Installation eines Shibboleth SP in Windows

erstellt von Peter Vogl, Jan 2017, peter.vogl@drvis.de.

Das Ziel dieser Dokumentation ist die Beschreibung der Konfiguration für eine authentifizierte Anmeldung an eine beliebige managed ASP.NET Web-Applikation mittels des Protokolls SAML 2.0 (Security Assertion Markup Language 2.0). Dazu wird ein sog. Shibboleth ServiceProvider (im Weiteren mit SP abgekürzt) installiert, der Anmeldedaten des Benutzers an das Gegenstück des ServiceProviders, dem IdentityProvider (mit idP bezeichnet), weiterleitet und dann eine Antwort bekommt, mit der man entscheiden kann, ob der Benutzer zur Nutzung der Applikation zugelassen wird oder nicht.

Die technische Voraussetzung für die beschriebene Konfiguration ist ein Windows Server mit IIS 7 oder neuer. Diese Doku wurde mit Windows Server 2012 R2 und IIS 8.5 erstellt. Außerdem ist es notwendig, ein Webserver SSL/TLS Zertifikat auszustellen bzw. ausstellen zu lassen. Das kann mit Hilfe einer eigenen PKI oder mittels openssl und einer öffentlichen Zertifizierungsstelle erfolgen. Wie man ein Zertifikat im Detail ausstellt, wird hier nicht besprochen.

Grundlagen

Shibboleth ist aktuell eine SAML 2.0 Implementation. Obwohl die Unterstützung von Windows deutlich geringer ist als für diverse Linux-Distributionen, ist Shibboleth durch den häufigen Einsatz an Unis weltweit auch unter Windows weitgehend ausgereift. Es gibt ein zentrales Wiki, https://wiki.shibboleth.net/confluence/display/SHIB2/Home , das die primäre Quelle für Dokumentation und Software ist. Allerdings krankt diese Quelle an den typischen Open Source Problemen: die Dokumentation ist ziemlich holprig, es gibt Inkonsistenzen, Fehler und Widersprüche, einzelne Angaben sollte man nicht unbedingt wörtlich, sondern als Hinweise interpretieren, weil unterschiedliche Kapitel von verschiedenen Autoren zu verschiedenen Versionen geschrieben und später nicht immer auf Konsistenz überprüft und korrigiert wurden.

Die ausführlichste Dokumentation über Shibboleth für Windows, die ich gefunden habe, ist https://doit.missouri.edu/wp-content/uploads/2014/09/WindowsGuide.pdf . Der einzige Schwachpunkt dieser Dokumentation ist, dass auf die Einstellungen im IIS nicht eingegangen wird. Andererseits finden Sie in der Dokumentation der Uni Missouri eine ausführliche Diskusssion zur Installation eines reverse Proxy für den Fall, dass die zu schützende Anwendung keine Webapplikation ist. Darauf gehe ich hier nicht ein.

Situation in Deutschland

Der Einsatz von Shibboleth wird in Deutschland durch den Verein DFN (Deutsches Forschungsnetz) über https://www.aai.dfn.de/ koordiniert und unterstützt. Insbesondere betreibt das DFN nach Schweizer Vorbild (http://www.switch.ch/) eine Shibboleth Föderation, die den Einsatz von Shibboleth stark vereinfacht, weil durch die Föderation der erforderliche Austausch der Metadaten zwischen den IdPs und SPs automatisiert abläuft und nicht individuell ausgehandelt werden muss. Zusätzlich bietet das DFN kostenlose und exzellente technische Unterstützung.

Planung einer Shibboleth Webseite

Nehmen wir an, die URL der managed Webapplikation lautet http(s)://www.meineapp.de. Der Zugang zu dieser Webapplikation soll nur über SAML2 authentifizierten Benutzern möglich sein. Dazu wird man im Falle größerer Anforderungen zwei virtuelle Server oder Cluster benötigen, einen für die Webapplikation und einen für Shibboleth. Es spricht nichts dagegen, die eigene Webapplikation auf der gleichen IIS Installation und damit am gleichen Server zu nutzen wie Shibboleth, in jedem Fall benötigt man für Shibboleth eine eigene Web-Site (im Sinne von IIS), die mit SSL/TLS geschützt werden muss. Wir nennen diese Seite https://shib.org.de.

