Metadaten

Organisationsstrukturen und Sicherheit in Shibboleth

3. Shibboleth-Workshop

Freiburg, 10. Oktober 2006

Bernd Oberknapp, UB Freiburg
E-Mail: bo@ub.uni-freiburg.de

Föderationen und Metadaten

Auf Softwareebene gibt
es **keine Föderationen**,
sondern **nur Metadaten**!

Bedeutung der Metadaten

- Metadaten bilden **Organisationsstrukturen** auf die **Ebene der Shibboleth-Software** ab.
- Metadaten sind die **Basis für das Vertrauen und die Sicherheit** in Shibboleth. Sie stellen sicher, dass man weiß, mit wem man Daten austauscht.
- Metadaten werden **signiert** um ihre **Authentizität und Integrität** zu gewährleisten – mit gefälschten Metadaten wäre es möglich, sämtliche Shibboleth-Sicherheitsmechanismen auszuhebeln!

Aktualisierung der Metadaten

- Die Metadaten müssen bei allen Teilnehmern **aktuell und synchron** sein um die Sicherheit zu gewährleisten und Interoperabilitätsprobleme zu vermeiden.
- Empfehlung: Metadaten **täglich aktualisieren**.
- **Die Signatur muss unbedingt geprüft werden!**
- **Shibboleth-Tools** für die Aktualisierung: metadatatool (IdP) und siterefresh (SP).
- **Wrapper-Skripte** von SWITCH vereinfachen die automatische Aktualisierung per **cron**-Job.

Shibboleth-Metadaten

- Shibboleth verwendet **SAML 2.0** Metadaten (bzw. ein Subset davon) mit eigenen **Erweiterungen**.
- Eine **Metadaten-Datei** definiert entweder eine **Entity** (einen IdP oder SP, EntityDescriptor) oder mehrere **Entities** (gruppiert mit EntitiesDescriptor).
- Der Name eines EntitiesDescriptor definiert die **RelyingParty** für die darin enthaltenen Entities.
- Die Einbindung der Metadaten-Dateien in die Konfiguration erfolgt über **MetadataProvider**.

Aufbau einer Metadaten-Datei

- Root-Element ist entweder **EntitiesDescriptor** oder EntityDescriptor**,** es enthält üblicherweise
- Signature-Element(e) mit **Signatur(en)**,
- Extensions-Element mit Shibboleth KeyAuthority-Element mit **CA-Zertifikaten** (optional) und
- EntityDescriptor-Element(e) mit den **Metadaten** für die **Identity Provider** und **Service Provider.**
- EntitiesDescriptor-Elemente können verschachtelt werden um hierarchische Strukturen abzubilden.

Aufbau eines EntityDescriptor

- **entityID**-Attribut mit dem **Namen des IdP/SP** (in der IdP/SP-Konfiguration: **providerId**)
- **Scope** (Shibboleth-Erweiterung)
- Informationen über die verwendeten **Zertifikate**:
  - Identifier für die Keys (bei Verifikation über CAs)
  - oder die Zertifikate selbst (inline, Base64-kodiert)
- **Kommunikationsendpunkte**
- Informationen zur **Einrichtung**:
  - Name und Homepage der Einrichtung
  - technische und administrative Ansprechpartner

Kommunikationsendpunkte

- **Basis-URL**s für die Interaktion mit dem **Browser** und die **interne Kommunikation** von IdP und SP
- Identity Provider:
  - SingleSignOnService (Browser)
  - AttributeService (intern)
  - ArtifactResolutionService (intern)
- Service Provider:
  - AssertionConsumerService für Browser/Artifact
  - AssertionConsumerService für Browser/POST

Beispiel: ReDI-metadata.xml

- Name des Root-Elements/RelyingParty: urn:mace:ub.uni-freiburg.de:redi
- Signatur mit Trustcenter-Zertifikat
- 3 CA-Zertifikate
- 66 Identity Provider
- 10 Service Provider
- Konvention für entityIDs und Zertifikate:
  - URLs als entityIDs und Inline-Zertifikate für aus ReDI-Sicht externe IdPs/SPs
  - URNs als entityIDs und Key-Identifier für von ReDI gehostete IdPs/SPs

Inline-Zertifikate

- Wenn **XML-Encryption** (ab Shibboleth 2.0) verwendet werden soll, müssen die Zertifikate selbst in die Metadaten aufgenommen werden.
- Bei **Verwendung von Inline-Zertifikaten** erfolgt lediglich ein Vergleich mit dem Zertifikat/Key-Paar, das die Gegenseite verwendet, es erfolgt **keine Prüfung anhand von CA-Zertifikaten**!
- Inline-Zertifikate ermöglichen eine **genauere Kontrolle** darüber, welches Zertifikat von welchem IdP/SP für welchen Zweck (SSL Server/Client, Signing, Encryption) akzeptiert werden.

