bwIDM: Föderieren auch nicht-webbasierter Dienste auf Basis von SAML

Michael Simon, Marcel Waldvogel, Sven Schober, Saher Semaan, Martin Nussbaumer

simon@kit.edu, marcel.waldvogel@uni-konstanz.de,
sven.schober@uni-ulm.de, semaan@uni-freiburg.de, nussbaumer@kit.edu

Abstract: Zur organisationsübergreifenden Nutzung von IT-Diensten werden Dienst-Föderationen gebildet. Dabei kann das Nutzerkonto der sogenannten Heimateinrichtung auch zum Zugriff auf nicht-lokale Dienste genutzt werden. Während die Integration webbasierter Dienste in Föderationen mit SAML und beispielsweise Shibboleth mittlerweile in vielen Anwendungsbereichen allgegenwärtig ist, fällt die Integration nicht-webbasierter IT-Dienste schwer. Existierende Ansätze, mit denen sich prinzipiell auch nicht-webbasierte Dienste integrieren lassen, erfüllen essentielle Anforderungen nicht und/oder sind nach ihrem heutigen Entwicklungsstand noch nicht betriebsfähig. In diesem Papier werden zwei Verfahren für nicht-webbasierte, föderative Dienstzugriffe (Moonshot und PAM/ECP) evaluiert und notwendige Erweiterungen zur Sicherstellung der Betriebsfähigkeit vorgestellt. Ein implementierter Proof-of-Concept zeigt die Umsetzbarkeit der Lösung.

1 Einleitung

Die zunehmende organisationsübergreifende Nutzung von IT-Diensten ist heutzutage nicht nur in der Forschungsgemeinschaft ein deutlich erkennbarer Trend. Zur effizienten und effektiven Arbeit ist die Nutzung von Diensten, die nicht lokal in der eigenen Organisation zur Verfügung stehen oder gestellt werden können, oft unabdingbar. Sind diese Dienste über einen Browser - demnach webbasiert - erreichbar, stellt dies für den Nutzer kaum noch besondere Hürden dar, obwohl oft ein Registrierungsschritt und die Preisgabe diverser personenbezogener Daten notwendig sind, um Zugang zum gewünschten Dienst zu erhalten. Insbesondere im Umfeld der wissenschaftlichen Dienstnutzung werden diese lokal beim Dienst implementierten Registriervorgänge und damit auch Dienst-lokale Nutzerverwaltungen bereits immer häufiger durch die für den Nutzer wesentlich komfortableren SAML[1]-basierten Implementierungen, wie zum Beispiel Shibboleth[2] oder SimpleSAMLphp[3], ersetzt. Diese Verfahren ermöglichen es, ein Nutzerkonto, das die eigene Organisation für den Nutzer verwaltet, zur Nutzung von Diensten Dritter einzusetzen.

Durch den Einsatz solch föderativer Verfahren zur Authentifikation (AuthN) und Autorisierung (AuthZ) wird Nutzern ein größtmöglicher Komfort geboten, indem die Lokalität

[1] SAML: Security Assertion Markup Language, http://saml.xml.org/saml-specifications
[2] http://shibboleth.internet2.edu/
[3] http://simplesamlphp.org/

eines Dienstes weitestgehend verborgen wird und keine zusätzlichen Nutzerkonten zum Zugriff notwendig sind. Ferner ist dadurch in der Regel Single Sign-on (SSO) zwischen den in einer Föderation beteiligten Diensten möglich. Im Folgenden werden Mechanismen als *föderative Verfahren (FV)* bezeichnet, bei denen Nutzer dasjenige Konto für den Zugriff auf Dienste Dritter nutzen können, das ihre sogenannte Heimateinrichtung verwaltet.