Shibboleth lässt sich nur schwer auf Cluster verteilen, am einfachsten kann man mehrere Shibboleth-Server per round-robin betreiben. Diese Dokumentation geht daher der Übersichtlichkeit halber davon aus, dass der Shibboleth-SP auf einer eigenen virtuellen Maschine mit IIS Installation läuft.

Es ist wichtig zu verstehen, dass die Toplevelseite der Shibboleth SP Site selbst nicht geschützt ist, sondern ein Unterverzeichnis, z. B. https://shib.org.de/login. Shibboleth nutzt nämlich selbst bestimmte virtuelle Unterverzeichnisse wie https://shib.org.de/Shibboleth.sso/... zum Austausch von Daten. Allerdings kann man die Startseite (z. B. Default.htm oder iisstart.htm) von https://shib.org.de per Javascript oder ASP umleiten, sodass der Nutzer automatisch auf https://shib.org.de/login landet. Dies besprechen wir noch später. Sobald ein Nutzer diese login-Seite aufruft, muss er sich authentifizieren. Sobald das geschehen ist, kann man den Nutzer auf die Webseite http(s)://www.meineapp.de weiterleiten und dort entsprechend den vom idP empfangenen Daten die eigene Applikation anzeigen. Der Datentransfer kann über die URL (GET-Methode) erfolgen, die transferierten Daten müssen dabei so verschlüsselt werden, dass spoofing durch einen falschen idP oder durch Modifikation der Daten ausgeschlossen werden kann.

Wichtig ist auch zu verstehen, welche Daten man vom IdP erhält. Im universitären Umfeld stellen die IdPs in Deutschland nahezu ausschließlich nur pseudonyme Benutzer-Daten zur Verfügung. Diese enthalten den Status des Benutzers (Studierender oder Mitarbeiter) und einen HashWert, der dem Benutzer universitätsintern eindeutig zugeordnet ist, ähnlich der Matrikelnummer. Leider ist es mit der Eindeutigkeit oft nicht weit her, z. B. wenn der Hashwert den Nachnamen enthält oder die Uni die Matrikelnummern bei Änderung der Studienrichtung ändert.

Anmeldung an DFN

Die ersten Schritte zur Installation von Shibboleth sind auf https://www.aai.dfn.de/ genau beschrieben: man muss sich beim DFN als SP "in spe" anmelden und einen Vertrag unterzeichnen. Daraus resultiert ein Konto, mit dessen Hilfe letztlich die eigenen Metadaten des SP in die Föderationsdatenbank eingetragen werden können. Insbesondere bietet DFN eine sehr nützliche Test-Föderation zu Testzwecken. Sinnvollerweise ersteht man nun ein öffentliches SSL Zertifikat für shib.org.de, evtl. noch mit einem SubjectAlternateName wie z. B. www.shib.org.de. Wenn man eine eigene Windows PKI betreibt, erstellt man dazu einen request-File und ergänzt diesen dann mit dem signierten Zertifikat der öffentlichen Zertifizierungsstelle zu einem pfx-File. Mit der Installation von Shibboleth wird auch openssl installiert. Das ist nützlich, wenn man keine eigene PKI betreibt und andererseits selbst mit einer PKI erforderlich, um den pfx File (nennen wir ihn shib.pfx), den eine Windows PKI erzeugt, in den privaten (shibpriv.pem) und öffentlichen Schlüsselteil (shibpub.pem) zu trennen. Die Befehle dazu lauten

"C:\Program Files (x86)\Shibboleth\sp\lib\openssl.exe" pkcs12 -in shib.pfx -nocerts -nodes -out shibpriv.pem

"C:\Program Files (x86)\Shibboleth\sp\lib\openssl.exe" pkcs12 -in shib.pfx -clcerts -nokeys -out shibpub.pem

Diese Dateien speichert man in <Installationsverzeichnis>\opt\shibboleth-sp\etc\shibboleth, in dem auch die Default Zertifikatsdateien sp.cert.pem und sp-key.pem liegen.

