Securing Web Services –
From Encryption to a Web Service Security Infrastructure

Kerberos
WS-Security
X.509
TLS
Gateway
OWSM
WS-Addressing
WS-Policy
Agent
SAML
XML Signature
Policy Manager
XML Encryption
wsmAdmin

Peter Lorenzen
Technology Manager
WM-data Denmark
a LogicaCMG Company
peloz@wmdata.com
• Describe 3 ways to **secure** Web Services **without programming**

• Security **measures**
  – Transport Layer Security (TLS/SSL/VPN)
  – Message Level Security (WS-Security)
  – Oracle Web Service Manager (OWSM)

• Discus **when to use** which

• Focus on **OWSM**
Agenda

• Intro
• Transport Layer Security
• Message Level Security
• When to use which?
• Web Service Security Standards
• Oracle Web Service Manager
  – OWSM – SOA and Identity Management
  – OWSM Control
  – OWSM Gateways
  – OWSM Agents
  – OWSM Policy Manager
  – OWSM Monitoring
  – Conclusion on OWSM
• Conclusion
• In the past security has often been coded directly into the Web Services
  – No standards
  – Difficult to maintain, administer and monitor

• I have **focused** on 3 methods that requires **no programming**
  – Transport security e.g. TLS/SSL/VPN (The most widely used)
  – WS-Security (Open Standard)
  – OWSM (Build on WS-Security and related standards)
Security are Many Things

- **Authentication**: Who are you?
- **Authorization**: OK we know you, what are you allowed to do?
- **Integrity**: Has anybody tampered with my message?
- **Confidentiality**: How to keep the message secret?
- **Non-Repudiation**: I know you received the message and I can prove it!

All are important
Simple Use Case

Internet

Message

Web Service consumer

Web Service provider
Request

<env:Envelope>
  <env:Header>
  
  </env:Header>
  <env:Body>
    <ns0:getsalaryElement>
      <ns0:pEmpId>100</ns0:pEmpId>
    </ns0:getsalaryElement>
  </env:Body>
</env:Envelope>

Response

<env:Envelope>
  <env:Header>
  
  </env:Header>
  <env:Body>
    <ns0:getsalaryResponseElement>
      <ns0:result>24000</ns0:result>
    </ns0:getsalaryResponseElement>
  </env:Body>
</env:Envelope>
Transport Layer Security (TLS/SSL/VPN)

- Known
- Mature
- Simple
- Cheap

- Point-to-point
- Last mile security
- Encryption overhead
- Secure logging

Web Service consumer

Web Service provider

Web Service Provider 2

Authentication
- Basic auth. in HTTP header
- Two-way SSL

Internet

Message

SSL
Message Level Security (WS-Security)

- SOAP based Security
- Build on existing standards
- **Directly supported** in JDeveloper and Oracle Application Server

- **Encrypt** the message or part of it
- **Sign** the message or part of it
- Pass security tokens, signatures and trust assertions in the message
- All the security setup information goes in the **SOAP header**

**Pros:**
- End-to-end
- Secure logging
- Flexible
- Standard based

**Cons:**
- Complex
- “New” but maturing
- Message overhead

WS-Security

- X.509 Certificates
- Kerberos
- SAML
- XML Encryption
- XML Signature

a.k.a. XML-Dsig, XML-Sig
WS-Security SOAP Envelope Example

- Request and Response using **encryption** and **signing** (3DES, Sha1WithRSA)
- Key size 128 bytes

![Request and Response Sizes](request-x509.xml response-x509.xml)

<table>
<thead>
<tr>
<th></th>
<th>No Security</th>
<th>WS-Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>754 bytes</td>
<td>8074 bytes</td>
</tr>
<tr>
<td>Response</td>
<td>770 bytes</td>
<td>6605 bytes</td>
</tr>
</tbody>
</table>
When to use which?

**TLS/SSL/VPN**
- Point-to-point
- **Few** Service Providers
- **Limited knowledge** about identity management and encryption
- No SOA strategy
- No demand for Secure Logging

**WS-Security**
- SOA strategy
- Flows/Orchestrations
- Context-based Routing
- LDAP integration
- Need for
  - Authorization
  - Integrity (after transport)
  - Secure logging
  - Non-repudiation
  - Higher level of security such as biometrics etc.
<table>
<thead>
<tr>
<th>Standard</th>
<th>Support Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS-Federation</td>
<td>✔</td>
</tr>
<tr>
<td>WS-Authorization</td>
<td>✔</td>
</tr>
<tr>
<td>WS-Secure-Conversation</td>
<td>✔</td>
</tr>
<tr>
<td>WS-Policy Attachment</td>
<td>✔</td>
</tr>
<tr>
<td>WS-Policy</td>
<td>✔</td>
</tr>
<tr>
<td>WS-Addressing</td>
<td>✔</td>
</tr>
<tr>
<td>WS-Reliability</td>
<td>✔</td>
</tr>
<tr>
<td>WS-Trust</td>
<td>✔</td>
</tr>
<tr>
<td>WS-Security</td>
<td>✔</td>
</tr>
</tbody>
</table>

- **WS-Security Standards**
  - SOAP

- OWSM 10g Support
  - OWSM 11g Support
Oracle Web Service Manager (OWSM)

- Standalone platform for **securing** and **managing** access to Web Services
  - Used by a developer, deployer or security administrator
  - **Declarative**: Does not require modification of existing applications
  - Policies updated in **real time**
  - **Monitors** access-control events
  - Defines and monitors against **SLAs**
  - **Leverages existing** identity and access management standards and architecture

Oracle acquired Oblix in 2005
Original Oblix COREsv
Simple Use Cases

- Client Gateway
- OWSM OAS Web Service
- Agent + Web Service
- DB Client

Diagram:
- Client
- Gateway
- OWSM OAS
- Web Service
- Agent + Web Service
- Policy Manager and Monitor
- DB
## OWSM Component Overview

<table>
<thead>
<tr>
<th>Component</th>
<th>Components and Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWSM Control</td>
<td>• Enterprise Manager&lt;br&gt;• Policy Manager (Rules for PEPs)&lt;br&gt;• Monitor</td>
</tr>
<tr>
<td>Policy Enforcement Point (PEP)</td>
<td>• Gateways (Proxy in the OAS)&lt;br&gt;• Agents (Client or Server side)</td>
</tr>
<tr>
<td>Database</td>
<td>• Policies, SLA&lt;br&gt;• Monitor data&lt;br&gt;• Users and roles</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>• Java KeyStore, Oracle Wallet&lt;br&gt;• LDAP, OID&lt;br&gt;• Oracle Identity Manager</td>
</tr>
</tbody>
</table>
OWSM Security Principles

BUILD
Policies

ENFORCE and EXECUTE
Policies

Web Services Clients

Applications or Web Services

MONITOR
Policies

Policy Manager

Gateway

Monitor

Agents
• Enterprise Manager **plug-in**

• Used for
  
  – **Configuring** Gateways and some Agents
  – **Monitoring** Gateways and some Agents
  – Defining **Alerts** and **SLA thresholds**
  – Defining Custom **Reports/Views**
OWSM Gateways

- **Proxy** that runs in the OAS
- **Accepts** or **rejects** incoming requests like a firewall
- **Transport protocol** **translation**
  - Incoming: HTTP(S), JMS, MQ
  - Outgoing: HTTP(S), JMS, MQ, or a custom protocol
- Supports both SOAP and standard XML messages
- **Content Routing**
- **Logging point**
- “No end-point security”

See Oracle Web Services Manager Extensibility Guide

A client must use a **different URL** to access a Gateway than the one for the Web Service
OWSM Agents

- Deployed with the **Web Service** or **Client** (Consumer)
- **Accepts** or **rejects** incoming requests like a firewall
- Can **enrich** the message with security
- Logging point
- End-point security
- **On-line** or **off-line** with OWSM Policy Manager
- Use cases
  - Missing support for security protocols
  - Last mile security
- **Java Container** support: OC4J, AXIS, WEBLOGIC, WEBSPHERE, TIBCO-BW

An OWSM **must** be installed on the machine where you are installing the client agent.
Installing Agents

- Agents are installed via the **WSMadmin** command line tool
- Situated in ORACLE_HOME\owsm\bin
- Example
  - `wsmadmin installAgent`
- The **agent.properties** file contains the setup information

```python
agent.componentType=OC4JClientInterceptor
#
# agent.componentType can be AXIS, WEBLOGIC, WEBSHERE, TIBCO-BW
#
# Specifying the container version for OC4J 10.1.3 is a must
#
# The only allowed container type for Interceptor agent is OC4J a
#
agent.containerType=OC4J
agent.containerVersion=10.1.3
#
# oc4j.home - Absolute path to the OC4J Installation
# This property is only needed for Server Interceptor agents
# the path should point to a valid OC4J 10.1.3+ installation

oc4j.home=D:/product/10.1.3.1/OracleAS_1
oc4j.j2ee.home=D:/product/10.1.3.1/OracleAS_1/j2ee/home
external.oc4j.port=80
external.oc4j.adminPort=6000
external.oc4j.adminID=oc4jadmin
#
# client.home - Absolute path to the Client into which Interceptor
# This property is only needed for Client Interceptor agents

client.home=c:/oracles/client
```
Some Uses Cases

Client

Client Gateway Agent + Web Service

OWSM OAS

Gateway

Policy Manager and Monitor

Web Service

Agent + Web Service

Client + Agent

Client + Agent

Client + Agent

Agent + Web Service

DB
• **Policy** - Set of operational tasks that are performed at a PEP
• **Policy Step** – One operational task. Ex. Decrypt
• A Policy is separated into
  – **Request Pipeline** – A set of policy steps that are executed during the processing of a Web Service request
  – **Response Pipeline** – A set of policy steps that are executed during the processing of a response to a Web service request
• **Pipeline Template** – Reusable Policy Pipeline
• Policies are **versioned**
  – The most current version is the policy that is enforced
  – You can revert to old versions

Remember to **purge** obsolete policies once in a while
• **Build-in Policy Steps** examples
  - Active Directory Authenticate
  - Active Directory Authorize
  - Decrypt and Verify Signature
  - LDAP Authenticate
  - LDAP Authorize
  - Log
  - SAML - Verify WSS 1.0 Token
  - Sign Message
  - Sign Message and Encrypt
  - Verify Certificate
  - Verify Signature
  - XML Decrypt
  - XML Encrypt
  - XML Transform
Policy Example

Pipeline: "Request"

Pipeline Steps:

Start Pipeline

Log

Decrypt and Verify Signature

End Pipeline

Pipeline: "Response"

Pipeline Steps:

Start Pipeline

Log

XML Encrypt

Sign message

End Pipeline
• Collects metrics from Gateways and Agents
• Views/Reports
  – Snapshot for a component and service
  – SLA compliance report
  – Execution Details view
  – Message Logs
  – Flows: Grouping of invocations within some context
  – Security Statistics (Access control)
  – Service Statistics (Latency Variance and Traffic Analysis)
  – My Views: Custom views
• Alarms: Create rules for e-mail alerts

By default, the monitor data is only persisted for the last 100 minutes. Max. is 60 days.
Monitoring Examples

Snapshot

- Overall Statistics
  - Overall Failures: 2%
  - Overall Latency: 203

- Security Statistics
  - Authentication Failures: 0%
  - Authorization Failures: 0%

- Service Statistics
  - Service Failures: 0%
  - Service Latency: 29
SLA Compliance

Monitoring Examples

SLA Compliance

| Component: MyGateway | Service ALL SERVICES | Start: Tuesday, November 13, 2007 11:11:00 AM | End: Tuesday, November 13, 2007 12:11:00 PM |

Percentage over time

- Success with low latency (below SLA)
- Success with high latency (above SLA)
- Failure

Click on bars to drill down into details, move your mouse over to see more information.
• Flow – **Collection of Web service**, grouped together within some context

• Ex. services required to fulfill a client request in processing an order.

• **Unique flow id** inserted into the SOA header by the Web Service consumer

• Tags defined by **WS-Addressing**

• OWMS extracts and stores flow IDs to **correlate** the messages containing the same IDs

```xml
<soap:Header>
    <wsa:RelatesTo RelationshipType="cswm:ParentContext">
        uuid:8EB9-C6A3-75AA-7EBA
    </wsa:RelatesTo>
<soap:Header>
```
What You Get From OWSM

- Security that is based on **Open Standards**
- **Declarative security** that makes it easy to make changes to your security settings
- “**Single point**” for monitoring Web Service security
- Lots of ways to **report** the system status
- **Mail alerts** if there are problems
- **SLA documentation**
What You Don’t Get From OWSM

• Help with key and certificate management
• Not possible to use a public key included in a request to encrypt the response
• You can not backup policies or export/import them
• Unified Enterprise Manager interface
Conclusion on OWSM?

- **Easy** to use

- Use it for
  - SOA strategy
  - Flows/Orchestrations
  - Identity Management integration
  - Higher level of security

- I think Gateways are **more useful** than Agents
  - No installation is required
  - An Agent requires its own OWSM installation
Conclusion

- Transport layer security (TLS/SSL/VPN), WS-Security and OWSM are all viable tools for securing Web Services in an Oracle environment

- Choose the right tool from your situation
  - Knowledge
  - Point-to-point or end-to-end
  - Number of service providers
  - SOA strategy
  - Flows/Orchestrations
  - Identity Management integration

- The future
  - Belongs to the WS-* standards
  - If you have external partners you might be forced to use WS-Security etc.
Securing Web Services – From Encryption to a Web Service Security Infrastructure

Questions?

Contact Information

Peter Lorenzen
peloz@wmdata.com

For More Information

- http://www.wiki.oracle.com/page/Web+Service+Manager
- Securing Web Services with WS-Security by Jothy Rosenberg & David Remy