Werden - kontrastierend zu webbasierten Diensten - nun die Dienste betrachtet, zu denen keine Möglichkeiten des webbasierten Zugriffs existieren, ist eine deutlich schlechtere Nutzerunterstützung in Form von FV feststellbar. High Performance Computing- (HPC), Grid- oder Cloud-Dienste sowie großskalige Datendienste bedürfen Dienst-lokale Nutzerkonten. Dies wird aus technischen Gründen auch in Zukunft nicht vermeidbar sein, jedoch lassen sich Dienst-lokale Konten durch FV vor einem Nutzer verbergen, indem das föderative Konto auf ein z.B. ad-hoc eingerichtetes Dienst-lokales Konto abgebildet wird. Obwohl Nutzern zusätzliche initiale Aufwände zur Dienstnutzung und oft auch ungewollte Hürden aufgebürdet werden, setzen Dienstbetreiber heute zumeist Personen und/oder umständliche Registriervorgänge zur Einrichtung Dienst-lokaler Konten ein. Begründet ist dies dadurch, dass bereits implementierte FV derzeit nur mit sehr viel Aufwand auf nicht-webbasierte Zugänge, wie zum Beispiel SSH-Zugänge, adaptierbar sind.

In diesem Papier werden zwei Ansätze vorgestellt, mit denen sich FV für nicht-webbasierte Dienste implementieren lassen, und im Hinblick auf ihre Tauglichkeit im produktiven Einsatz untersucht. Als Basis für diese Evaluation dient ein im Folgenden vorgestellter Anforderungskatalog. Darauf aufbauend werden die Anforderungen identifiziert, denen durch die beiden föderativen Verfahren derzeit nicht nachgekommen werden kann: Dies betrifft insbesondere die Provisionierung und Deprovisionierung sowie Möglichkeiten zur Implementierung von Zustimmungsverfahren. Ein Ansatz für eine Kombination aus einem dieser Verfahren und den notwendigen Erweiterungen wird vorgestellt. Ein entsprechend implementierter Proof-of-Concept zeigt die Umsetzbarkeit des zuvor beschriebenen Konzepts. Zusammenfassend sind die folgenden Beiträge Inhalt dieser Arbeit:

- Aufstellung eines Anforderungskatalogs zur Bewertung von FV
- Evaluation zweier existierender bzw. im Aufbau befindlicher FV
- Konzept zur Implementierung eines den Anforderungen entsprechenden Verfahrens
- *Proof-of-Concept* zur Verdeutlichung der Betriebsfähigkeit des Konzepts

Die hier vorgestellten Ergebnisse entstammen dem vom Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg (MWK) unterstützten Landesprojekt bwIDM[4].

Das Papier ist folgendermaßen strukturiert: Kapitel 2 gibt einen Überblick über den Stand der Technik in Bezug auf ausgewählte Komponenten und Mechanismen, die bei FV zum Einsatz kommen. In Kapitel 3 werden die Anforderungen an ein Verfahren für föderative und nicht-webbasierte Dienstzugänge zusammengetragen, bevor in Kapitel 4 zwei existierende bzw. im Aufbau befindliche Verfahren auf Basis dieser Anforderungen evaluiert

[4] www.bw-grid.de/bwservices/bwidm/

werden. Kapitel 5 beschäftigt sich mit den notwendigen Erweiterungen eines dieser Verfahren und Kapitel 6 fasst das Papier abschließend zusammen und gibt einen Ausblick auf die nächsten Schritte im Aufbau eines betriebsfähigen, föderativen Verfahrens.

2 Stand der Technik

FV müssen die Delegation der AuthN von Nutzern an Identitäts-Provider (IdP) unterstützen sowie die Möglichkeit bieten, AuthZ-Merkmale an einen Dienst zu liefern. Dies erfordert etablierte Vertrauensstellungen zwischen den Föderationspartnern. Im Folgenden werden ausgewählte System-APIs sowie zwei FV kurz vorgestellt.

2.1 Ausgewählte System-APIs zur Authentifikation

Eine Integration nicht-webbasierter Dienste in FV ist nicht trivial. Es bietet sich an, dafür Komponenten zu verwenden, die eine einheitliche AuthN-Schnittstelle bieten. Im Folgenden werden bekannte Schnittstellen unixoider Systeme vorgestellt.

