Universität Bamberg

Shibboleth und Kerberos in gemischten Umgebungen

Erfahrungen und Grenzen

66. DFN-Betriebstagung, AAI-Forum, 21.03.2017

Frank Schreiterer – Universität Bamberg

Agenda

1. Ausgangssituation
2. Anpassungen
3. Implementierung

1. Ausgangssituation

- 2 getrennte Active Directories (RZ und ZUV)
- Clients ZUV domain-integriert
- Pools und VDI der UB bereits domain-integriert (RZ)
- Umstellung übriger Clients (= PCs der Lehrstühle und Zentren) auf Windows 10 domain-integriert (RZ)
- UB möchte von der IP-basierten Authentifizierung bei Verlagen, auch in der UB für Stadtnutzer, abschalten

Was bietet das Shibboleth-Standart-Paket?

Kerberos-Plugin (SPNEGO)
https://wiki.shibboleth.net/confluence/display/IDP30/SPNEGOAuthnConfiguration mit ggf. automatischer Anmeldung

1. Ausgangssituation

Probleme:

- automatische Anmeldung per Cookie, was aber manuell in der Anmeldemaske aktiviert werden muss
- Wie erklärt man nicht technisch versierten Nutzern, dass der Button „Windows-Anmeldung verwenden“ geklickt werden soll, statt nochmals Nutzername + Passwort einzugeben?
- Kerberos und keine Vertrauensstellung im IE führt zu 401-Request (Nutzername + Passwort) und / oder Fehlermeldung durch ntlm-Fallback
- 2 getrennte ADs

Ziel: transparente Nutzung, d.h. Anmeldung ohne Login-Masken und Mausklicks

2. Anpassungen

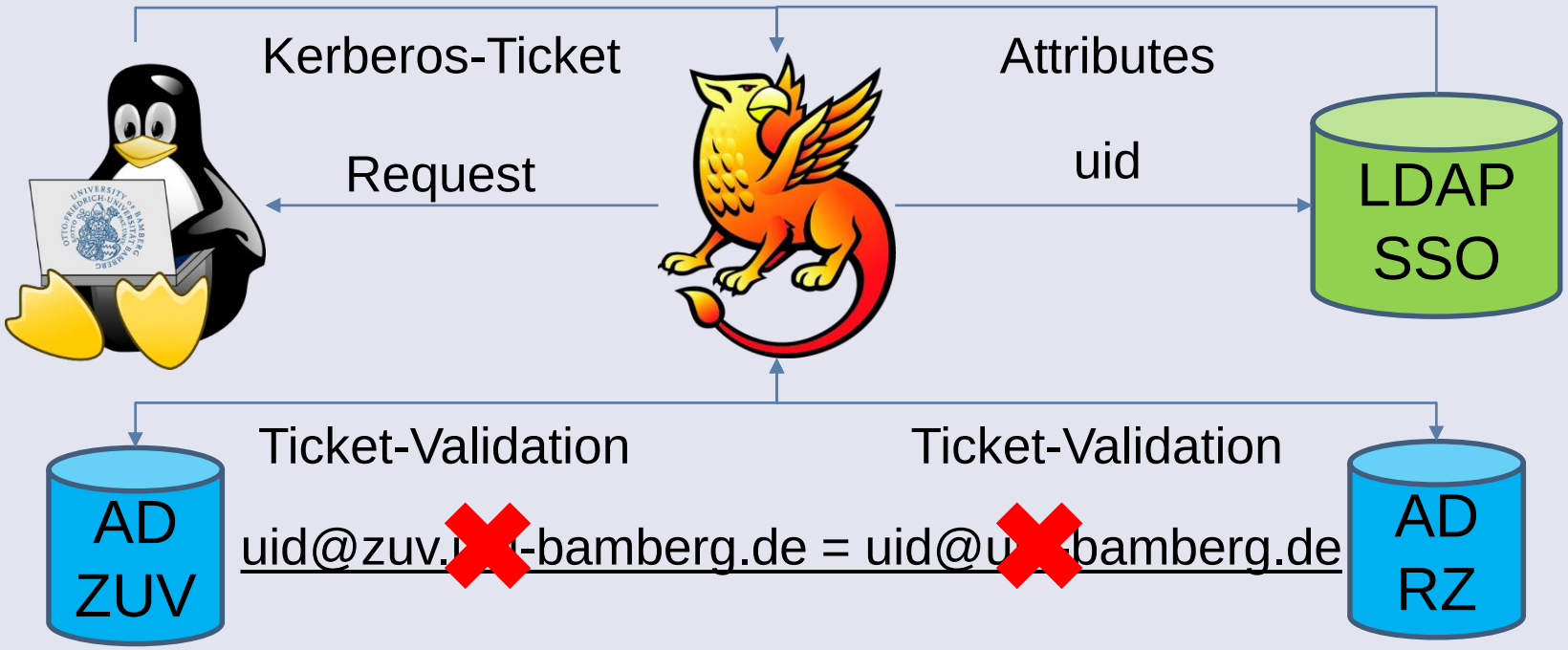
Windows: Hinzufügen der Vertrauensstellung

- nicht-domain-integrierte Clients: mittels Patch per WSUS verteilt
- Domain-integrierte Clients: über Gruppenrichtlinie
- Clients, die nicht im Einflussbereich des RZ liegen: Ausschluss erforderlich

Shibboleth-Standart-Paket:

