Approaches to Simplify Server Authentication

Frederick Hirsch - Nokia
Hubert A. Le Van Gong – Sun microsystems

W3C Workshop on Transparency and Usability of Web Authentication
15/16 March 2006 — New York City, USA
Motivation

- No effective authentication of Service Providers to clients today
- We describe two possible approaches:
  - Shared secret between parties (the service provider & client),
  - Simplified Server Authentication (SSA) – SSO for service providers.
Shared Secret Approach

**Concept** - The SP registers itself to the client which creates an “account” and generates a shared secret to be presented by the SP.

**Pros** - No 3rd party is introduced between the SP and the client

**Issues**
- Requires standardization of format and sharing of secret
- Heavy lifting done on the client
  - Plugin approach is possible (albeit difficult)
- Potential scalability issue ($N^2$)
Simplified Server Authentication

Concept

- Focus on relatively simple browser client
- Applying the SSO concept and infrastructure to support Service Provider authentication
- Not discussing web service client
  - This offers additional solutions
    - e.g. Require SP to be authenticated before it can be discovered (e.g. Liberty Alliance discovery model)
    - Support signatures in web service security headers
SSA Advantages and Issues

Pros

- Can combine with existing SSO
- Easy to deploy on existing clients
- Re-uses well-established SSO framework with different scenario choices

Issues

- Some SSO mechanisms introduce more complicated protocol flows
- Some difficult security issues may remain (replay, confidentiality, adversarial SP)
SSA Approaches

- IDP provides secret
- IDP accessed as portal
- Enhanced Client or Proxy (ECP)
IDP Secret Approach

- Client expects secret to be provided
  - Client has stored IDP Secret at IDP
  - This is made unique per SP by using SP name
  - Protect against replay, provide confidentiality from SP and others by using hash, including time.

- Client may indicate capability & requirement in request

- In addition, SP knows IDP will not authenticate client unless SP presents SP authentication token, for the cases where SP requires client authentication
IDP Secret Approach

- SP authenticates (to Authentication Service) and obtains a token to be presented to the IDP.
- Redirect methods of ID-FF or SAML 2.0 can be used to achieve client authentication.
- Hash verification is simple, and does not require client signature verification (PKI and general key distribution).
- Issue – requires client to check secret, possible extension or plugin.
IDP Secret

Browser | Authentication Svc | Service Provider | Identity Provider
---|---|---|---
1. Access protected resource | 2. Authentication of SP Obtains token for IdP | Require client authentication | Require SP authentication
3. HTTP Redirect | 4. <AuthnRequest> | | |
5. Regular SAML 2.0 or ID-FF authentication | 6. HTTP Redirect | | |
7. Response with <SAML artifact> | | | |
10. Access granted Derived-secret | | | |

Frederick Hirsch – Hubert A. Le Van Gong - W3C Authentication Workshop - March 2006
IDP Accessed As Portal

- Identity provider can in fact also be a service provider and portal
- Client does not need to see shared secret, can rely on trusted IDP in this case
- Requires IDP configuration that SP authentication required
IDP Portal Operation

- Client authenticates to IDP
- Client then attempts to access another SP in circle of trust using link on portal site
- IDP can require SP authentication before allowing transfer
  - May pre-authenticate portal links
  - May remember recent authentication
- No dependency on SP requiring client authentication
**IDP portal operation when browser contacts IDP first**

1. Regular SAML 2.0 or ID-FF user authentication

2. Authentication of SPs (digital certs, challenge/response etc.)

3. Access protected resource (SP)

4. HTTP Redirect

5. <AuthnRequest>

6. HTTP Redirect + user AuthN assertion

7. Access granted

Frederick Hirsch – Hubert A. Le Van Gong - W3C Authentication Workshop - March 2006
Enhanced Client or Proxy Approach

• Intelligent client (or proxy) knows how to reach IDP
• Uses SOAP messages conveyed over reverse HTTP binding (PAOS)
• ECP enforces requirement for SP authentication, also actively participates in principal authentication to SP
• Re-uses mechanisms defined in SAML 2.0 standard
## SSA Approaches Summary

<table>
<thead>
<tr>
<th>Approach</th>
<th>ECP</th>
<th>IDP shared secret</th>
<th>IDP Portal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Benefits</strong></td>
<td>General, active component manages meeting mutual authentication requirements</td>
<td>Scalable shared secret with minimal client changes</td>
<td>Trusted intermediary (IDP)</td>
</tr>
<tr>
<td><strong>Limitations</strong></td>
<td>Requires enhanced client or proxy.</td>
<td>Agreement on the representation of the secret and implementation on the client.</td>
<td>Inherent portal limitations</td>
</tr>
<tr>
<td><strong>Additional Component?</strong></td>
<td>Yes (Enhanced client or Proxy)</td>
<td>Liberty Authentication Service technology – ID-FF</td>
<td>Liberty ID-FF technology or equivalent</td>
</tr>
<tr>
<td><strong>Specification Involved</strong></td>
<td>SAML 2.0 ECP or Liberty Alliance LECP</td>
<td>ID-FF &amp; ID-WSF (partial for AS)</td>
<td>ID-FF</td>
</tr>
<tr>
<td><strong>Changes to Client?</strong></td>
<td>No</td>
<td>Possibly</td>
<td>No</td>
</tr>
</tbody>
</table>
References

• Approaches to Simplify Server Authentication

• SAML 2.0 specifications

• Liberty specifications
  – https://www.projectliberty.org/resources/specifications.php