Pluggable Authentication Modules (PAM) [Sam96] ist eine von Samar und Lai entwickelte Schnittstelle, die es ermöglicht eine zentrale AuthN bereit zu stellen. Sie stellt ferner grundlegende Funktionen zum Account-, Passwort- und Session-Handling zur Verfügung. PAM wird heute in nahezu allen unixoiden Systemen verwendet und kann aufgrund seiner Modularität erweitert werden.

Das **Generic Security Service Application Program Interface (GSS-API)** [Lin93] stellt eine Abstraktionsebene für Dienste bereit, die verschiedene Protokolle Betriebssystemunabhängig kapselt. Mittels GSS lassen sich AuthN-Mechanismen nutzen, ohne diese selbst zu implementieren. Ein Beispiel hierfür ist die Nutzung von Kerberos [NYHR05] über GSS-API [Lin96] (Implementierungen: MIT [NYHR05] oder Heimdal [DW98]).

2.2 Ausgewählte föderative Implementierungen

Eduroam (Education Roaming) [MRW+08] (gefördert durch GÉANT2-Projekt) ist ein Verfahren, das es Nutzern ermöglicht, lokale WLAN-Infrastruktur mit den Zugangsdaten der Heimateinrichtung zu verwenden. Basis dafür bildet das Extensible Authentication Protocol (EAP), durch welches das Ergebnis der AuthN an den Access Point vor Ort gelangt [ABV+04]. Die Föderation wird durch eine baumartige RADIUS-Server-Hierarchie aufgespannt [RWRS00]. Eine robuste RADIUS-Implementierung stellt Radiator dar, welche die Weiterleitung von Nachrichten über gesicherte Verbindungen mit definierten Vertrauensstellungen unterstützt [WMVW11]. Eine Adaption weiterer Dienste - neben WLAN - ist beispielsweise im Rahmen des eduGAIN-Projektes geplant [HVS10].

Die SAML-basierte **Shibboleth**-Software wurde von der Internet2 Middleware Initiati-

ve entwickelt und erlaubt die Umsetzung einer *Authentication and Authorization Infrastructure* (AAI). Die AAI des Deutschen Forschungsnetz (DFN-AAI) verwaltet SAML-basierte Föderationen für die deutschen Hochschulen. Bei einem Dienstzugriff innerhalb einer AAI delegiert ein Dienst (Service Provider, SP) die AuthN des Nutzers an einen IdP. Darüber hinaus unterstützt SAML den verschlüsselten Austausch von Attributen zwischen IdP und SP zur AuthZ des Nutzers. Neben Shibboleth existieren weitere SAML-Implementierungen, wie beispielsweise SimpleSAMLphp.

3 bwIDM-Anforderungskatalog für föderative Verfahren

Die Auswahl einer technischen Grundlage für die AuthN/AuthZ für nicht-webbasierte IT-Dienste richtet sich nach den Anforderungen anzubindender Dienste. Im Rahmen von bwIDM wird auf Cloud-Speicherdienste und damit Protokolle wie CIFS oder NFS sowie insbesondere Computing-Dienste (HPC, Grid etc.) mit SSH-Zugängen fokussiert.

Neben Anforderungen an die AuthN müssen AuthZ-Merkmale an IT-Dienste übermittelt werden können *(Anforderungsblock A-1)*. Dazu gehören beispielsweise Attribute wie Name, E-Mail-Adresse oder die Institutionszugehörigkeit im Sinne der Landeshochschulgesetze (siehe auch eduPerson-Schema des DFN-Vereins[5]). Die Übertragung soll gemäß den Schutzzielen der IT-Sicherheit erfolgen (vgl. [Eck09], Kapitel 1.2). Ferner gilt der datenschutzrechtliche Grundsatz der Datensparsamkeit. Werden die Attribute an Dritte übertragen, sind darüber hinaus Einverständniserklärungen der Nutzer sowie ggf. Zustimmungen zu Nutzungsrichtlinien (engl.: Acceptable Use Policies, AUP) notwendig.