3. Konfiguration eines virtuellen Servers

Wenn der virtuelle Server in Microsoft Azure liegt, sind folgende Punkte bei der Erzeugung der viertuellen Maschine (VM) zu beachten:

- Da sehr wenig Rechenkapazität erforderlich ist, genügt eine A0 Standard Instanz.
- Man muss die Endpoints für Port 80 und 443 erzeugen.
- Eine Daten-Platte (z.B. 10 GB) der VM hinzufügen, z. B. Volume F. Dies ist erforderlich, da die Datenverarbeitung auf der Systemplatte extrem langsam wäre.
- Firewall Regel: Zumindest Firewall Port 443 inbound öffnen.

Nennen Sie den hostname der VM shib ohne Hinzufügen eines Suffixes. Editieren Sie dann c:\windows\system32\drivers\etc\hosts wie folgt:

```
127.0.0.1  localhost
::1 localhost
192.168.x.x shib.org.de shib
```

wobei 192.168.x.x die private IP-Adresse der VM ist. Nun kann man mittels Servermanager den IIS installieren und dabei folgende Rollen und Features auswählen:

a. Web Server
    i. Common HTTP Features
        1. Default Document
        2. Directory Browsing
        3. HTTP Errors
        4. Static Content
        5. HTTP Redirection
    ii. Health and Diagnostics
        1. HTTP Logging
        2. Request Monitor
        3. Tracing
    iii. Performance
        1. Static Content Compression
    iv. Security
        1. Request Filtering
        2. Basic Authentication
        3. Windows Authentication
        4. IP and Domain Restriction
    v. Application Development
        1. .NET Extensibility 4.5
        2. ASP
        3. ASP .NET 4.5
        4. ISAPI Extensions
        5. ISAPI Filters
b. Management Tools
    i. IIS Management Console
    ii. IIS 6 Management Compatibility
        1. IIS 6 Metabase Compatibility

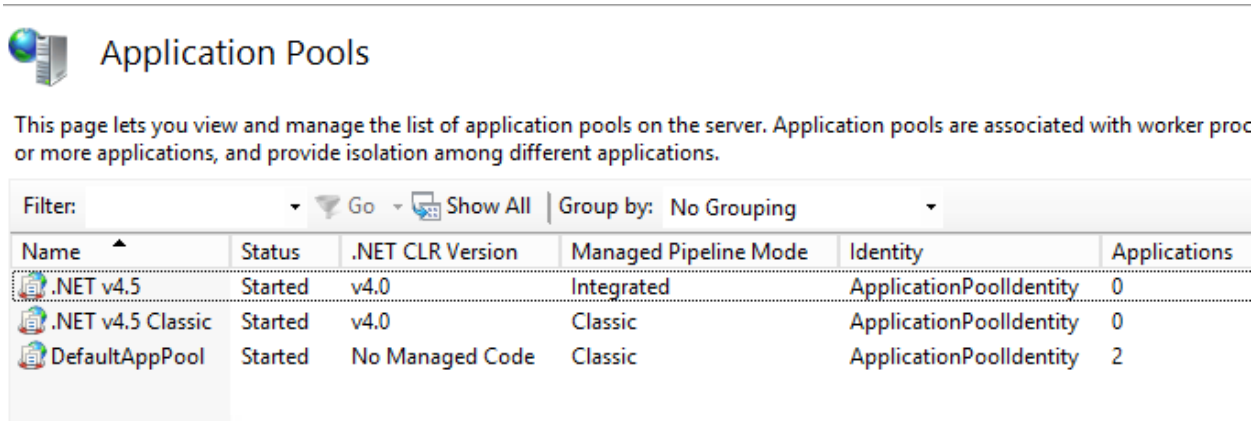
2. IIS Management Console
3. IIS 6 Scripting Tools
4. IIS 6 WMI Compatibility

iii. IIS Management Scripts and Tools
iv. Management Service

4. Konfiguration des IIS

Als nächstes sollte das Zertifikat für die Webseite https://shib.org.de in den Computerstore importiert werden (mmc -> Ctrl-M -> Doppel-Click Certificates -> Computer Account und in den private store importieren. Es ist wichtig, dass die Zwischenzertifikate und das Stammzertifikat in den richtigen Ordnern abgelegt werden). Anschließend kann man die Default Web Site für https konfigurieren, in dem man das Zertifikat an https bindet.
Schließlich wird man die öffentliche IP der VM bei einem DNS Hoster publizieren (hier für shib.org.de und evtl. www.shib.org.de).

Shibboleth ist kein managed .NET code, aber der IIS konfiguriert standardmäßig jede Webseite mit einem Application Pool, der die managed code pipeline mit .NET Framwork 4.5 integriert. Für Shibboleth alleine ist das egal, aber man benötigt klassische ASP oder ASPX Seiten zur Steuerung der Daten, die man vom idP erhält, und diese sind nicht kompatibel mit dem managed code. Sie müssen daher den Application Pool, der die Shibboleth WebSite enthält (standardmäßig also DefaultAppPool), auf "no Managed Code" und "Classic" umstellen, wie hier gezeigt:



5. Herunterladen von Shibboleth

Auf der Webseite

https://shibboleth.net/downloads/service-provider/latest/win64/

kann man die msi Datei vom Shibboleth SP, shibboleth-sp-2.6.0.0-win64.msi, herunterladen.

6. Installieren von Shibboleth

- **Beachten Sie, dass der Shibboleth Parser case-sensitive ist. Daher ist es wichtig, bei den URLs Großbuchstaben wie Shibboleth.sso, Status usw. mit Großbuchstaben anzugeben. Ansonsten erhält man kryptische Fehlermeldungen.**
- Erzeugung eines Verzeichnisses auf einer Datenplatte, z. B. F:\opt\shibboleth-sp\
- Shibboleth SP auf diesem Verzeichnis installieren und rebooten.
- Sofort testen, ob http://127.0.0.1/Shibboleth.sso/Status funktioniert (Achtung auf Großschreibung!)
- Rechte: Read-Rechte für Local Users und IIS_IUSRS in F:\opt und C:\Program Files (x86)\Shibboleth

Die einzig wirklich essentielle Datei, die man anpassen muss, ist die Konfigurationsdatei F:\opt\shibboleth-sp\etc\shibboleth\shibboleth2.xml und wir besprechen dies im Detail anschließend. Die Bezeichnung "2" beruht auf der aktuellen Shibboleth Version (zum Zeitpunkt dieses Artikels ist die aktuelle Version für ServiceProvider 2.6, für IdentityProvider aber bereits 3.x).

Konfigurieren der Test-Föderation ohne EDS (Embedded Discovery Service)

- Herunterladen der neuesten Versionen der Metadaten von https://www.aai.dfn.de/teilnahme/metadaten/ in das Verzeichnis F:\opt\shibboleth-sp\etc\shibboleth
  - die signierten Metadaten der DFN-AAI-Test-Föderation
  - die signierten Metadaten der DFN-AAI-Basic-Föderation
  - die signierten Metadaten der DFN-AAI-Advanced-Föderation
  - das DFN-AAI Zertifikat dfn-aai.pem im PEM-Format
- Hinzufügen der Zertifkatfiles shibpriv.pem und shibpub.pem.
- Erzeugung eines Unter-Verzeichnisses login in inetpub\wwwroot für das geschützte Verzeichnis
- Erzeugung eines Unter-Verzeichnisses help in inetpub\wwwroot für die Hilfe-Webseite
- Erzeugung eines Unter-Verzeichnisses error in inetpub\wwwroot für die Error-Webseite
- Erzeugung eines Unter-Verzeichnisses images in inetpub\wwwroot für die Logos und hinzufügen der Logos: ein großes Logo (H: 64-164px W: 64-350px und ein kleines Logo (16x16 px). Das kleine Logo ist de facto überflüssig und sollte auch nicht in die Metdatenbank aufgenommen werden.
- Anpassen von shibboleth2.xml zunächst für die DFN Testföderation
  - Der Handler Status muss die privaten und öffentl. IP Adressen aller Rechner enthalten, von denen man aus den Status überprüfen will

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    clockSkew="180">
    <InProcess logger="native.logger">
        <ISAPI normalizeRequest="true" safeHeaderNames="true">
            <Site id="1" name="shib.org.de" scheme="https" port="443"/>
        </ISAPI>
    </InProcess>

    <RequestMapper type="Native">
```

```
        <RequestMap>
            <Host name="shib.org.de">
                <Path name="login" authType="shibboleth" requireSession="true"/>
            </Host>
        </RequestMap>
    </RequestMapper>

    <ApplicationDefaults entityID="https://shib.org.de/shibboleth"
            REMOTE_USER="eppn persistent-id targeted-id"
            cipherSuites="ECDHE+AESGCM:ECDHE:!aNULL:!eNULL:!LOW:!EXPORT:!RC4:!SHA:!SSLv2">
    <Sessions lifetime="28800" timeout="3600" relayState="ss:mem" checkAddress="false"
              handlerSSL="true" cookieProps="https">
      <SSO
            discoveryProtocol="SAMLDS" discoveryURL="https://wayf.aai.dfn.de/DFN-AAI-Test/wayf">
              SAML2
      </SSO>
      <Logout>SAML2 Local</Logout>
      <Handler type="MetadataGenerator" Location="/Metadata" signing="false"/>
      <Handler type="Status" Location="/Status" acl="127.0.0.1 ::1 10.x.x.x y.x.x.x"/>
      <Handler type="Session" Location="/Session" showAttributeValues="true"/>
      <Handler type="DiscoveryFeed" Location="/DiscoFeed"/>
    </Sessions>
    <Errors supportContact="your_support@org.de"
            helpLocation="/help/iisstart.htm"
            styleSheet="/shibboleth-sp/main.css"/>
    <MetadataProvider type="Chaining">
    <MetadataProvider type="XML"
           uri=https://www.aai.dfn.de/fileadmin/metadata/DFN-AAI-Test-metadata.xml
           backingFilePath="F:\opt\shibboleth-sp\etc\shibboleth\DFN-AAI-Test-metadata.xml"
           reloadInterval="7200">
         <MetadataFilter type="RequireValidUntil" maxValidityInterval="2419200"/>
         <MetadataFilter type="Signature"
             certificate="F:\opt\shibboleth-sp\etc\shibboleth\dfn-aai.pem"/>
     </MetadataProvider>
     </MetadataProvider>
     <AttributeExtractor type="XML" validate="true" reloadChanges="false"
           path="attribute-map.xml"/>
        <AttributeResolver type="Query" subjectMatch="true"/>
        <AttributeFilter type="XML" validate="true" path="attribute-policy.xml"/>
        <CredentialResolver type="File" key="F:\opt\shibboleth-sp\etc\shibboleth\shibpriv.pem"
          certificate="F:\opt\shibboleth-sp\etc\shibboleth\shibpub.pem"/>
    </ApplicationDefaults>
    <SecurityPolicyProvider type="XML" validate="true" path="security-policy.xml"/>
    <ProtocolProvider type="XML" validate="true" reloadChanges="false" path="protocols.xml"/>
</SPConfig>
```

- Nun sollte man zunächst die Syntax des xml Files in einer shell überprüfen:
  `F:\opt\shibboleth-sp\sbin64\shibd.exe -check F:\opt\shibboleth-sp\etc\shibboleth\shibboleth2.xml`
- Wenn dies keine Fehler liefert, kann man nun den shibboleth Service (shibd.exe) neu starten und anschließend den IIS neu starten.
- Nun sollte man per Browser testen, ob https://shib.org.de/Shibboleth.sso/Status funktioniert. Dies ist nur auf den Rechnern möglich, deren IP Adresse man im Status eingetragen hat.

Einrichten der Testföderation

Als Nächstes muss man den eigenen ServiceProvider den anderen Mitgliedern der DFN Föderation, also anderen IdPs und SPs, bekannt machen. Dazu meldet man sich auf https://www.aai.dfn.de/teilnahme/online-verwaltung/ mit dem Konto an und trägt die eigenen Metadaten am Metadatengenerator ein:

URL	https://shib.org.de/Shibboleth.sso/Metadata
Entity ID	https://shib.org.de/shibboleth
DisplayName (de und en)	Your_Org
Beschreibung (de)	Ihr ANgebot
Beschreibung (en)	Free of charge Microsoft Office and Windows for German students
Information URL (deutsch)	https://shib.org.de/help/iisstart.htm
Information URL (en)	https://shib.org.de/help/iisstart.htm
Privacy Statement URL (de)	https://shib.org.de/.../Datenschutz_DE.htm
Privacy Statement URL (en)	https://shib.org.de/.../Datenschutz_EN.htm
Logo klein (URL)	leer lassen, das große wird automatisch skaliert
Logo groß (URL)	https://shib.org.de/images/logo.gif
Kontakte	
Typ	Technical nach dem Speichern wiederholter Aufruf und Wahl: Support
Vorname	Max
Nachname	Muster
Email	max.muster@mail.de
Discovery Response	
Location	https://shib.org.de/Shibboleth.sso/Login
NameId Formate	urn:oasis:names:tc:SAML:2.0:nameid-format:persistentID
Attribute Consuming Service	eduPersonScopedAffiliation nach dem Speichern wiederholter Aufruf und Wahl: eduPersonTargetedID

Man kann nun mit den auf der DFN Seite angegebenen Testkonten versuchen anzumelden. Wenn dies klappt, ist der nächste Schritt der Übergang zur produktiven Föderation

Wechsel zur Basic+Advanced Föderation

In der Online-Verwaltung https://www.aai.dfn.de/teilnahme/online-verwaltung/ der DFN Metadaten wird eine Auswahl zwischen der DFN-AAI (Advanced) Föderation und der DFN-AAI-Basic Föderation angezeigt. Tatsächlich aber inkludiert die Basic-Föderation beide, also Basic und Advanced, daher gibt es für SPs zumindest aktuell keinen Grund, nicht die Basic-Föderation auszuwählen. Dazu sind nur kleine Änderungen im Konfigurationsfile shibboleth2.xml erforderlich:

