NIST’s Guide to Secure Web Services

Presented by
Gaspar Modelo-Howard
and Ratsameetip Wita

Secure and Dependable Web Services

Goals

- Give a broad overview of field (secure and dependable web services)
- Provide challenges and open problems in research

Outline

- Introduction to Web Services
- Web Security Dimensions
- Attacks to Web Services
- Web Service Security Functions
- Challenge and Open Problems
Introduction to Web Services

- WS: Software system designed to support interoperable machine to machine interaction over a network [W3C]
  - Ex: Web APIs accessed over a network and executed remotely

Introduction to Web Services

- Make a collection of software services accessible via standardized protocols whose functionality can be automatically discovered and integrated into applications (loosely coupled)
  - Allows WS to dynamically bind to other WS at run-time, depending on needs of user or app
  - Allows data and apps to interact (no human intervention) through dynamic and ad hoc connections
A Few More Acronyms to Your Vocabulary

- OWL-S: Ontology Web Language for Services
- SOA: Service Oriented Architecture
- SOAP: Service Oriented Architecture Protocol
- SAML: Security Assertion Markup Language
- UDDI: Universal Description, Discovery and Integration
- WSDL: Web Services Description Language
- XACML: eXtensible Access Control Markup Language

Introduction to Web Services

- Components of a SOA application
  - Discovery
    - UDDI / WSDL
  - Messaging
  - Portals
  - Roles
  - Coordination

Figure 2-1. Web Service Discovery Example
Introduction to Web Services

- Components of a SOA application
  - Discovery
  - Messaging
    - SOAP (XML)
  - Portals
  - Roles
  - Coordination

Figure 2-2. Web Service Messaging Example

Introduction to Web Services

- Components of a SOA application
  - Discovery
  - Messaging
  - Portals
  - Roles
  - Coordination

Figure 2-3. Example Portal Interface
Introduction to Web Services

- Components of a SOA application
  - Discovery
  - Messaging
  - Portals
  - Roles
    - Requester
    - Intermediary
    - Provider
  - Coordination

Introduction to Web Services

- Components of a SOA application
  - Discovery
  - Messaging
  - Portals
  - Roles
  - Coordination
    - Orchestration
    - Choreography
Introduction to Web Services

- Components of a SOA application
  - Discovery
  - Messaging
  - Portals
  - Roles
  - Coordination
    - Orchestration
    - Choreography

Outline

- Introduction to Web Services
- Web Security Dimensions
- Attacks to Web Services
- Web Service Security Functions
- Challenge and Open Problems
WS Security Dimensions

- Secure Messaging
  - SOAP was not designed with security in mind
  - Possible approaches: HTTP over SSL, XML Encryption and XML Signature, WS-Security

- Protecting Resources
  - WS are intended to be accessible only to authorized requesters
  - Protection involves more than just access control (disruption, mitm, eavesdropping, impersonating)

WS Security Dimensions

- Negotiation of Contracts
  - WS should automatically negotiate and agree upon contracts (ebXML, WSDL)
  - No standards that support enforcement of implicit contracts (WSDL)
  - QoP: Only some support for negotiating security requirements
  - Area of research: Semantic Web Services Architecture (SWSA)
WS Security Dimensions

- Trust Relationships
  - Currently limited to trust of the service identity
  - Architecture models:
    - Pairwise trust
      - Each WS is provided the sec info of all other WS
    - Brokered trust
      - Uses TTP, WS should be designed with this in mind
    - Federated trust
      - WS from different organizations can interact
    - Perimeter defense
      - XML gateways placed between providers and requesters

- Some Pitfalls
  - XML Encryption / XML Signature: no std for informing recipients how were applied to message

<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, SSL/TLS</td>
</tr>
<tr>
<td></td>
<td>Authentication</td>
<td>WS-Security Tokens, SSL/TLS, X.509 Certificates</td>
</tr>
<tr>
<td>Resource Protection</td>
<td>Authorization</td>
<td>SAML, XACML, XMMS, IRAC, ABAC</td>
</tr>
<tr>
<td></td>
<td>Priority</td>
<td>SAML, XACML</td>
</tr>
<tr>
<td></td>
<td>Accountability</td>
<td>None</td>
</tr>
<tr>
<td>Negotiation of Contracts</td>
<td>Regulators</td>
<td>ebXML</td>
</tr>
<tr>
<td></td>
<td>Semantic Discovery</td>
<td>SWSA</td>
</tr>
<tr>
<td></td>
<td>Business Contracts</td>
<td>ebXML</td>
</tr>
<tr>
<td>Trust Management</td>
<td>Establishment</td>
<td>WS-Trust, SAML, X.509</td>
</tr>
<tr>
<td></td>
<td>Trust Propagation</td>
<td>SAML, WS-Trust</td>
</tr>
<tr>
<td></td>
<td>Federation</td>
<td>WS-Federation, Liberty IDF, OASIS, Shibboleth</td>
</tr>
<tr>
<td>Security Properties</td>
<td>Policy</td>
<td>WS-Policy</td>
</tr>
<tr>
<td></td>
<td>Security Policy</td>
<td>WS-Security Policy</td>
</tr>
<tr>
<td></td>
<td>Availability</td>
<td>WS-Reliability, WS-Reliability</td>
</tr>
</tbody>
</table>
Web Services Security Standards

Figure 2.8 - Web Services Security Standards: Notional Reference Model

Web Services Example
Attacks to Web Services

- Reconnaissance Attacks
- Code Templates
- Forceful Browsing Attack
- Directory Traversal Attack
- WSDL Scanning
- Registry Disclosure Attack
- Privilege Escalation Attack
- Dictionary Attack

Table 2-2. Threats Addressed by Current Web Service Standards

<table>
<thead>
<tr>
<th></th>
<th>Message Alteration</th>
<th>Loss of Confidentiality</th>
<th>Phished Message</th>
<th>Man-in-the-Middle</th>
<th>Physical Spoofing</th>
<th>Forged Claims</th>
<th>Replay of Message Pads</th>
<th>Replay of Message</th>
<th>Denial of Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Encryption</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>XML Signature</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WS-Security Tokens</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WS-Addressing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSL/TLS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSL/TLS with client certificates</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP Authentication</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Introduction to Web Services
- Web Security Dimensions
- Attacks to Web Services
- Web Service Security Functions
- Challenge and Open Problems

Web Service Security Functions

- Service-to-Service Authentication
- Identity Management
- Establishing Trust between Services
- Describing Web Services Policies (WS-Policy)
- Distributed Authorization and Access Management
- Confidentiality and Integrity of Service to Service Interchanges
- Accountability End-to-end throughout a Service Chain
- Availability of WS
- Securing the Discovery Service
Identity Management

- An Identity Management System (IDMS) is responsible for
  - Verifying identities of entities
  - Registering them
  - Issuing them digital identifiers

Three major identity architectures for WS

- Isolated identity management
  - Credential and identity providers are merged
  - Service must know for all requesters (scalability issue)

- Federated identity management
  - Group of providers agrees to recognize user ids from one another
  - More feasible in a single enterprise-wide SOA

- Centralized identity management
  - Rely on single TTP to provide credentials/identifiers
  - Single point of failure
Establishing Trust between Services

- Trust relationships need to be established between remote WSs for SAML or WS-Security to be useful on a large scale
  - Trust models like Kerberos have worked well for a single organization
- Pairwise trust circle
  - Each entity that is authorized to communicate with another must share its key information (unscalable)
- Brokered trust model
  - TTP is used to exchange key information between services to communicate
- Community trust model
  - Relies on an external PKI for establishing trust

Establishing Trust between Services

- Practical approaches for federation of trust
  - Liberty Alliance
    - Provides both Web app and WS federation using SAML to perform trust brokering
    - Suitable for businesses and governments
  - WS-Federation
    - Allows different security realms to federate by defining trust brokers, who will validate security tokens used between WS using WS-Trust (tokens)
Describing Web Services Policies (WS-Policy)

- Extension to WSDL, allows to express capabilities, requirements and characteristics of WS
  - WSDL is limited to describing what is included in the message itself
- WS-Policy requirement types
  - On the wire (WS-Sec encryption, signature)
  - Abstract (QoS, privacy)
- WS-Policy expression contains a set of policy alternatives encompassing sets of assertions
- Policy expressions are external to metadata stored in UDDI and WSDL, need distribution mechanism
  - WS-MetadataExchange or WS-PolicyAttachment

Describing Web Services Policies (WS-Policy)

- Specifications defining WS-Policy assertions
  - WS-SecurityPolicy defines assertions to specify integrity, confidentiality, and information about security tokens
  - WS-RM Policy defines assertions that can be used to specify how a WS uses WS-Reliable Messaging
  - WS-Addressing WSDL Binding defines elements that can be used within a WSDL descriptor to specify the use of WS-Addressing
Describing Web Services Policies (WS-Policy)

```xml
<Policy>
  <ws:Policy>
    <ws:AbstractPolicyReference />
  </ws:Policy>
</Policy>
```

Figure 3.2. Sample WS-Policy Expression

```xml
<Policy>
  <ws:Policy>
    <ws:AbstractPolicyReference />
  </ws:Policy>
</Policy>
```

Figure 3.3. Sample WS-ReliableMessaging Policy Expression

Describing Web Services Policies (WS-Policy)

```xml
<Policy>
  <ws:Policy>
    <ws:AbstractPolicyReference />
  </ws:Policy>
</Policy>
```

Figure 3.4. Sample WS-Policy Expression Using ExactlyOne
Availability of WS

- Availability is intended to ensure that QoS and reliability are maintained even under intentional attempts to compromise the WS operation (DoS)
  - Recognize and react to DoS patterns
  - Constrain and isolate the DoS attack
  - Recover and resume secure operation after DoS
- Necessary to include redundancy, error handling capabilities and defensive techniques
- Most common accidental threats
  - Service recursion
  - Service deadlock

Availability of WS

- Failover
  - UDDI supports listing multiple URIs for each WS
  - Makes the UDDI registry the point of failure
  - UDDI supports replication
- QoS
  - Expected level of performance
  - Most WS do not provide guarantees for QoS
  - In distributed deadlock, a WS may lead entire choreography stalling
Outline

- Introduction to Web Services
- Web Security Dimensions
- Attacks to Web Services
- Web Service Security Functions
- Challenge and Open Problems

Challenge and Open Problems

- Security remains a major challenge, in the presence of dynamic composition and heterogeneity in large, autonomous and untrusted environments
- Shift from old paradigm of Alice and Bob
Challenge and Open Problems

- Service description, automatic service discovery, and QoS
  - Make WS simpler? Feasible?
- Secure issuance of credentials
- Repudiation of transactions
  - Few logging implementations that can be used across an entire SOA
- Relationship between contracts and federated identity management

Challenge and Open Problems

- DoS attacks
  - Protection from DoS attacks that exploit vulnerabilities unique to WS (discovery service)
- Spread of malware
- Compromised services
  - Functional integrity of WS that requires the establishment of trust between services on transaction-by-transaction basis
  - Confidentiality and integrity of data transmitted via WS protocols in service-to-service transaction
- Exploitation of covert channels
NIST’s Guide to Secure Web Services

Thanks!
Happy Halloween