- ActivationCondition für IP-Range und Ausschluss der IPs WLAN (Fremdclients) in *conf/authn/general-authn.xml*
- entfernen der Realms in *conf/c14n/simple-subject-c14n-config.xml* (ggf. auch Konfiguration in der attributeResolver.xml notwendig)

2. Anpassungen - Realms



Realm: zuv.uni-bamberg.de
Service-Account:
http://idp.rz.uni-bamberg.de

Realm: uni-bamberg.de
Service-Account:
http://idp.rz.uni-bamberg.de

```
<bean parent="shibboleth.Pair" p:first="^(.+)@UNI-BAMBERG\.DE$" p:second="$1"/>
<bean parent="shibboleth.Pair" p:first="^(.+)@ZUV.UNI-BAMBERG\.DE$" p:second="$1"/>
```

3. Implementierung

Versuch:

Kann das Anmeldecookie im Login-Vorgang erzeugt werden? ➔ keine Möglichkeit gefunden; aber: Es gibt JavaScript.

Idee:

Automatisches submit des SPNEGO-Buttons per JS und fallback LDAP, falls Kerberos nicht zur Verfügung steht oder fehl schlägt.

3. Implementierung login.vm

- Login-Buttons bekommen eindeutige id
- Login-Button SPNEGO per CSS nicht angezeigt ➔ mit und ohne JS nicht direkt verfügbar
- Einbindung der erforderlichen JavaScripts

Funktionsweise Auto-Login

1. gesamtes html wird per JS nicht angezeigt

   var div = document.getElementById("nodisplay");
   div.style.display = "none";
2. Beim laden wird per JS das SPNEGO-Login ausgelöst
   var e = document.getElementById("authn/SPNEGO");
   e.click;

3. Implementierung login.vm

SPNEGO nicht erfolgreich ➔ fallback zum Standard-Login

1. Fehlermeldungen von SPNEGO und ntlm unterdrücken
   auth-messages.properties:
   spnego-unavailable.message=
   ntlm.message=
2. gesamtes html wird per JS angezeigt

   var div = document.getElementById("nodisplay");
   div.style.csstext = null;

3. Implementierung login.vm

Erweiterungen:

1. Kein SPNEGO-Login für mobile Browser und nicht-Windows-Clients per JS
2. Unterdrückung Auto-Login per Cookie mit JS gesetzt über separate URL
3. Erzwungenes Auto-Login ebenfalls per Cookie mit JS gesetzt über separate URL (z.B. Laptop domain-integriert im WLAN)

3. Implementierung Login Stadtnutzer UB

- per LDAP durch Filter verhindert
- nur per SPNEGO für Stadtnutzer aus dem IP-Bereich der UB gewünscht
- kein Shibboleth-Login für andere Dienste der DFN-AAI und eduGain

➔ Post-authentication-Interceptor-Flow
https://wiki.shibboleth.net/confluence/display/IDP30/ContextCheckInterceptConfiguration

check-beans.xml:

```
<bean id="ContextCheckPredicate" parent="shibboleth.Conditions.OR">
  <constructor-arg>
    <list>
    <!-- Damit Stadtnutzer aus dem Subnetz der UB trotzdem die
    Dienste nutzen können muss der Zugriff gewährt werden -->
```

3. Implementierung Login Stadtnutzer UB

```
<bean parent="shibboleth.Conditions.AND">
  <constructor-arg>
    <list>
      <!-- Liste der zugelassenen SPs, in centralconditions.xml def.-->
      <ref bean="IsUBSP" />
      <!-- Das Format der uid prüfen -->
      <bean class="net.shibboleth.idp.profile.logic.
        RegexAttributePredicate" p:useUnfilteredAttributes="true">
        <property name="attributeId">
          <value>uid</value>
        </property>
        <property name="pattern">
          <value>^UB-user-pattern$</value>
        </property>
      </bean>
      <!-- Den IP-Bereich einschränken -->
      <bean class="org.opensaml.profile.logic.IPRangePredicate"
            p:httpServletRequest-ref="shibboleth.HttpServletRequest"
            p:ranges="#{{ '192.168.1.0/24','10.0.0.1/24' }}" />
    </list>
  </constructor-arg>
</bean>
```

3. Implementierung Login Stadtnutzer UB

```
<!-- jeder, der in der Gruppe usesso ist und die uid mit ba beginnt-->
<bean parent="shibboleth.Conditions.AND">
  <constructor-arg>
    <list>
      <bean class="net.shibboleth.idp.profile.logic.
       SimpleAttributePredicate" p:useUnfilteredAttributes="true">
        <property name="attributeValueMap">
          <map>
            <entry key="isMemberOf">
              <list>
                <value>usesso</value>
              </list>
            </entry>
          </map>
        </property>
      </bean>
```

3. Implementierung Login Stadtnutzer UB

```
            <bean class="net.shibboleth.idp.profile.logic.
             RegexAttributePredicate" p:useUnfilteredAttributes="true">
              <property name="attributeId">
                <value>uid</value>
              </property>
              <property name="pattern">
                <value>^ba.*$</value>
              </property>
            </bean>
          </list>
        </constructor-arg>
      </bean>
    </list>
  </constructor-arg>
</bean>
```

Zentrale Bedingungen einbinden in check-flow.xml

```
<bean-import resource="../../../conf/centralconditions.xml" />
```

3. Implementierung Demo