XML Security Gateway Evaluation Criteria Project Update

Gunnar Peterson, OWASP XSGEC Project Lead
Managing Principal, Arctec Group
gunnar@arctecgroup.net

The OWASP Foundation
http://www.owasp.org/
About Arctec Group

- Best in class enterprise architecture consulting provider focused on enterprise, software, and security architecture
- Client list includes numerous global 500 companies, world’s largest electronic financial exchanges, emerging startups and Dept. Homeland Security
- Headquarters: IDS Center, Minneapolis, MN; Clientele: global
- Web: www.arctecgroup.net
About the speaker

Gunnar Peterson
Managing Principal, Arctec Group
Editor Build Security In software security column for IEEE Security & Privacy Journal (www.computer.org/security)
Primary and contributing author for DHS/CERT Build Security In portal on Web Services security, Identity, and Risk management (https://buildsecurityin.us-cert.gov/daisy/bsi/home.html)


Associate editor Information Security Bulletin (www.chi-publishing.com)
Contributor Web Application Firewall Evaluation Criteria (http://www.webappsec.org/projects/wafec/)

Blog: (http://1raindrop.typepad.com)

Slides/presentations (http://www.arctecgroup.net/articles.htm)
OWASP XML Security Gateway Evaluation Criteria Project (XSGEC)

 Goals:

- Defines an open standard for evaluating XML Security Gateways such as those used to protect and provide security services for Web services applications
- Add clarity to the process of assessing the XML Security Gateway strengths and weaknesses
- Enlighten the community as to the utility of XML Security Gateways to deliver security services for distributed systems.

Team: Mix of industry professionals, vendors, and consultants
XSGEC Guiding Principles

- Define evaluation criteria supporting a transparent, level playing field for XML Security Gateway solutions to define their solution's key value proposition
- Where practical, attempt to standardize nomenclature and metrics
- Educate the community on the design considerations for XML security
XML Security Gateway Pattern

- Context: The primary goal of Web services is to solve interoperability and integration problems. Web services traverse multiple technologies and runtimes.

- Problem: Web service requesters and providers do not agree upon binary runtimes like J2EE, instead they agree upon service contracts, message exchange patterns, and schema. Service and message level authentication, authorization, and auditing services for Web services are not delivered by a single container, rather these services must span technical and organizational boundaries.
XML Security Gateway Pattern

Solution: Use a XML Security Gateway to provide decentralized security services for Web services.
SOAP in the clear

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getCustomerDetails xmlns="http://servicehost"/>
    <name>Joe Smith</name>
    <password>hard2guess</password>
    <customernumber>1234</customernumber>
  </soap:Body>
</soap:Envelope>
Authentication Services
- Mutual SSL
- HTTP Basic Authentication
- HTTP Digest Authentication
- WS-Security Username Token
- WS-Security X.509 Certificate
- WS-Security: Kerberos Token
- SAML Authentication assertion
Let’s authenticate the message

```xml
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security
xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
      <wsse:UsernameToken
xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility"
wsu:Id="Id-00000112932fa600-0000000000000003">
        <wsse:Username>joesmith</wsse:Username>

        <wsse:Password
Type="wsse:PasswordText">hard2guess</wsse:Password>

        <wsu:Created>2007-05-16T04:40:12Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>

  <soap:Body>
    <ns0:getCustomerDetails xmlns:ns0="http://servicehost"/>
    <customernumber>1234</customernumber>
  ...
```
Can we make it stronger?

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
      <wsse:UsernameToken
        xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility" wsu:Id="Id-00000112933021b1-0000000000000004">
        <wsse:Username>joesmith</wsse:Username>
        <wsse:Nonce EncodingType="utf-8">2rxMB18VSv3jctE9Nr+xKRWpK0VZu8sp7wg527Fr+7U=</wsse:Nonce>
        <wsse:Password Type="wsse:PasswordDigest">mkCDa1QjlnGA32+1L2ywCp4oMT8=</wsse:Password>
        <wsu:Created>2007-05-16T04:40:44Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <ns0:getCustomerDetails xmlns:ns0="http://servicehost"/>
    <customernumber>1234</customernumber>
  </soap:Body>
</soap:Envelope>
Can we make it stronger?

