XML, Web Services & SOA: Data Protection and Privacy Opportunities and Challenges in the Government Sector

Rich Salz
STSM, Senior Security Architect
IBM
Agenda

- XML and Web Services Impact on Security
- Security Underlies Government SOA Success
- Why SOA Security is a Concern
- Major Categories of SOA Security Functions
- Web Services Security and SOA
- WS-Trust, SAML, Access Control
- The Need for Hardware-based XML Security
- XML Hardware Encourages Interoperability
- IBM SOA Appliances Overview
- Summary
XML and Web Services can Impact Security

They help form the foundation of SOA, but bring new security obstacles:

- Scalability: XML is bandwidth, CPU and memory intensive
- Performance: some XML apps literally grind to a halt
- Privacy: connecting systems never before connected
- Data Protection: clear text over HTTP with no inherent security
- Integration: exposing Web services to legacy applications
- Standards are still in flux
- Financial, technical and organizational challenge
Government SOA

- IP-based network data flow
- Internal access moving to external access
- Federal Enterprise Architecture (FEA) composed of interrelated ‘reference models’
- eGov Initiatives built upon XML, Web services
  - Procurement, Supply Chain, etc.
  - Promote services re-use and consolidation
  - Increased integration and communication
- Cross-domain services, information, identity sharing
- DOD Net-Centricity transformation
Security Underlies Government’s SOA Success

- Shift to Message-Level Security
- SAML & Federation - eAuthentication & eAuthorization certificates
- COTS products that support standards
- DHS integration
- Netcentricity Phase II: Service-oriented Fusion
- Privacy, Integrity, ID management
- PKI
- Right information to right people in timely fashion
- Ubiquitous access vs. control, policy enforcement
**Why SOA Security is a Concern**

- Any new technology has new security implications
- XML and SOAP easily connect to backend systems
- For a business-centric SOA, the exposed systems are critical business systems
- Traditional packet-level security devices do not secure XML/SOAP
- New compliance and regulatory requirements
- In addition to application developers, many other parts of the organization need to be involved
Roles of Different Protocol Layers

SSL is not enough
- XML-level threats and XML-aware security
- securing stored or spooled messages
- multi-party transactions, multi-hop networks
Major categories of SOA Security Functions

- XML threat protection
  - Concerned with keeping out malicious XML
  - Sometimes called XML firewall or XML intrusion prevention
- Message confidentiality & tamper-protection
- Secure enablement
  - Concerned with allowing in only XML compliant with access policy
  - Example: access control policy enforcement
  - Some vendors may call this “trust management”
- Identity management
- Misc. web services management functions
  - Example: service level management
**XML/SOAP Firewall**

- Integrated multi-layer filters
  - IP-layer params (e.g., client IP address)
  - SSL params (e.g., client certificate)
  - Any part of HTTP header
  - XPath or XML configuration files for any part of SOAP header
  - XPath or XML configuration files on any part of XML payload
  - First-level filter select based on service, URL, etc.

- Easy “point and click” XPath Filtering
- Enable/Disable each SOAP method using WSDL wizard
- Can be applied at any point in message processing
Multiple Level of Defense for SOA

- First Level: XML Security Gateway for enhanced security, scalability, and simplicity
- Second level: Application server for additional processing
XML Threat Protection

- XML Entity Expansion and Recursion Attacks
- XML Document Size Attacks
- XML Document Width Attacks
- XML Document Depth Attacks
- XML Wellformedness-based Parser Attacks
- Jumbo Payloads
- Recursive Elements
- MegaTags – aka Jumbo Tag Names
- Public Key DoS
- XML Flood
- Resource Hijack
- Dictionary Attack
- Message Tampering
- Data Tampering
- Message Snooping
- XPath Injection
- SQL injection
- WSDL Enumeration
- Routing Detour
- Schema Poisoning
- Malicious Morphing
- Malicious Include – also called XML External Entity (XXE) Attack
- Memory Space Breach
- XML Encapsulation
- XML Virus
- Falsified Message
- Replay Attack
**XML/SOAP Data Validation**

- Raw XML and SOAP message inspection **(inbound and outbound)**
- XML well-formedness checks
- SOAP protocol checks
- XML Schema validation options:
  - Explicitly set XSD in validate step
  - Fetch “trusted” copy of XSD based on XSD self-declared by incoming XML document
  - Validate from WSDL for SOAP web services
- Streaming schema and well-formedness processing
  - Errors can be detected before the entire message is read in
- Business logic and other arbitrary validation
  - XSLT transformations to extract or validate business-level information contained in XML/SOAP payload
Enforcing Access Control

- High-speed Security Hardware access policy enforcement point
- Modular authentication/authorization architecture
  
  \[
  \begin{align*}
  x &= \text{extract-identity()} \\
  z &= \text{extract-resource()} \\
  zm &= \text{map-resource}(z) \\
  y &= \text{authenticate}(x); \text{if}\ (y = \text{null}) \text{reject} \\
  ym &= \text{map-credentials-attributes}(y) \\
  \text{allowed} &= \text{authorize}(ym, zm); \text{if}\ (!\text{allowed}) \text{reject} \\
  \text{audit-and-post-processing}();
  \end{align*}
  \]

