A Guideline to Secure Web Services

Anoop Singhal
Computer Security Division
NIST
anoop.singhal@nist.gov
Outline

- Web Services and their Relation to Security
- Dimensions for Secure Web Services
- Web Services Security Standards
- Secure Implementation Tools and Techniques
- Challenges and Conclusions
What are Web Services?

- Today, we normally use Web browsers to talk to Web sites
  - Browser names document via URL (lots of fun and games can happen here)
  - Request and reply encoded in HTML, using HTTP to issue request to the site
- Web Services generalize this model so that Applications can talk to Applications
Web service definition

“A Web Service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in WSDL. Other systems interact with the Web service in a manner prescribed by its description using SOAP messages and XML.”

Source: http://www.w3.org/TR/ws-arch/
Web Services Example

1. SOAP request
2. SOAP response
3. SOAP request
4. SOAP response

Loan Service
Credit Service
Rate Service
Web Services Example

Diagram:
- UDDI Registry
- Rate Service
- Loan Service

1. Rate Service requests WSDL from UDDI Registry.
2. UDDI Registry responds to UDDI inquiry.
3. UDDI Registry sends UDDI response.

Actions:
- UDDI inquiry
- UDDI response
- Rate Service
- Loan Service

Notes:
- UDDI: Universal Description, Discovery, and Integration
- WSDL: Web Services Description Language
Web Service Example

1. User makes a request.
2. SOAP request to Loan Service.
3. SOAP response from Loan Service.
4. HTML page from Web Portal.

Web Portal

Loan Service
Advantages of web services?

- Web services provide interoperability between various software applications running on various platforms.
  - “vendor, platform, and language agnostic”
- Web services leverage open standards and protocols. Protocols and data formats are text based where possible
  - Easy for developers to understand what is going on.
- By piggybacking on HTTP, web services can work through many common firewall security measures without requiring changes to their filtering rules.
The Web Services “stack”

- **BPEL4WS (IBM only, for now)**
- **Business Processes**
  - Reliable Messaging
  - Security
  - Transactions
  - Coordination
  - Quality of Service
  - WSDL, UDDI, Inspection
  - SOAP
  - Other Protocols
  - Messaging
  - Description
  - TCP/IP or other network transport protocols
  - Transport
Elements of Security

- Authentication
- Authorization
- Integrity
- Non-repudiation
- Confidentiality
- Privacy
Dimensions for Secure Web Services

- Secure Messaging
  - HTTPS
  - XML Encrypt. XML Digital Sig.
  - WS-Security

- Resources Protection
  - Access Control
  - Authorization
  - Protection from DOS Attacks

- Contracts and Obligations

- Trust Management
Components to be secured

Securing the network resource

Securing the application: Identification; Authentication; Authorization

Browser

Securing network traffic

Firewall

Web server

Apps

Host

Firewall

Application server

Apps

Host

Database server

Database

Host
Standardization bodies

**W3C**

WS-I is an open industry organization chartered to promote Web services interoperability across platforms, operating systems, and programming languages.

**OASIS**

OASIS is a not-for-profit, global consortium that drives the development, convergence and adoption of e-business standards.
### Standards

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Requirement</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messaging</td>
<td>Confidentiality and Integrity</td>
<td>WS-Security (XML DSig/Enc)</td>
</tr>
<tr>
<td></td>
<td>Authentication</td>
<td>SSL/TLS (HTTPS)</td>
</tr>
<tr>
<td></td>
<td>Privacy</td>
<td>SSL/TLS (X.509)</td>
</tr>
<tr>
<td></td>
<td>Accountability</td>
<td>XACML</td>
</tr>
<tr>
<td>Discovery</td>
<td>Registries</td>
<td>XrML</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RBAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EPAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XACML</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Auditing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UDDI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebXML</td>
</tr>
</tbody>
</table>
WS-* security Standards framework

- Security mgmt.
  - XKMS
  - WS-Trust
- Identity Mgmt.
  - WS-federation
  - Liberty
  - SAML
- Message security
  - WS Security
  - WS SecureConversation
- Reliable Messaging
  - WS ReliableMessaging
- Policy & Access Control
  - WS-Policy
  - XACML
  - SAML
- SOAP foundation
- XML security
  - XML Encryption
  - XML Signature
- Transport level security
  - SSL/TLS
- Network level security
  - IPSec
What is WS-Security?