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
      <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="Id-000001129332ad32-0000000000000000">
        <dsig:SignedInfo>
          <dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        </dsig:SignedInfo>
        <dsig:SignatureValue>dZdbQbIysLPvzuS5xsf57nUMB/M=</dsig:SignatureValue>
      </dsig:Signature>
    </wsse:Security>
  </soap:Header>
  <ns0:getCustomerDetails xmlns:ns0="http://servicehost"/>
</soap:Envelope>
```
XSGEC Authentication

- Evaluate XSG’s ability to
  - Authenticate service requesters
  - Assert security tokens and other authentication primitives to service providers
What about the message content?

XML Messages can contain a number of nasty things...

- Injection attacks
  - SQL Injection, Xpath Injection, Xquery Injection
- XML Denial of Service (XDoS)
  - Using XML as an attack vector
  - Jumbo payloads
  - Recursion
- Virus in SOAP attachments
XSG Validation Services

- Schema validation based on hardened schemas
  
  ```xml
  <xs:simpleType name="Zipcode">
    <xs:restriction base="xs:string">
      <xs:pattern value="([0-9]{5})-([0-9]{4})" />
    </xs:restriction>
  </xs:simpleType>
  ```

- Semantic validation based on white list or blacklist
  - Regex

- Virus scanning

- XDoS Countermeasures
  - Example: min/max message size
XSGEC Content Validation

■ Evaluate XSG’s ability to enable
  ▸ Schema validation
  ▸ Semantic validation
  ▸ XDoS protection
  ▸ Virus scanning
Service Requester
- Implementation
- Mapping
- Interface

XML Security Gateway
- Requester Interface
- Mapping
- Implementation
- Provider Interface

Service Provider
- Implementation
- Mapping
- Interface

Validate Input

Sign Content Policy Applied
XSG Sign Request

                      wsu:Id="Id-000001129356fac9-000000000000000e">
    <wsse:Username>XSG</wsse:Username>

    <wsse:Nonce EncodingType="utf-8">hFOW+PM/JIwKVQUz11Xt/r1EE73Wx1SPyqAfijguG
    Mk=</wsse:Nonce>

    <wsse:Password Type="wsse:PasswordDigest">7jwftIDmLZjBSN5zopmyEd4iY6w=</wsse:Password>
    <wsu:Created>2007-05-16T05:23:10Z</wsu:Created>
  </wsse:UsernameToken>
</wsse:Security>
</soap:Header>
Sign Content - Policy Applied

   <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
   AssertionID="Id-000001129354af1c-0000000000000002" IssueInstant="2007-05-16T05:20:39Z" Issuer="CN=Test,OU=Unknown" MajorVersion="1" MinorVersion="1">
      <saml:AuthorizationDecisionStatement Decision="Permit"
      Resource="http://host/service">
         <saml:Subject>
            <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">Test</saml:NameIdentifier>
         </saml:Subject>
      </saml:AuthorizationDecisionStatement>
   </saml:Assertion>
</wsse:Security>
</soap:Header>
<soap:Body>
<ns0:getCustomerDetails xmlns:ns0="http://servicehost"/>
<customernumber>1234</customernumber>
XSGEC Authorization Support

- Evaluate XSG’s ability to
  - Assert that a specific policy has been applied by a XSG at a certain time for a given request, subject, condition, and action
Encrypt for Remote Hosts

Service Requester
- Implementation
- Mapping
- Interface

XML Security Gateway
- Requester Interface
- Mapping
- Implementation
- Mapping
- Provider Interface

Service Provider
- Implementation
- Mapping
- Interface

Authenticate

Encrypt
XML Encryption

<soap:Body>
<ns0:getCustomerDetails xmlns:ns0="http://servicehost"/>

<enc:EncryptedData..>
  <enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"/>

  <enc:CipherData>
  <enc:CipherValue>EjADnNmGlVK9wTiG+La+uHaDthMnSsI\n\n6CXvO1I1DYVT/M1asHgM+
  </enc:CipherData>
  <enc:ReferenceList>
  <enc:DataReference URI="#Id-000001129355dbad-0000000000000009"/>
  </enc:ReferenceList>
  <enc:CarriedKeyName>Id-000001129355dbad-0000000000000008</enc:CarriedKeyName>

  <wsu:Timestamp xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility"
wsu:Id="Id-000001129355dbb6-000000000000000d">
  <wsu:Created>2007-05-16T05:21:56Z</wsu:Created>
</wsu:Timestamp>
</enc:EncryptedData>
XSGEC

■ Evaluate XSG’s ability to
  ▸ Protect/encrypt/sign outbound messages
STS - Attribute & Identity Mapping
XSGEC

Evaluate XSG’s ability to:

- Map inbound/outbound security tokens
- Map inbound/outbound attributes
XSG Metrics

- Asset metrics
  - Transactional throughput volume and performance
- Threat metrics
  - Malicious requests/responses
- Vulnerability metrics
  - Policy violations
XSGEC

Evaluate XSG’s ability to:

- Provide historical, predictive, and real time metrics for
  - Assets
  - Threats
  - Vulnerabilities
Understand key architecture tradeoffs & design considerations
XSGEC

- Interested in participating? More information: