The Role of Identity Enabled Web Services in Cloud Computing

April 20, 2009

Patrick Harding
CTO

PingIdentity™
Agenda

- Web Services and the Cloud
- Identity Enabled Web Services
- Some Use Cases and Case Studies
- Questions
Web Services and The Cloud
Applications Rely on Identity to Meet Fundamental Business Requirements

- As Distributed Computing Evolves, Users are Pushed Further and Further Away from Applications
- How Do You Securely Control Access in an Increasingly Distributed World While Supporting Emerging Technologies like Web Services?
Web Services Security Soup

WS-Security

Mobile

Customers

Business Partners

WS-Trust

REST

X.509

WS-SecurityPolicy

SAML

Username/Password

Customer Service Reps

Gadgets/Widgets

3rd Party Mashups

SaaS

Web Service

Web Portal

SOAP

PKI

Kerberos

WS-SecureConversation

Username/Password

SaaS

WS-SecurityPolicy

SSL

XMLSignature

oAuth
Web Services and the Cloud

• Cloud Computing & SaaS are changing the requirements for addressing how to secure web services

• Cloud applications & SaaS will need to securely access on-premise data
  - Google Enterprise Developer Platform
  - SaaS Integration

• On premise applications will need to access SaaS API’s
  - Salesforce.com API’s are heavily used - 2 Billion API calls a month
Security & Integration

Figure 3: Buyers' Concerns With Adopting SaaS

“Why aren’t you interested in software-as-a-service?”

- Total cost concerns: 37%
- Security concerns: 30%
- We can't find the specific application we need: 25%
- Integration issues: 25%
- Lack of customization: 21%
- Application performance (e.g., downtime, speed): 20%
- Complicated pricing models: 16%
- We're locked in with our current vendor: 14%
- Other reason: 13%

Base: 352 US packaged application software decision-makers that are not interested in SaaS
Source: Enterprise And SMB Software Survey, North America And Europe, Q4 2008
Source: Forrester Research, Inc.
Identity Enabled Web Services
Securing Web Services Today

- Web Services Principles
  - Standards-based, Loosely Coupled, Scalable Applications
- Most Current Security Approaches Violate Web Service Principles
  - Customized, Tightly Coupled, Not Scalable, No Delegation
  - e.g. Mutually Authenticated TLS, User Identity in SOAP Body
    - Each Web Services app must know in advance where identity information is located in SOAP body; TLS session is point-to-point
WS-Security Enables a New Standards-based Trust Model

- Encrypted, Signed, Standard WS-Security SOAP Message
- WS-Security SOAP Header Includes Standard WS-Trust Security Token with Identity Information

Key Question: How To Create Security Tokens?
WS-Trust Basics

- WS-Trust is an OASIS standard and an extension of WS-Security
- WS-Trust enables security token exchange
- A WS-Trust STS Issues and Validates Security Tokens
  - Kerberos, UserName/Password, X.509, SAML, ....
Identity Enabled Web Services

- WS-Security SAML Token Profile
- Enables Delegation
- STS Implements Federated Identity Concepts
  - Attribute Contracts
  - Session Integration
  - Attribute Retrieval
  - Subject, Attribute and Role Mapping
Centralized Security Token Processing

- Every Web Service Client and Provider Processes Security Tokens
- STS Handles All Security Token Processing

- Gets Identity-Related Security and Crypto Code Out of Applications
- Centralized Administration, Auditing
- Requestor and Recipient Based Policy and Behavior
- Enables Web Services SSO with both Web Clients and Rich Clients
- Supports Client-Side, Provider-Side or Both
What about OAuth?

- Open standard to secure REST based web service API’s between SaaS/Web 2.0 applications

- Driven by SaaS/Web 2.0 community

- OAuth handles delegated web service authentication
  - Secure API authentication
  - Secure access to web service data API’s
  - Generally with explicit user consent
‘Two Legged’ oAuth

1. Consumer Key and Secret

2. Request Resource

3. Call API with OAuth Signature

oAuth Service Provider (Enterprise)

API

STS

OAuth Consumer (Google EDP)

Browser

Copyright (c) 2009 Ping Identity Corporation
'Three Legged' OAuth

1. Consumer Key and Secret

2. Request Resource

3. Get Unauthorized Request Token

4. User Redirect with Unauthorized Request Token

5. User Authenticates & Handles User Consent

6. User Redirect back with Authorized Request Token

7. Exchange Authorized Token for Access Token

8. Call API with OAuth Signature & Access Token

9. Display Resource

10. Display Resource
Use Cases & Case Studies
Web Services Use Cases

- Portal Initiated Web Services
  - Browser based app requires user specific data from internal or external web services

- Security for Web Service Providers and SaaS API’s
  - Web service providers require authenticated user to authorize access to user specific data

- Rich Desktop Clients
  - Desktop client applications SSO to web service providers inside or outside of security domain
Large Electronics Retailer

1. Browser
2. REST API
3. SOAP API
4. STS
5. Customer DB

3rd Party Shopping Applications

Partner

oAuth

WS-*
Recommendations

- Leverage SSL for Confidentiality
- Use SAML Tokens to Identity Enable SOAP Web Services
- Leverage a WS-Trust STS to Centralize SAML Token & Federation Processing
- Consider OAuth to secure your REST API’s
- ... and check out the new PingFederate 6.0 up the back
Questions