- WS-Security enhances SOAP messaging to provide *quality of protection* through:
  - message integrity,
  - message confidentiality, and
  - single message authentication.

- These mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

- WS-Security also provides a general-purpose, extensible mechanism for associating security tokens with messages:
Security policies for Web Services

- The concept of Policy: Guiding principles and procedures

- Security policy might mean different things to different people:
  - Firewall filtering rules
  - Access control policy
  - Privacy policy
Standards for Web services policies

- WS-Policy

- XACML

- XACML profile for Web Services
WS-Policy

- **Main goal**: The WS-Policy and WS-PolicyAttachment aim to offer mechanisms to represent the capabilities and requirements of Web services as Policies.
XACML

- eXtensible Access Control Markup Language 2 (XACML) Version 2.0 OASIS Standard, 1 Feb 2005

- Status: approved OASIS Standard within the OASIS Access Control TC.
XACML Overview

- XACML is a general purpose access control policy language for managing access to resources.

- It describes both a policy language and an access control decision request/response language.

- Access control based on subject and object attributes.

- Consistent with and building upon SAML.
Security Assertion Markup Language (SAML)

- Developed by the OASIS XML-Based Security Services Technical Committee (SSTC)

- **Status**: SAML V2.0 OASIS Standard specification set was approved on 15 March 2005

- **Main goal**: authentication and authorization
The goal:
- promote interoperability between disparate authentication and authorization systems

How:
- defining an XML-based framework for communicating security and identity information (e.g., authentication, entitlements, and attribute) between computing entities
Secure Implementation Tools and Techniques

- **XML Parsers**
  - XML Parsers are the first component to process input to Web services
  - They must be robust
  - Large or specially formed XML documents can lead to DOS Attacks
Secure Implementation Tools and Techniques

- **Procedural Languages:**
  - C and C++
    - Less overhead, which is useful for embedded systems: J2EE and .NET frameworks take up hundreds of megabytes of hard disk space
    - Can directly interface with legacy applications developed in C or C++
    - Susceptibility to programming errors may require addition protections like XML Gateways or OS level restrictions

- **Java and .NET**
  - Widely considered to be more secure languages
  - Two of the most popular languages for developing Web services
  - Provide robust sandboxes (JVM and .NET Code Access Security)
  - Large number of third-party libraries available for Java and .NET Web services
Secure Implementation Tools and Techniques

- **Security Testing**
  - Functional testing of security mechanisms
    - Ensure that Web service security mechanisms work as required
  - Security-focused unit testing
    - Perform security testing on individual components of the Web service
  - Vulnerability assessments
    - Attempt to attack the Web service using known attack types
  - Web service code reviews and testing
    - Check the source code for vulnerabilities or security errors
    - Perform testing with unexpected or random input to find susceptibility to unknown attacks
Common Attacks against Web Services

- Attacks on Integrity: Parameter Tampering, XML Schema Poisoning
- DOS Attacks: Flooding Attacks, Send Oversized Payloads to XML Parsers, Buffer Overflow Exploits
- Command Injection: SQL Injection, XML Injection
- Malicious Code Attacks: Virus, Worms, Trojan Horse
Challenges for Secure Web Services

- Contracts and Negotiation
- Protection from Common Attacks
- End to End QoS and QoP
- Interoperability among competing Standards
- Methodologies for Secure Web Services
- Life Cycle Management
Conclusions

- Web Services based computing has benefits
- W3C and OASIS have made good progress in laying the foundation
- Several research problems need to be solved
  - QOS & QOP
  - Automatic Service Discovery
  - Availability and protection from DOS Attacks