Leitfaden zur Installation und Konfiguration des Shibboleth Identity Provider 2

Shibboleth-Workshop

Köln, 30. September 2009

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

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

Übersicht

- Vorüberlegungen
  - Identity Management
  - Verfügbarkeit
- Installation eines Basissystems
  - IP-Kontrolle
  - statische Attribute
- Anbindung des IdP an das IdM
  - Authentifizierung
  - Attribute
  - Beispiel: LDAP-Server des RZ Freiburg
- Produktionssystem

Vorüberlegungen

Identity Management

- Welche Anwendungen sollen unterstützt werden?
  - Welche Benutzergruppen müssen dafür abgedeckt werden?
  - Welche Informationen über die Nutzer werden benötigt?
- Voraussetzungen für den Beitritt zur DFN-AAI erfüllt?
  - Sind die Benutzerdaten entsprechend aktuell?
  - Wird ein sicheres Authentifizierungsverfahren verwendet (z.B. hinreichend sichere Passwörter)?
- Notwendige Verbesserungen so früh wie möglich angehen, da Änderungen beim Identity Management (IdM) meistens deutlich mehr Zeit kosten als der technische Aufbau eines Identity Providers (IdP)

Verfügbarkeit und Sicherheit

UNI FREIBURG

- Bei Ausfall des IdP – oder einer der verwendeten IdM-Komponenten – können die Nutzer auf keine der angebundenen SPs/Anwendungen mehr zugreifen!
- Um die notwendige Verfügbarkeit zur gewährleisten gibt es verschiedene Möglichkeiten, z.B.
  - Virtueller Server (mindestens zwei physikalische Server…)
  - Hochverfügbarkeitslösung (z.B. Linux HA wie bei myLogin)
  - Clustering (z.B. mit Terracotta)
- Der IdP muss durch Härtung des Betriebssystems und der Dienste entsprechend abgesichert werden!

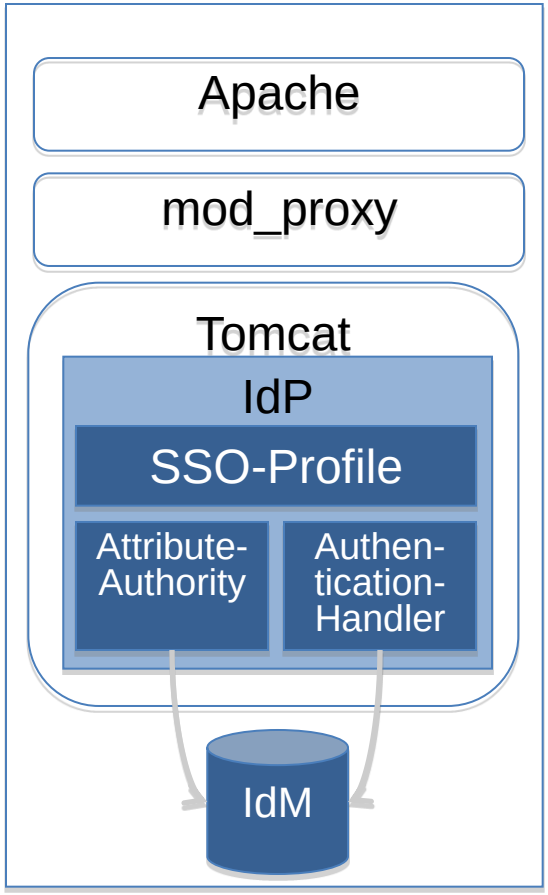
entityIDs

- Jeder IdP und Service Provider (SP) hat einen weltweit eindeutigen Namen, die sogenannte entityID
- Die entityID des IdP wird nicht nur in den Metadaten verwendet, sondern auch intern in SPs/Anwendungen für die Filterung von Attributen und die Autorisierung
- Die entityID muss daher langfristig stabil sein, sie im laufenden Betrieb zu ändern ist aufwendig und führt zu Störungen beim Login in SPs/Anwendungen!
- In der DFN-AAI werden URLs als entityIDs verwendet (die nicht auf eine Webseite verweisen müssen)
- Beispiel: https://idp.uni-tuebingen.de/shibboleth

Installation eines Basissystem

Komponenten des IdP

- Beispielkonfiguration:
  - openSUSE 11.1
  - Sun Java 1.6.0
  - Tomcat 6 als Servlet-Container
  - Apache 2.2 als Webserver
  - mod_proxy_ajp zur Anbindung von Tomcat an Apache
- Achtung: GNU Java VM wird nicht unterstützt!



Warum Tomcat und Apache?

- Tomcat ist der empfohlene Servlet-Container
- JBoss, WebSphere und andere Servlet-Container sollten funktionieren, dabei ist man aber weitgehend auf sich gestellt - das DFN-AAI-Team kann dafür auch keinen Support leisten
- Unter Linux/Unix müsste Tomcat als Webserver auf dem privilegierten Port 443 als root laufen
- Handhabung von Zertifikaten und Keys ist bei Apache (Dateien im PEM-Format) einfacher als bei Tomcat (Java Key Stores)
- Apache ist flexibler als Tomcat (URL Rewriting usw.)

Vorgehensweise

- Versuchen Sie nicht, zu Anfang gleich IdP und SP zu installieren und gegeneinander zu testen – das erhöht die Komplexität signifikant!
- Zunächst sollte ein möglichst einfaches, lauffähiges Basissystem installiert werden, das als stabile Ausgangsbasis für weitere Anpassungen dient:
  - für IdP und Apache wird das bei der Installation generierte selbst-signierte Zertifikat verwendet
  - Authentifizierung erfolgt über IP-Adressen
  - Attribute werden statisch generiert
- Der IdP kann dann in der DFN-AAI-Test angemeldet und gegen die DFN Test-SPs getestet werden

Installation IdP

- Bei der Installation wird gefragt nach
  - Zielverzeichnis (Default: /opt/shibboleth-idp)
  - Hostname, unter dem der IdP angesprochen wird (im Folgenden als IdP-Hostname bezeichnet)
  - Passwort für einen Java Keystore (wird bei der Konfiguration mit Apache nicht benötigt und sollte gelöscht werden)
- Installiert werden im Zielverzeichnis u.a.
  - Konfigurationsdateien des IdP (conf)
  - Selbst-signiertes Zertifikat und Key (credentials)
  - Metadaten für den IdP (metadata)
  - Web Application Archiv (war)

entityID ändern

- Default: https://IdP-Hostname/idp/shibboleth
- Wenn die entityID geändert werden soll, muss dies vor Aufnahme des Produktionsbetriebs erfolgen!
- Geändert werden müssen
  - conf/relying-party.xml (provider="…")
  - metadata/idp-metadata.xml – diese Datei wird bei der Erstinstallation des IdP generiert und muss danach bei Änderungen der Konfiguration mit angepasst werden, einen Automatismus wie beim SP-Metadatengenerator gibt es bisher nicht!
  - ggf. Eintrag in den DFN-AAI-Test Metadaten (Achtung, bei Änderungen der entityID erzeugt die Metadatenverwaltung automatisch einen neuen IdP/SP-Eintrag)

Tomcat konfigurieren

- Achtung: Bei RHEL/CentOS nicht den mitgelieferten „nativen“ Tomcat verwenden!
- Tomcat Manager-Anwendung
  - mit installieren und einrichten
  - Zugriff einschränken, am besten auf ein Management-Netz
  - ermöglicht Reload der IdP-Anwendung nach Änderungen der Konfiguration und damit ein einfacheres Testen
- Tomcat für die IdP-Anwendung vorbereiten:
  - verfügbaren Speicher für Java vergrößern, z.B. JAVA_OPTS="-Xmx1024M -XX:MaxPermSize=512M"
  - Xerces und Xalan als XML-Parser einbinden: IdP endorsed-Verzeichnis ins Tomcat endorsed-Verzeichnis kopieren

idp.war in Tomcat einbinden

UNI FREIBURG

- Verwenden Sie ein Context Deployment Fragment um die IdP-Anwendung in den Tomcat einzubinden:
  - einfache und saubere Lösung
  - erspart erfahrungsgemäß Ärger mit von Tomcat nicht erkannten Änderungen an der IdP-Anwendung
- Änderungen an der IdP-Anwendung müssen dann im Quellverzeichnis (ausgepacktes IdP-Archiv, z.B. src/main/webapps/login.jsp) vorgenommen werden
- Um Änderungen zu aktivieren, wird der IdP einfach erneut installiert, ohne dabei die Konfiguration zu überschreiben

Apache konfigurieren

- Die Dienste des IdP werden unterschiedlich genutzt, daher sind zwei VirtualHosts mit verschiedenen Konfigurationen erforderlich
- SingleSignOnService auf Default-Port 443:
  - Zugriff erfolgt per Browser durch den Nutzer
  - Zertifikat für diesen Port ist unabhängig vom IdP, es kann aber natürlich dasselbe Zertifikat verwendet werden
  - Anfragen an den IdP unter /idp/ werden per mod_proxy_ajp weitergegeben: ProxyPass /idp/ ajp://localhost:8009/idp/
  - Details siehe DFN-AAI-Dokumentation

Apache konfigurieren

- AttributeService und ArtifactResolutionService, üblicherweise auf Port 8443:
  - Zugriff per SOAP-Request durch SPs mit Authentifizierung per Zertifikat, das an den IdP durchgereicht werden muss:
    SSLVerifyClient optional_no_ca
    SSLVerifyDepth 10
    SSLOptions +StdEnvVars +ExportCertData
  - Zertifikat für diesen Port muss mit dem Zertifikat des IdP übereinstimmen, da SPs bei der Verbindungsaufnahme das Zertifikat anhand der Metadaten überprüfen
  - mod_proxy_ajp-Konfiguration wie Port 443

IP-Kontrolle

- IP-Kontrolle ist für den Produktionsbetrieb nicht sonderlich sinnvoll, aber für einen ersten Test besonders einfach zu konfigurieren
- In conf/handler.xml
  - LoginHandler RemoteUser auskommentieren
  - LoginHandler IPAddress mit den IP-Adressen, die Zugriff haben sollen, eintragen:

```
<LoginHandler xsi:type="IPAddress"
      username="ip-user" defaultDeny="true">
  <AuthenticationMethod>
      urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
  </AuthenticationMethod>
  <IPEntry>132.230.25.229/32</IPEntry>
</LoginHandler>
```

Attribute und NameIDs

- Attribute und NameIDs werden in vier Schritten erzeugt, bearbeitet und freigegeben:
  - DataConnector und PrincipalConnector erzeugen Daten
  - AttributeDefinitions definieren die Attribute und NameIDs
  - AttributeEncoder bringen sie in das richtige Format
  - AttributeFilterPolicies (AFP) geben sie ggf. frei
  - Dies ist ausführlich im Shibboleth-Wiki dokumentiert!
- NameIDs
  - identifizieren den Nutzer eindeutig, der SP kann damit z.B. Attribute für den Nutzer beim IdP anfordern
  - sind bei Shibboleth meist anonym (transient ID) oder pseudonym (persistent ID, bei 1.3 eduPersonTargetedID)

Statische Attribute

- Wie IP-Kontrolle für den Produktionsbetrieb nicht sonderlich sinnvoll, aber für einen ersten Test gut geeignet
- DataConnector, AttributeDefinition und -Encoder werden in conf/attribute-resolver.xml konfiguriert:
  - Example Static Connector aktivieren (einkommentieren)
  - AttributeDefinition für eduPersonAffiliation, eduPersonScopedAffiliation und eduPersonEntitlement aktivieren und ref="staticAttributes" eintragen
- AttributeFilterPolicy wird in conf/attribute-filter.xml konfiguriert, gibt diese Attribute für alle SPs in der DFN-AAI-Test frei

Anmeldung DFN-AAI-Test

UNI FREIBURG

- Metadaten für den IdP in der DFN-AAI-Test eintragen, am besten über den „Metadatengenerator“: https://IdP-Hostname/idp/profile/Metadata/SAML
- Zertifikat zum Verifizieren der Metadaten herunterladen und unter credentials speichern: https://www.aai.dfn.de/fileadmin/metadata/dfn-aai.pem
- DFN-AAI-Test-Metadaten in conf/relying-party.xml konfigurieren, Details siehe DFN-AAI-Dokumentation
- DFN-AAI-Metadaten werden jeweils zur vollen Stunde neu generiert, nach Änderungen müssen Sie solange mit dem Testen gegen die DFN Test-SPs warten

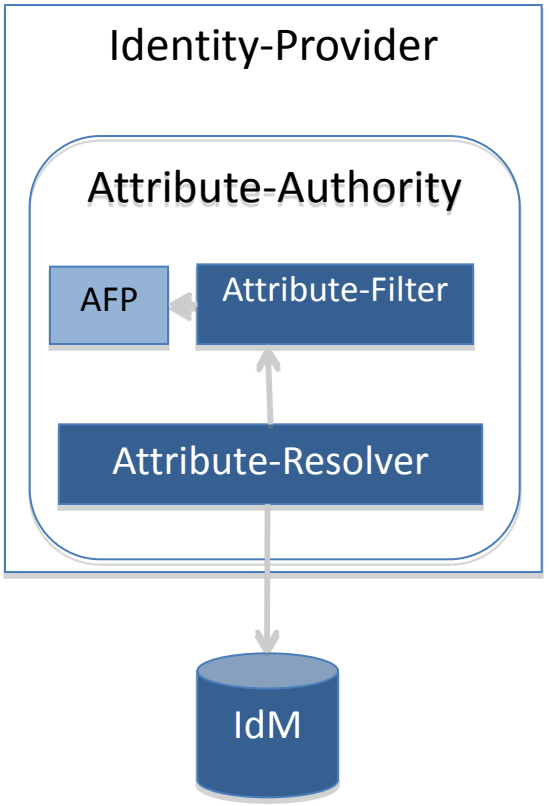
Funktionstest

- Wurde der IdP korrekt gestartet?
  - Tomcat Manager sollte anzeigen, dass der IdP läuft
  - https://IdP-Hostname/idp/profile/Status sollte „ok“ anzeigen
  - falls nicht, sollten Fehlermeldungen in idp-process.log (IdP) und catalina.out (Tomcat) Hinweise auf die Ursache liefern
- Test gegen DFN Test-SP 2.x und 1.3
  - falls Fehler auftreten oder Attribute nicht übermittelt werden, idp-process.log und ggf. Logdateien der SPs prüfen
  - Loglevel in logging.xml auf DEBUG hochsetzen, insbesondere bei Problemen mit Attributen
  - wenn etwas nicht funktioniert, ist fast immer der IdP schuld, da die DFN Test-SPs normalerweise funktionieren …

Anbindung des IdP an das IdM

Vorgehensweise

- Die Konfiguration sollte schrittweise angepasst werden
  - Authentifizierung
  - Attribute
  - Zertifikate
  - ggf. weitere Anpassungen
- Nach jedem Schritt sollte erneut ein Funktionstest durchgeführt werden, um bei Fehlern die Ursache einfacher eingrenzen zu können



Authentifizierung

- Mitgeliefert werden vier LoginHandler:
  - IPAddress
  - UsernamePassword
    - verwendet Java Authentication and Authorization Service (JAAS)
    - IdP hat die volle Kontrolle, es werden alle Funktionen unterstützt
  - RemoteUser
    - Authentifizierung wird an den Servlet-Container delegiert
    - nicht unterstützt werden Passive, ForceAuthn (und Logout)
  - PreviousSession
    - für Single Sign-on verantwortlich
- Einbindung eigener LoginHandler ist möglich, erfordert aber Java-, Spring- und XML Schema-Kenntnisse

UsernamePassword/LDAP

- LoginHandler IPAddress in conf/handler.xml deaktivieren und UsernamePassword aktivieren
- LDAP-Anbindung in login.config konfigurieren
- ggf. muss das LDAP-Zertifikat / das entsprechende CA-Zertifikat in den Java Keystore importiert werden
- Tomcat neu starten um die Änderungen zu aktivieren
- Genau wie die Konfiguration der Attribute ist auch die Konfiguration der Authentifizierung ausführlich im Shibboleth-Wiki dokumentiert!

LDAP login.config

- login.config für den LDAP-Server des RZ der Universität Freiburg (einfache Variante ohne Failover):

```
ShibUserPassAuth {
  edu.vt.middleware.ldap.jaas.LdapLoginModule required
  host="bv1.ruf.uni-freiburg.de"
  port="389"
  tls="true"
  base="ou=People,dc=uni-freiburg,dc=DE"
  userField="uid"
  authorizationFilter="rufStatus=enabled"
  ;
};
```

- Nach Änderungen muss Tomcat neu gestartet werden
- Zum Testen Loglevel in logging.xml hochsetzen

LDAP DataConnector

- DataConnector für den LDAP-Server des RZ der Universität Freiburg (einfache Variante ohne Failover):

```
<resolver:DataConnector id="rzLDAP" xsi:type="LDAPDirectory"
      xmlns="urn:mace:shibboleth:2.0:resolver:dc"
      ldapURL="ldap://bv1.ruf.uni-freiburg.de" useStartTLS="true"
      baseDN="ou=People,dc=uni-freiburg,dc=DE">
  <FilterTemplate>
    <![CDATA[
      (uid=$requestContext.principalName)
    ]]>
  </FilterTemplate>
</resolver:DataConnector>
```

Beispiele AttributeDefinition

- AttributDefinition Simple:
  - erzeugt Attribut aus DataConnector-Daten oder anderen Attributen, Attributwerte werden unverändert übernommen
  - Beispiel: Kostenstelle ist im RZ-LDAP im proprietären Attribut rufKostenstelle gespeichert, die Übermittlung an die SPs soll aber im Attribut departmentNumber erfolgen
- AttributeDefinition Mapping:
  - ermöglicht Abbildung von DataConnector-Daten oder Werten eines anderen Attributs auf die gewünschten Attributwerte
  - Beispiel (stark vereinfacht): eduPersonAffiliation wird aus dem Attribut rufAccountType des RZ-LDAP ermittelt, Mitgliedern und Angehörigen der Hochschule wird member zugeordnet

Beispiele AttributeDefinition

- AttributeDefinition Script:
  - erzeugt Attribute per ECMAScript (JavaScript, auch andere Skriptsprachen wären im Prinzip möglich)
  - ermöglicht auch sehr komplexe Bedingungen, Abhängigkeiten und Abbildungen von Attributwerten
  - Skripte werden automatisch in Java-Klassen kompiliert
  - Beispiel: urn:mace:dir:entitlement:common-lib-terms wird abhängig von den eduPersonAffiliations des Nutzers zu eduPersonEntitlements hinzugefügt
- Weitere AttributDefinitions siehe Shibboleth-Wiki

Entitlements

- Mit Entitlements lassen sich praktisch beliebige Nutzerrechte ausdrücken
- Zulässige Werte sind URNs und URLs
- URNs müssen registriert werden, z.B. für den Namensraum urn:geant:dfn.de beim DFN
- Empfehlung: Verwenden Sie URLs statt URNs!
- Beispiel: Nutzer ist berechtigt, die für den Zugriff von außerhalb der Badischen Landesbibliothek (BLB) freigegebenen Datenbanken in ReDI zu nutzen: https://www.blb-karlsruhe.de/entitlement/redi/extern

Persistent IDs

UNI FREIBURG

- Persistent IDs sind NameIDs, die eine Wiedererkennung des Nutzers ermöglichen
- Bei Shibboleth 1.3 wurde statt dessen das Attribut eduPersonTargetedID (deprecated) verwendet
- IdP 2 bringt mit dem StoredID DataConnector eine fertige Implementierung mit, die eine JDBC fähige Datenbank zu Speicherung der IDs verwendet
- Achtung: Persistent IDs müssen **wirklich langfristig** persistent sein, d.h. es muss auch in mehreren Jahren noch garantiert sein, dass derselbe Nutzer für denselben SP dieselbe persistent ID bekommt!

Produktionssystem

Attributfreigabe

- Aus Datenschutzgründen sollten grundsätzlich nur Attribute und Attributwerte freigegeben werden, die tatsächlich benötigt werden
- Achtung: Es dürfen auf keinen Fall Default-Regeln für Personen bezogene Attribute verwendet werden!
- Empfehlung: Definieren Sie pro SP eine eigene AttributeFilterPolicy mit den notwendigen Attributen

```
<AttributeFilterPolicy>
  <PolicyRequirementRule xsi:type="basic:AttributeRequesterString"
       value="entityID des SP" />

  … AttributeRules für den SP …
</AttributeFilterPolicy>
```

Weitere Anpassungen

- Logging auf das notwendige Maß beschränken (WARN-Level, alte Logdateien automatisch löschen)
- Layout der Login- und Fehler-Seiten anpassen
- Selbst signiertes Zertifikat gegen ein DFN-PKI Global Zertifikat austauschen (Typ Shibboleth IdP/SP, 5 Jahre gültig, auch für Apache verwendbar)
- Test-IdP zum Testen von Änderungen aufbauen
- Test-SP für weitergehende Tests installieren
- Firewall-Konfiguration anpassen
- Monitoring einrichten
- …

Und zum Schluss …

UNI FREIBURG

Mit der DFN-AAI in Betrieb gehen  ,
mehr dazu im folgenden Vortrag.

Vielen Dank für Ihre Aufmerksamkeit!

Fragen? Fragen!