Applications
  • Provisioning Policy
  • Partner SSO Contracts
On Demand Solution Lifecycle

Conceive & Modify Business Idea

Define Model
- Business Process Modeling
- Business Application Modeling

Implement Model
- Design and Implement

Acquire & Map to Infrastructure
- Solution Install
- Subscribe
- Manage/Administer

Monitor & React
- Monitor

Integration

Infrastructure Management
Lifecycle – Tools

PowerPoint etc
Security Officer, Privacy Officer

Corporate security and privacy policies; policies specific to business app

WBI Modeler
Business Analyst

Model security and privacy primitives to business processes; Model organization; etc.

WSAD-IE
Developer

J2EE deployment descriptors; WS-SecurityPolicy for WSDL; Application call outs, etc.

Security Management and Monitoring
Security administrator

Manage security and privacy policies; manage identities; monitor security events, compliance, etc.

Rational XDE
Solution Architect

Refine UML Model w/ security; Secure interaction diagrams, etc.
Mediated Exchange Scenario

Service Consumers
"We sell burgers. Let's get someone else to handle our travel stuff."

Mediator
"Hmm. There's a business opportunity here!"

Service Providers
"Let's make it easier for people to book with us."

Burgers Inc.
Executive
Traveler
Business Analyst
Manager (Travel Approver)

consumer1

TravelLinks.com

Executive
Business Analyst
Travel Analyst

Travel Process

consumer2

Fast Air Inc.
Executive
Business Analyst

airline

car rental

hotel

GDS
Example: Travel Application

- Travelers and Travel Agents may change Itinerary Data for planned trips for the purpose of addressing customer requests.
Travelers and Travel Agents may change Itinerary Data for planned trips for the purpose of addressing customer requests.

<role> Travel agent </role>
<role> Traveler </role>
<actions>
  <action> view </action>
  <action> change </action>
</actions>
<resource> itinerary data </resource>
<condition> purpose == customer request </condition>
Authoring the Policies

Traveler and travel agents can view or change traveler's itinerary.

Example Rule Guide:

[User Category(ies)] can [Action(s)] [Data Category(ies)] for the purpose(s) of [Purpose(s)] if [optional Condition(s)] with [optional Obligation(s)].

1. Travel Agents can collect and use customer name, contact information, credit card information, and customer preferences for the purpose of purchasing airline tickets.

2. Customer Service Reps can use customer name and contact information for the purpose of notifying customers of flight changes.

3. Marketing Reps can use customer name and contact information for the purpose of sending marketing information if the customer has opted-in.

Traveler and travel agents can view or change traveler's itinerary.
Policy Element Identification

The elements of the privacy policy rules are summarized below. Select each rule in the policy to review the elements highlighted as part of the policy.
Policy Authoring

- Create and customize policy templates
  - Pre-built templates for Travel industry
  - TravelLinks will customize TravelIndustry template further to their business

- Policy officers author policies
  - By inheriting templates and adding their specific business requirements
  - Helps them to inherit industry requirements and reflect business specific requirements

- Tools to author policies
  - Allow usage of templates
  - Reflects the skills set of these policy officers (*e.g.*, powerpoint, Word doc, *etc.*)

- Policies are stored
  - In a form they can continue to work with
  - Tools allow export/importing these formats into some form (*e.g.*, XML) that can be parsed and later used by tools and runtime
Business Process Modeling

Service level offerings added e.g., Platinum service: Gold Security (high confidentiality) and Gold Response (<1 sec).

Authorize: Traveler and travel agent to view and change itinerary.

“Platinum” Service Level Offering specified on “Travel” process.
Task: Organization Modeling

Model organization, roles and users including management

Model roles

Collapsible Views

Drag & Drop

Organizational Template
## Defining Primitives to Model Security Policies

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audit</td>
<td>Used to denote that the specified communication is to be audited. One can assume that the audit will contain both the identity of all parties as well as any data communicated between the parties.</td>
</tr>
<tr>
<td>authenticate</td>
<td>Used to denote a party that has to be authenticated in the scope of a given context.</td>
</tr>
<tr>
<td>authorize</td>
<td>Used to denote that a communication between two parties must ensure that the requester is authorized to perform the request.</td>
</tr>
<tr>
<td>confidentiality</td>
<td>Used to denote that the marked information should be treated as confidential and that all reasonable effort (considering technical implications) should be made to ensure the data remains protected against unauthorized viewing</td>
</tr>
<tr>
<td>integrity</td>
<td>Used to denote that the marked information is guaranteed to reach the recipient unaltered during transit (e.g. includes the digital signatures of parties related to the document)</td>
</tr>
</tbody>
</table>
Modeling and Security Policies

```
Reservation
- viewItinerary()
- changeReservation()

SecurityAuthorization
- AccessPolicy
  - roles: String

SecurityConditions
- access_time = businesshours
- access_type = intranet
- purpose = ticketing
```

```
reservation:Reservation
airlineProvider:AirlineProvider

1: makeAirlineReservation
```
Transformation – Modeling to Implementation