- Identity examples include:
  - WS-Security user/pass token
  - SSL client certificate
  - SAML assertion
  - HTTP basic-auth
  - Proprietary SSO cookie/token

- Resource examples:
  - URL
  - SOAP method
Web Services and SOA Security


- WS-Security
- WS-Coordination
- WS-Transactions
- WSDL
- WS-Policy
- UDDI
- SOAP, SOAP Attachments
- XML, XML Infoset
- Transports

Quality of Service

Description and Discovery

OASIS Secure eXchange TC

WS-Security

- WS-Secure Conversation
- WS-Transactions
- WS-Trust
- WS-Federation
- WS-Authorization
- WS-Privacy

OASIS 1.0 WS-Security (framework)

- Kerberos profile
- X.509 profile
- REL profile
- Mobile profile
- Username profile
- SAML profile

SAML
Liberty
**What “supports SAML” can mean**

- SAML browser artifacts
  - Support for exchange of several interoperable token information via HTTP (without XML) for web single-sign-on
- Consume SAML assertions
  - Ability to accept a SAML in an incoming web service request or web service transaction, use it to enable access to some protect service
- Produce SAML assertions
  - Generating a SAML assertion based on AAA processing that took place for subsequent access control purposes
- Make SAML queries
  - Make web service calls to a SAML server for AAA decisions
- Accept SAML queries
  - Respond to authentication, authorization or audit request via web service protocol defined by SAML
**WS-Trust**

- Extends WS-* and WS-Security directly
- Security tokens:
  - Issue
  - Renew
  - Validate
- Trust relationships
  - Establish
  - Assess the presence of
  - Broker trust relationships

Figure courtesy of WS-Trust specification
The Need for Hardware Based XML Security

- Hardware XML Security Reduces Complexity
- Hardware XML Provides Hardened Security
- Hardware XML Security Delivers superior Performance
- Hardware XML Security Encourages Interoperability
**Hardware provides Hardened Security**

- **Accountability:**
  - OS upgrades
  - Security software upgrades
  - Hardware upgrades

- **Hardened OS**
  - Eliminate generic processes, daemons or listeners.

- **Hardware-based crypto Algorithms**
  - Prevent application developers from using weak crypto implementations

- **Separation of Security Policies from Applications**
XML Cryptography & Security Performance

- Crypto operations are resource-intensive
- Public-key crypto operations are very expensive
- Familiar example SSL
  - A couple RSA ops per connection, bulk encryption
  - Today, SSL hardware acceleration is well-accepted practice
- XML example: WS-Security based XML message
  - Signed header(s)
  - Public-key encrypted symmetric key
  - Encrypted payload sections
  - Signed payload sections
  - 10+ public-key ops per message is quite likely
- Multiple messages per connection
- XML processing also significant
**XML hardware encourages interoperability**

- Coupled to the other systems by Ethernet jack, not custom code

- Separation of concerns

- Network gear business model based on “out-of-the-box” interop

- Large software vendors focused on creating XML-enabled platforms
  - Functionality and development tools benefit
  - Interop is necessarily secondary, standards wars looming

- Network vendors architecturally unable to achieve “lock-in”

- Focused on a concrete set of challenges
  - XML security performance
  - Interoperability.
Interoperability promoted through Standards Bodies

- Interoperability is hard work, but much more likely
  - WSI promotes webServices Interoperability.
    - The WS-I testing tools are designed to help developers determine whether their Web services are conformant with Profile Guidelines.
  - “SOAP Specifications Assertions and Test Collection”
    - A SOAP 1.2 implementation that passes all of the tests specified in this document may claim to conform to the SOAP 1.2

- Baseline Standards have matured, for example:
  - SOAP 1.1 – May 2000
  - XML DSIG – Feb 2002
  - SAML 1.0 – November 2002
  - WS-Security – April 2002

- Integration with CA’s, policy stores, schema repositories, service repository registries

- Interoperability in a heterogeneous environment with application servers, in-house software, hardware devices from other vendors
**SOA Appliances Fit with FEA**

- **Performance Reference Model (PRM)**
  - Government-Wide, LOB-Specific Performance Measures & Outcomes

- **Business Reference Model (BRM)**
  - Lines of Business, Agencies, Customers, Partners

- **Service Component Reference Model (SRM)**
  - Service Domains, Service Types, Components, Access, Delivery Channels

- **Data and Information Reference Model (DRM)**
  - Business-Focused Data Standardization, Cross-Agency Exchange

- **Technical Reference Model (TRM)**
  - Technologies, Standards, Specifications, Component Framework

- Hardware approach provides price/performance & manageability

- Hardware security gateway enables higher security assurance for cross-agency exchange

- Hardware approach fits well within the Component Framework
Example of other SOA appliance use: XML Routers

- Content-based routing based on dynamic XPath tables
- SOAP protocol routing and load balancing
- Message enrichment via headers
- Publish-Subscribe based on content in messages
- Message duplication & relay
- QoS and QoP based on message content
- Routing and delivery independent of producers or consumers
Thank You