Web Services Security with SOAP Security Proxies

Gerald Brose, PhD
Technical Product Manager
Xtradyne Technologies AG

OMG Web Services Workshop USA
22 April 2003, Philadelphia
Web Services Security Risks

- Exposure of critical resources is a risk
  - integration = new access paths and data flows
- SOAP itself has *no* security mechanisms to mitigate risks
- existing security technology does not apply well
  - HTTP is not filtered by standard firewalls
  - SSL does not provide end-to-end security
Roadmap

- Web Services Security
  - Threats, Security services, Challenges
  - Protocol layers
- Web Services Security Standards
  - WS-Security
  - XML DSig, XML Encryption
  - SAML
- Web Services Security Proxies
  - Functionality, Deployment Scenarios
Security Threats

- **Attacks on messages**
  - read and record
    - espionage, privacy breaches
  - replay
    - sabotage, fraud
  - modify in transit
    - sabotage, fraud
  - redirect or drop
    - sabotage, fraud
- **Attacks on services**
  - unauthorized access
    - read, write, use
    - espionage, sabotage, fraud, theft
  - denial of service
Security Services that help

- Authentication
  - "where does this (part of a) message come from?"
- Authorization (access control)
  - "may this message pass?"
- Confidentiality
  - "who can read this (part of a) message?"
- Integrity
  - "has this (part of a) message been tampered with?"
- Audit
  - "what happened?"
- Administration
  - "how do I manage this?"

... but SOAP has none of this!
Web Services Security Challenges

- **Loose coupling**
  - Web Services are message-based
    - transport security sessions don't fit
- **HTTP transport**
  - SOAP messages pass firewalls uninspected
    - existing perimeter protections don't apply
- **Service composition**
  - a single message can traverse many intermediaries
    - who do you trust with what?
- **Document-based workflows**
  - different parts of a message
    - are processed by different processors
    - may need different access modes for different parties
Security and Protocol Layers

WS-Security
- XML DSig
- XML Encryption

SOAP

HTTPS
Sender
Intermediary
Receiver

point-to-point
end-to-end
point-to-point

SAML
X.509
Kerberos
WS-Security

- OASIS-Standard
  - Working Draft since 11/2002
- Message-level Security Model for SOAP
  - can embed a wide variety of existing technologies
  - end-to-end security with multiple trust domains
- Extensible security message header `<wsse:security>`
  - for security information in and about messages
- Security Token format
  - express claim(s) made by entities
  - text/binary, signed/unsigned, e.g. username or certificate
- Integrity, Authentication, Confidentiality
  - processing rules for XML Digital Signature and XML Encryption
- Common basis for future specifications
  - WS-Policy, WS-Trust, WS-Privacy, ...
XML Digital Signature

- W3C-Standard
  - "Recommendation" since 2/2002
- XML-Syntax for digital signatures
  - not just for XML content!
  - enveloped, enveloping, detached

Usage in WS-Security
- detached
  - Integrity protection for individual parts of a message (header and body)
- Authentication of security tokens
- Binding security tokens to messages
General Form of a Digital Signature

```xml
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
    (<Reference URI?>
     (<Transorms>)?
     <DigestMethod>
     <DigestValue>
     </Reference>)+
  </SignedInfo>
  <SignatureValue>
    (<KeyInfo>)?
    (<Object ID?>)*
  </SignatureValue>
</Signature>
```

- Information about the signed object
- References the signed object
- The actual signature
XML Encryption

- W3C-Standard
  - "Recommendation" since 12/2002
- XML syntax to represent encrypted data
  - not just encryption of XML content!
  - no new algorithms
- Usage in WS-Security:
  - protect confidentiality of individual parts of a message
    - header (e.g., session keys)
    - body
    - attachments
Security Assertion Markup Language (SAML)

- OASIS-Standard (1.0, since 5/2002)
- XML-based framework for the exchange of security information
  - *assertions* = *statements* by an *issuer* about a *subject*
    - *authentication assertion* - subject is authenticated
    - *authorization decision assertion* - subject is authorized
    - *attribute assertion* - subject has given attributes

- **SAML Protocol**
  - between *Policy Enforcement Points* (PEP) and *Policy Decision Points* (PDP)
  - defines request and response messages

- **Usage of SAML assertions in WS-S**
  - format for Security Tokens
    - Binding to WS-Security in progress ("SAML Token binding")
Standards in Concert

WS-S
XML DSig
SAML

SOAP:Env
SOAP:Header

<wsse:Security>
Signature
Assertion
</wsse:Security>

<SOAP:Body wsu:Id="x">
</SOAP:Body>

Copyright © 2003 XTRADYNE Technologies AG
How to deploy WS-Security?

- Secure endpoints: AppServer + client software
  - Drawbacks
    - integration may involve modifying software
    - management of multiple hosts and pieces of software
    - possible vendor-dependencies
- Secure gateways: *Web Services Security Proxies*
  - Advantages:
    - transparent integration into *existing* systems
      - separates application and security functionality
    - simpler, centralized administration
      - only the proxies need to be configured and managed
    - platform and vendor independency, interoperability
    - offloads processing (cryptography, etc.)
Web Services Security Proxies

- Transparent Proxy for Web Services
  - messages are sent to the proxy, inspected there, and forwarded

- Application-level Gateway
  - security in the application layer
    - proxy understands SOAP/HTTP and WS-Security
    - content inspection

- Deployed at both sender and receiver
  - outgoing SOAP messages are extended with WS-Security information
  - supports B2B through federated trust!
Web Services Security Proxies

![Diagram of Web Services Security Proxies]

- SOAP
- WS-Security
- HTTP
- HTTP(S)
- Intermediaries
- Trust Boundary
- WS-Security Proxy

Copyright © 2003 XTRADYNE Technologies AG
Security Services in the Proxy

sender side
- Authentication
- Insertion of WS-S headers
- Authorization (outgoing)
- Integrity
  - Verification and Signing
- Content Filtering
  - XML Schema checking
- Confidentiality
- Audit

receiver side
- Authentication (SAML or basic mechanism)
- Authorization
- Integrity
  - Verification and Signing
- Content Filtering
  - XML Schema checking
- Confidentiality
- Audit
## Deployment Scenarios

### Intranet

Web Services used internally for
- cross department service use
- application integration

WS-Security Proxy
- controls access to Web-Service resources from different departments
- Secure inter-application communication

### Internet

Deployment of new Web Services
- Application services for broad range of users
- UDDI registered services

WS-Security Proxy
- allows broad service access
- provides authentication and authorization services

### Federated Extranet

Web Services used to integrate applications and services with
- trading partner
- branch offices

WS-Security Proxy
- Federated Trust eliminates duplication of policy and user information
Internet Scenario

Original SOAP Message

Client application

SOAP Enabling Platform

Legacy application

SOAP

Internet

Security Proxy

DMZ

Protected/Verified SOAP Message

SOAP

WS-Security

SAML

Web Service

SOAP

WS-Security

SAML
Federated Extranet Scenario

- Original SOAP Message
  - Client application
  - SOAP Enabling Platform
    - SOAP
  - Legacy application

- Protected SOAP Message
  - Security Proxy
    - WS-Security
    - SAML
  - Internet
  - DMZ
  - Web Service

- Protected/Verified SOAP Message
  - Security Proxy
    - WS-Security
    - SAML
  - SOAP
  - DMZ
  - Web Service

- WS-DBC
- Policy Server
- Policy DB
- Web Service
- Admin Console
- SOAP
- IIOP/SSL
- (LDAP)
Summary

- Web Services need
  - suitable message-based security models
  - standards for interoperability
- Emerging security standards have strong industry support
  - consortia, vendors, products
- WS-Security Proxies as security solution
  - platform-neutral standards support
  - comprehensive security functionality for Web Services at the application layer
  - transparent integration without software modifications ("pluggable")
  - ideal support for B2B scenarios