- Implementing modeled artifacts
  - Auto-generate code as much as possible
  - Manually implement the artifacts

- Auto-generated code
  - Infrastructure-managed security
    - To include policies declared as deployment artifacts
    - *e.g.*, J2EE deployment descriptors
  - Application-managed security
    - To insert call outs to security services based on security profiles

- Declarative policies
  - Helps manage security policies without changing application
  - (a) policies to be enforced internally (service consumer awareness not required)
  - (b) policies to be followed by service consumer (awareness not required)
  - Recommended approach – leverages infrastructure
  - J2EE deployment descriptors, web service policies, *etc.*

Colorado Software Summit: October 23 – 28, 2005 © Copyright 2005, IBM Corporation
Sample Security Policies in WSDL and J2EE (After Implement Stage)

**WS-Policy**

- **Service side**

  ```xml
  <wsp:Policy>
    <wsp:ExactlyOne>
      <wsse:SecurityToken> username </>
      <wsse:SecurityToken> X509 </>
    </ExactlyOne>
    <wsp:AttributeCert Issuer=TravelOrg>
      wsp:optional=true
    </wsp:AttributeCert>
    <wsse:Confidentiality>
      <wsse:Issuer=Verisign>
      </wsse:Issuer>
    </wsse:Confidentiality>
    <wsp:Privacy>
      <wsp:PolicyReference
        URI="TravelIncPrivacyPolicy"
    </wsp:Privacy>
  </wsp:Policy>
  ```

**J2EE Deployment Descriptors**

- **EJB Role assignment**

  ```xml
  <method-permission>
    <role-name>Traveler</role-name>
    <role-name>TravelAgent</role-name>
    <method>
      <ejb-name>ReservationBean</ejb-name>
      <method-name>view</method-name>
      <method-name>change</method-name>
    </method>
  </method-permission>
  ```
<Policy PolicyId="P1" PolicyCombiningAlgoId="path-more-specific-deny-overrides-with-propagation">
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="user-role-match">
          <SubjectAttributeDesignator AttributeId="subject-id" DataType="string"/>
          <AttributeValue DataType="string">traveler</AttributeValue>
        </SubjectMatch>
        <ResourceMatch MatchId="path-match">
          <AttributeValue DataType="pattern-path">/Itinerary</AttributeValue>
        </ResourceMatch>
      </Subject>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="path-match">
          <AttributeValue DataType="simple-path" AttributeId="resource-id"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch MatchId="action-id">
          <ActionAttributeDesignator AttributeId="subject-id" DataType="string"/>
          <AttributeValue DataType="string">view</AttributeValue>
        </ActionMatch>
      </Action>
    </Actions>
  </Target>
</Policy>

<Rule RuleId="R1" Effect="Permit">
  <Target>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="path-match">
          <AttributeValue DataType="pattern-path">/Itinerary</AttributeValue>
        </ResourceMatch>
      </Resource>
    </Resources>
  </Target>
</Rule>
Binding Policy Elements to Application Environment

- Role (from business vocabulary) – to – enterprise identity (user, groups, etc.) relationship
  
  - *e.g.*, TravelAgent -> “TravAgentGroups” in LDAP, or “users whose job.category=agent”

- Resource (from business vocabulary) to information/application artifacts
  
  - *e.g.*, “itinerary data” to “TravelInformationTable” on DB, accessed through “getItinerary” on application
  
  - Needs meta-data about information and mapping these

- Action (from business vocabulary) to runtime operations
  
  - *e.g.*, “view” to “getItineraryData”, GET on HTTP, ..., etc.
Using Policy for Application Provisioning

1. Application Added
2. Access Policy evaluated
3. Approvals gathered
4. Accounts created with Authorization

**Provisioning Policy**
Add entitlement for Service to Dynamic Org Role **Travel Agents Role** which includes **Travel Agent** group

Workflow provisions user to application
Conceptual

• High level ideas and concepts
• Behavior and data requirements gathering

Logical

• Database-Independent
• Logical data requirements
• Entities, relationships, rules
• Dimensions, Facts
• Highly Normalized

Physical

• Database-Dependent
• Heterogeneous, distributed
• Tables, XML, Messages, Legacy
• Star, snowflake schemas
• Denormalized, optimized
• Partitioning, Storage modeling

Virtual

Anthony Nadalin — Model Driven Security Architecture
Monitoring

- Conceptual flows (may include event emitters)
- Conceptual documents
- Statistics / history / thresholds / limits

**Integrate**
- Listen to events carrying data for business metrics
- Deploy event emitter configurations

**Model**
- Monitor
- Audit provider - To audit any event

**WebSphere**
- Applications (e.g. SAP)
- Enterprise Applications
- Legacy Systems
- Operating Systems
- Applications

**Tivoli**
- Correlate, analyse, react
- RiskManager, IDS Monitors, etc.

**Correlation Engine**
- Notify of business situations discovered

**Security events**
- Event Bus
- Hardware
- Operating Systems
- Applications

**WBI Infrastructure**
- WebSphere Applications
- Shared Event Bus