Inline-Zertifikate

Dies hat weitreichende Konsequenzen:

- Mit der Verwendung von Inline-Zertifikaten sind die **CA-Zertifikate** für die Shibboleth-Software **nicht mehr relevant**.
- Die **Signatur der Metadaten** bekommt damit eine noch höhere Bedeutung.

➢ **Wer die Metadaten signiert, übernimmt damit praktisch die Rolle einer CA!**

Bilaterale Konfiguration

- **Einfachste Konfigurationsvariante**, ermöglicht die Kommunikation von einem IdP mit einem SP
- Metadaten:
  - eine Datei mit Metadaten für den IdP und den SP oder
  - je eine Datei mit Metadaten für den IdP und für den SP, die jeweils von der Gegenseite genutzt wird
- Die eigenen Metadaten interessieren den IdP/SP nicht, dieser greift grundsätzlich auf die eigene Konfiguration (idp.xml/shibboleth.xml) zurück!
- **Bilateral skaliert schlecht...**

„Federation“

- Konfiguration mit einer **Metadaten-Datei mit mehreren IdPs und SPs**
- Wer verwaltet und signiert die Metadaten-Datei?
- Eine **kooperative Pflege** ist bei mehr als einer Handvoll IdPs und SPs praktisch **nicht möglich**.
- Es muss jemand die **Verwaltung übernehmen**, und dafür sind entsprechende **Regeln notwendig**!

➢ **Die Festlegung von Regeln (Policies) und das Verwalten und Signieren der Metadaten sind die zentralen Aufgaben einer Föderation.**

„Multi-Federation“

- Unter Umständen sind **nicht alle IdPs/SPs**, mit denen man kommunizieren möchte, **in einer Metadaten-Datei** enthalten sind.
- Es können **mehrere Metadaten-Dateien** über entsprechende **MetadataProvider**-Elemente in die IdP/SP-Konfiguration eingebunden werden.
- Wenn der IdP/SP **für alle MetadataProvider dieselbe entityID und dieselben Zertifikate** verwenden kann, genügt dies bereits!

„Multi-Federation“

- Muss ein IdP/SP für verschiedene Metadata-Provider **verschiedene entityIDs oder Zertifikate** verwenden, so müssen **entsprechende Relying-Parties** in der Konfiguration definiert werden.
- Muss ein **IdP verschiedene Zertifikate** für verschiedene RelyingParties verwenden, so müssen für jedes Zertifikat auch **verschiedene Kommunikationsendpunkte** verwendet werden!

➢ **Eine Föderation sollte einen IdP/SP möglichst nicht zwingen, eine bestimmte entityID oder bestimmte Zertifikate zu verwenden**.

Wie erkennt ein IdP einen SP?

- Die **entityID des SP** ist entscheidend!
- Der IdP bestimmt damit die **RelyingParty**, dies ist
  - die **entityID des SP,** falls eine RelyingParty mit diesem Namen definiert ist, oder
  - der **Name des (innersten) EntitiesDescriptor**, der die Metadaten für den SP enthält.
  - Falls keine Metadaten für den SP gefunden werden, wird die **defaultRelyingParty** des IdP verwendet und der SP als „**unauthenticated**“ eingestuft.
- Die RelyingParty bestimmt das **Antwortverhalten des IdP** (verwendetes Zertifikat, Profil usw.).

Weitere Metadaten

- Einige für Shibboleth **wichtige Informationen** sind **nicht in den Metadaten-Dateien enthalten**!
- Wichtig für die **Interoperabilität** sind:
  - Welcher SP benötigt welche Attribute?
  - Welcher IdP kann welche Attribute liefern?
- Wichtig im Hinblick auf den **Datenschutz** ist:
  - Welche ARPs sollten verwendet werden?

➢ Die Föderation sollte diese Informationen **mit in die Verwaltung der Metadaten einbeziehen** und Empfehlungen für die ARPs aussprechen bzw. **entsprechende ARPs zur Verfügung stellen**.

Metadatenverwaltung

- Beispiel 1: **MAMS**
  - **XML Service Description**-Dateien mit Informationen zu den vom SP angebotenen Diensten und den notwendigen Attributen
  - Grundlage für **ShARPE/Autograph**

  http://federation.org.au/
- Beispiel 2: **SWITCHaai** Resource Registry
  http://www.switch.ch/aai/resourceregistry/

Metadaten und Föderationen

Auf Softwareebene gibt es keine Föderationen, sondern nur Metadaten. Aber **ohne Föderationen** gäbe es **keine verlässlichen Metadaten**!