```
 <SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"

....
   <SSO
       discoveryProtocol="SAMLDS" discoveryURL="https://wayf.aai.dfn.de/DFN-AAI-Basic/wayf">
              SAML2
    </SSO>
    <MetadataProvider type="Chaining">
        <MetadataProvider type="XML" uri="https://www.aai.dfn.de/fileadmin/metadata/DFN-AAI-
Basic-metadata.xml" backingFilePath="F:\opt\shibboleth-sp\etc\shibboleth\DFN-AAI-Basic-
metadata.xml" reloadInterval="7200">
        ....
</SPConfig>
```

Als Hintergrund muss man wissen, dass Shibboleth die Metadaten stündlich aktualisiert. Wenn dies aus welchen Gründen auch immer nicht funktioniert, greift Shibboleth auf die lokal gespeicherte Kopie der Metadaten zurück.

Sofern man das noch nicht getan hat, muss man vor der Änderung der Föderation die DFN kontaktieren und evtl. einen Vertrag abschließen.

Nach dem Ändern der Föderation auf Basic+Advanced ist ein Neustart des shibboleth Service und iisreset erforderlich.

Umlenken der Startseite und Transfer der Daten an eigene App

Wie schon angemerkt, schützt Shibboleth nicht die Top-Levelseite, sondern ein im Konfigurationsfile anzugebendes Verzeichnis, in unserem Beispiel login. Damit der Nutzer aber die Top-Levelseite wählen kann, müssen wir diese auf das Verzeichnis login umleiten. Dazu gibt es viele Möglichkeiten, es ist aber nicht empfehlenswert, dies direkt im IIS zu setzen, denn für Statusabfragen benötigen wir die Haupseite. Daher setzen wir den Redirect entweder per ASP oder per Javascript. Dazu geben Sie im IIS in der Default Web Site als oberstes Default Dokument die Datei Default.htm an. Diese Datei (der vollständige Pfad lautet C:\inetpub\wwwroot\Default.htm) kann dann folgenden Inhalt bekommen:

```
<script type="text/javascript">
window.location = "https://shib.org.de/login/index.asp";
</script>
```

Der Benutzer darf Javascript im Browser nicht deaktiviert haben; dies ist aber eine generelle Bedingung für Shibboleth.

Nun können wir ein ASP Skript in die Startseite des Verzeichnisses login integieren, das die vom idP erhaltenen Benutzerdaten verarbeitet und im Erfolgsfall an unsere Applikation weiterleitet. Dazu erzeugen wir die Datei C:\inetpub\wwwroot\login\index.asp. Zum Testen empfehlen wir ein Skript im Appendix C der eingangs erwähnten Dokumentation https://doit.missouri.edu/wp-content/uploads/2014/09/WindowsGuide.pdf. Hier ein gekürztes Beispiel, das die vom idP erhaltenen Benutzerdaten "EduPersonScopedAffiliation" und "persistentID" in eine lokale Datei speichert.

```
<!DOCTYPE html>
<html>
<body>
<%
eduperson = Request.ServerVariables("HTTP_AFFILIATION")
persist = Request.ServerVariables("HTTP_PERSISTENTID")
filename = "F:\logs\output.txt"
Dim fso, f1
Set fso = CreateObject("Scripting.FileSystemObject")
Set f1 = fso.CreateTextFile(filename, True)
f1.WriteLine(eduperson)
f1.WriteLine(persist)
f1.Close
Response.Redirect "https://www.meineapp.de"
%>
</body>
```

```
</html>
```

Die URL für den Redirect ist in dieser angegebenen Form natürlich nicht sachgerecht, denn sie enthält keine Nutzerdaten. In Wirklichkeit wird man in dem Redirect auch die Daten (mit "&") in der URL anhängen, die dann in der eigenen Applikation verarbeitet werden. Dieser URL-basierte Transfer von Daten entspricht der GET Methode, die den Nachteil hat, dass die Daten in der URL für jeden lesbar (und daher veränderbar) sind. Die Daten in der URL wird man daher geeignet verschlüsseln und evtl. durch weitere Daten gegen Missbrauch schützen.

Konfiguration des Embedded Discovery Services

Der Nachteil der Standardkonfiguration ist, dass der Benutzer die Standard-Benutzeroberfläche des DFN vorgesetzt bekommt und dann aus allen idPs seine Uni auswählen muss. Außerdem ist es für den Benutzer verwirrend, wenn er als Einstiegsseite die DFN Seite vorgesetzt bekommt, dann die Login-Seite seiner Uni und erst dann die eigentliche von ihm erwartete Webseite - jede davon in einem anderem Layout - vorgesetzt bekommt.

Dies lässt sich durch ein Konzept, das in Shibboleth "Embedded Discovers Service" heißt, einfach ändern. Nehmen wir an, Sie möchten Ihre App 2 Universitäten anbieten, der Uni1 und der Uni2. Im ersten Schritt entnehmen Sie aus den Metadaten (entweder von https://www.aai.dfn.de/teilnahme/metadaten oder im Installationsverzeichnis) die EntityID der Unis, also die Webadressen deren idPs. Diese haben in der Regel eine Syntax ähnlich wie https://idp.uni1.de/shibboleth. Dann passen wir den shibboleth2.xml File an. Hier die Änderungen in der Konfigurationsdatei shibboleth2.xml

```
<SSO discoveryProtocol="SAMLDS" discoveryURL="https://shib.org.de/index.htm">
    SAML2
</SSO>
...
<MetadataProvider type="Chaining">
   <MetadataProvider type="XML" uri="https://www.aai.dfn.de/fileadmin/metadata/DFN-AAI-Basic-
metadata.xml" backingFilePath="C:\opt\shibboleth-sp\etc\shibboleth\DFN-AAI-Basic-metadata.xml"
reloadInterval="3600">
    <MetadataFilter type="RequireValidUntil" maxValidityInterval="2419200"/>
    <MetadataFilter type="Signature" certificate="C:\opt\shibboleth-sp\etc\shibboleth\dfn-
aai.pem"/>
    <MetadataFilter type="Whitelist">
         <Include>https://idp.uni1.de/shibboleth</Include>
         <Include>https://idp.uni2.de/idp/shibboleth</Include>
    </MetadataFilter>
  </MetadataProvider>
</MetadataProvider>
....
```

Die Whitelist enthält die idPs, die man dem Benutzer auswählen lassen möchte. In der obersten Zeile steht die Webadresse einer Datei index.htm, die in unserer Default Web Site liegt, und zwar nicht im geschützten Verzeichnis login, sondern im obersten und daher von Shibboleth nicht geschützten Ordner C:\inetpub\wwwroot. Diese Datei (C:\inetpub\wwwroot\index.htm) ist die Startseite, die der Benutzer tatsächlich angezeigt bekommt, wenn er https://shib.org.de eingibt.

Das klingt bestimmt verwirrend. Wir haben doch vorher einen redirect von der Seite https://shib.org.de[/Default.htm] auf die Seite https://shib.org.de/login/index.asp gesetzt. Ist dieser redirect jetzt außer Kraft gesetzt? Nein, das ist er nicht. Shibboleth leitet die Eingabe in der Tat auf login/index.asp weiter. Bevor diese Datei abgearbeitet wird, übernimmt Shibboleth aber die Kontrolle und verlangt eine Webseite, auf der der Benutzer seinen idP auswählen kann, der dann sofort kontaktiert wird. Erst anschließend wird das index.asp abgearbeitet. Standardmäßig stellt diese Webseite die DFN zur Verfügung, aber für den Benutzer ist es viel verständlicher, wenn er ein konsistentes Layout und nur eine relevante Auswahl von idPs bekommt.

Was steht nun in der Datei C:\inetpub\wwwroot\index.htm? Shibboleth stellt dazu einen Download bereit: https://shibboleth.net/downloads/embedded-discovery-service/latest/shibboleth-embedded-ds-1.1.0 und auch dazugehörige Dokumentation unter https://wiki.shibboleth.net/confluence/display/EDS10/2.+Installation#id-2.Installation-2.3WebPageSetup

Tatsächlich benötigt man nur 3 bzw. 4 Dateien, nämlich idpselect_config.js, idpselect.js, index.html und den Style-File idpselect.css, den man aber in index.html integrieren und anpassen kann. Angepasst werden muss vor allem index.html File und zwingend das Javascript File idpselect_config.js. Abgesehen vom generellen Layout, Hintergrundbild und den in der Vorlage schon eingetragenen Stilelemente enthält der index.htm File dann die Zeilen

