Keeping access control while moving to the cloud

Presented by Zdenek Nejedly
Computing & Communications Services
University of Guelph

Required reading webcomic: https://xkcd.com/792/
Keeping access control while moving to the cloud

Presented by Zdenek Nejedly
Computing & Communications Services
University of Guelph
Objectives

• Intro: University of Guelph mail migration

• Review: Access Management in the Cloud

• Conclusion: Solutions and Lessons Learned
University of Guelph mail migration

Can Access management help?
Migration project highlights

- Migrating 36k undergraduate students
- Production Sep 1, 2014
- Expanding from one to two mail systems
Migration project challenges

• User: two mail systems - am I on Google or Zimbra? Or both?

• University: policy confirmation before authorizing access to the service - how can we serve it to the users?
Access Management technologies

for the cloud services
• Do you provide Web Access Management on your campus?

• Do you provide authentication for cloud services? How?
  • Shibboleth? CAS? ADFS?
  • Other SAML 2 or non-SAML?
  • Custom SSO?
Why Web Access Management?

Functions:
• authN, authZ, SSO, attrs, audit

Benefits:
• Security: secured credentials
  • Password Reuse xkcd.com/792
• User experience: single identity, SSO
• Service Providers: ↓ friction - ↑ retention
• Identity providers: lower management cost
Cloud authentication: the early years

- SSO mostly as a custom solution
- Secret token exchange between the parties
- Individual solutions ➔ high cost
Cloud authentication: the protocols

- Gartner (2013) “…Gartner estimates a penetration well over 50% worldwide for SAML-based federations.”
What do I need to know?

Identity Provider (IdP)
Relying Party (RP)
Asserting Party (AP)

Role Based Access Control (RBAC)
Attribute Based Access Control (ABAC)
Security Assertion Markup Language (SAML)

Claims Consumer (CC)

One Time Password (OTP)
JSON Web Token (JWT)
Claims Provider (CP)
• **HTTP & HTTPS**
  • HTTP - application protocol (RFC 2616)
  • Stateless

• **GET & POST**
  • methods in HTTP
  • GET: resource retrieval, preserved in redirects
  • POST: sends data to the server in the body, may be lost in redirects

---

GET [http://example.com/stocks.cgi?name=IBM](http://example.com/stocks.cgi?name=IBM) HTTP/1.1

POST [https://example.com/authenticate](https://example.com/authenticate) HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
username=jane&password=w0rld2u

HTTP/1.1 302 Found
Location: [http://example.org/secure/docs/](http://example.org/secure/docs/)
**Extensible Markup Language**

**XML & JSON**
- free open standards

```xml
<person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <isAnalyst>true</isAnalyst>
  <phoneNumbers>
    <phone type="home">123 123-1234</phone>
    <phone type="cell">123 123-9999</phone>
  </phoneNumbers>
</person>
```

```json
{
  "firstName": "John",
  "lastName": "Smith",
  "isAnalyst": true,
  "phone": [
    { "type": "home", "number": "123 123-1234" },
    { "type": "fax", "number": "123 123-9999" }
  ]
}
```
Communication Protocol

SOAP & REST

Architectural design style

Tech Primer

- SAML & OAuth
- SOAP & REST
- XML & JSON
- GET & POST
- HTTP & HTTPS
Example of a SOAP fault message (http://www.w3.org/TR/soap12-part1/#faultcodes)

```xml
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
               xmlns:m="http://www.example.org/timeouts"
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env:Sender</env:Value>
        <env:Subcode>
          <env:Value>m:MessageTimeout</env:Value>
        </env:Subcode>
      </env:Code>
      <env:Reason>
        <env:Text xml:lang="en">Sender Timeout</env:Text>
      </env:Reason>
      <env:Detail>
        <m:MaxTime>P5M</m:MaxTime>
      </env:Detail>
    </env:Fault>
  </env:Body>
</env:Envelope>
```
REST (Roy Fielding 2000)
SAML 2.0 & OAuth 2.0

AuthN, authZ, attrs

Intended for Authorization

Web Browser SSO Profile

Server-side Web App
**SAML Authentication Flow for Google Apps** (Web Browser SSO Profile)

1) Browser requests Gmail content
2) Browser redirected to IdP with AuthnRequest
3) IdP identifies the user
4) Browser posts Response to Google with NameID
5) Google returns Gmail content
OAuth 2 Authorization flow (Server Side Web App profile)

1) Browser accesses Claim Consumer (CC)
2) Browser redirected to the Authorization Server (AS)
3) User authenticates, AS issues Authorization Code
4) Browser redirected to CC with
5) CC posts to AS
6) CC receives JSON response with Access Token
7) CC makes an API call to the API Provider with Access Token
More on OAuth 2.0 and OpenID Connect

• Talk by Ryan Boyd
  http://www.youtube.com/watch?v=YLHyeSuBspI

Getting started with OAuth 2.0
O’Reilly (2012)
Solutions, lessons learned

and the next steps
Challenge: where is my mail?

Staff, faculty, grads

Zimbra

Multiple roles?
Transient entitlements?

Undergrads

Gmail
Solution: Single access point

Mail SSO Middleware determines the correct mail system and routes the user accordingly.
Challenge: can we add a business process into the authN flow?

Default Google Apps
SAML2 AuthN Flow

1. User
2. UofG Identity Provider
3. Service Provider (Google)
4. User’s Gmail content returned
5. User's Gmail content returned
Solution: insert middleware

User confirms the Policies served by the Mail SSO Middleware (2a-2c)

1. Service Provider (Google)
2a. User’s Gmail content returned
2b. UofG Identity Provider
3. Mail SSO Middleware with the Policy engine
4. 2c. University of Guelph - Computing & Communications Services - www.uoguelph.ca/ccs
Mail SSO Middleware

Session Request for Gmail

Google Apps

SAML2 AuthN Request

SAML2 AuthN Response

UofG Shibboleth

OAM AuthN Request

OAM User ID and Attrs

OAM User Identity

UofG Oracle Access Manager

AuthN Request

OAM AuthN Request

Mail SSO Middleware

Session Request for Zimbra

Zimbra

Session Request for either Gmail or Zimbra

Mail SSO Middleware
Availability expectations for WAM?

- Clustering?
- Standby infrastructure?
Next steps - opportunities

- Weak points?
- Efficiency?

Build the policy module into the Access Manager authentication
Takeaway points

With Access Management we can:

• create a single access point for both email systems

• build a policy confirmation even into proprietary services

With increasing dependencies comes increasing requirement on high availability.

And remember - don’t reuse your password
Acknowledgements

Universities already on Google Apps - Thank you for sharing your experience with us.

University of Guelph Gryph Mail SSO team: Fazil, Hugh, Jill, Leo, Matt, Paul, Rob, Saveena, and Zdenek
External identities

Predicts 2014: Identity and Access Management (Gartner): “..by 2020 60% identities interacting with the enterprise will come from external IdPs (up from 10% today)…”

Are you using (or plan to) social identities on your campus?