```
...
        <p>
         Bitte w&auml;hlen Sie Ihre Hochschule aus.
        </p>
        <div id="idpSelect"></div>
        <script src="idpselect_config.js" type="text/javascript"></script>
        <script src="idpselect.js" type="text/javascript"></script>
...
```

Im File idpselect_config.js sind nur wenige Daten zwingend anzupassen

```
...
    this.dataSource = 'https://shib.org.de/Shibboleth.sso/DiscoFeed';
    this.defaultLanguage = 'de';
    this.defaultLogo = 'images/logo.gif';
    this.defaultReturn =
"https://shib.org.de/Shibboleth.sso/DS?SAMLDS=1&target=https://shib.org.de/login/index.asp";
    this.helpURL = 'https://www.hilfeseite.de;
```

Abfragen von Attributen (AttributeQuery)

Schließlich gibt es noch eine häufige Anforderung, die nur eine kleine Änderung im Konfigurationsfile erfordert. Nehmen wir an, Sie möchten zu einem bestimmten Zeitpunkt überprüfen, ob ein Benutzer, dessen persistentID Sie bereits kennen, noch Mitglied der Universität ist. Sofern der idP der Universität korrekt konfiguriert ist, ist das wie folgt möglich.

Zunächst die erforderlichen kleinen Ergänzungen in shibboleth2.xml:

```
<SPConfig ... clockSkew="180">
    <OutOfProcess>
```

```
      <Extensions>
        <Library path="plugins.so" fatal="true"/>
      </Extensions>
    </OutOfProcess>
    <InProcess logger="native.logger">
        <Extensions>
           <Library path="plugins-lite.so" fatal="true"/>
        </Extensions>
        <ISAPI normalizeRequest="true" safeHeaderNames="true">
            <Site id="1" name="shib.org.de" scheme="https" port="443"/>
        </ISAPI>
    </InProcess>
...
<Sessions ...>
...
  <Handler type="AttributeResolver" Location="/AttributeResolver"
       acl="127.0.0.1 ::1 10.x.x.x y.x.x.x" />
</Sessions>
```

Nach einem Neustart des shibd Services und einen iisreset kann Shibboleth nun Abfragen an den idP stellen. Sie können die Abfrage z. B. mit folgendem kleinen Powershell-Skript von allen Maschinen aus absetzen, deren IP-Adressen Sie im AttributeResolver angegeben haben:

```
$Gateway =
'https://shib.org.de/Shibboleth.sso/AttributeResolver?format=urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent'
$EntityId = 'https://idp.uni1.de/shibboleth'
$persistentID = abcdefgh'
$totalstring = $Gateway + '&entityID=' + $EntityId + '&nameId=' + $persistentID
$query = Invoke-RestMethod $totalstring
```

Es gibt noch viele weitere Ergänzungen und Erweiterungen und einige interessante sind in den Anhängen nvon https://doit.missouri.edu/wp-content/uploads/2014/09/WindowsGuide.pdf ausführlich besprochen.

Wir wünschen Ihnen gutes Gelingen für Ihren SP.