Werden Attribute zum Dienst übertragen, müssen diese Daten aktuell gehalten beziehungsweise widerrufen werden können. Demnach sind die Provisionierung und Deprovisionierung für FV inhärente Anforderungen *(Anforderungsblock A-2)*. Bei einer erstmaligen Anmeldung an einen Dienst über ein FV werden ggf. Attribute als Grundlage für eine Zugriffsentscheidung übermittelt. Die Aktualität dieser Attribute ist insbesondere dann wichtig, wenn diese für weitere Prozessschritte im Anschluss an die eigentliche Dienstnutzung Verwendung finden. Dies kann zum Beispiel bei einem Bibliotheksdienst der Fall sein, bei dem Ausleihen getätigt wurden und ein Mahnverfahren notwendig wird. Beispiele für eine notwendige Deprovisionierung sind Dienst-lokale Konten sowie die Reservierung von persistent adressierbaren Ressourcen bei Computing-Diensten. Verliert ein Nutzer die Zugriffsberechtigung, bedarf es entsprechender Richtlinien und technischer Verfahren.

Einen dritten Anforderungsblock *(A-3)* bilden Merkmale, die ausschlaggebend für die Betriebsfähigkeit des Ansatzes sind. Hierzu gehört die Vermeidung zentraler Komponenten, wenn deren Funktionalität in dezentralen Komponenten abgebildet werden kann, um Wartungsaufwände und Abhängigkeiten gering zu halten. Ferner soll das FV absehbar zukunftssicher sein. Kriterien für die Abschätzung der Zukunftssicherheit sind die Art der Fortentwicklung eines FV und die Möglichkeit dieses in bestehende föderative Verbände - wie beispielsweise die DFN-AAI - zu integrieren. Ferner sollen die Aufwände zur Implementierung und zum Betrieb möglichst gering gehalten werden (Lizenz-/Hardware/

[5] https://www.aai.dfn.de/der-dienst/attribute/

Wartungskosten und Nutzer-Support). Beispielsweise steigt der Aufwand für den Nutzer-Support, wenn Anmeldevorgänge nicht selbsterklärend sind oder wenn Nutzer-lokale Anpassungen erforderlich werden. Letztendlich muss das Verfahren skalierbar im Hinblick auf die Anzahl der Nutzer, Dienste und teilnehmenden Einrichtungen sein.

4 Evaluation föderativer Verfahren

Im Folgenden werden zwei Ansätze zur Implementierung von FV vorgestellt, jeweils hinsichtlich der in Kapitel 3 vorgestellten Anforderungen evaluiert und die Betriebsfähigkeit nach aktuellem Entwicklungsstand diskutiert. In die Evaluation fließen insbesondere auch die Erfahrungen aus einer jeweiligen Test-Implementierung beider Ansätze ein.

4.1 Moonshot-Ansatz

Im Projekt *Moonshot*[6] (gefördert von JANET (UK) und GÉANT) werden Anpassungen und Kombinationen von Standard-Software-Komponenten gängiger Betriebssysteme für FV entwickelt. Kern des Ansatzes bildet die Kombination der GSS-API und EAP (vgl. Kapitel 2) sowie die Integration in RADIUS-Infrastrukturen. Des Weiteren soll zur Übermittlung von AuthZ-Attributen SAML eingesetzt werden. Im Fokus der Entwicklung stehen nicht nur unixoide Systeme, die durch Anpassung der GSS-API integriert werden, sondern auch Windows-Systeme. Moonshot strebt an, Anpassungen von Standardkomponenten zukünftig zu vermeiden, indem veränderte Mechanismen in Form von RfCs dokumentiert und in die Distributionen der einzelnen Betriebssysteme integriert werden.

Moonshot lehnt sich technologisch sehr stark an die bereits etablierten Mechanismen an, die im Rahmen von eduroam zum Einsatz kommen. Ein Nutzer wird hierbei ausschließlich direkt gegenüber seiner Heimateinrichtung authentifiziert, so dass die Anmeldedaten sowohl im Sinne des Datenschutzes als auch der IT-Sicherheit ausreichend geschützt sind. Bislang ist noch offen, wie Merkmale des Nutzers von der Heimateinrichtung an den Dienst übertragen werden, die etwa zur AuthZ benötigt werden; gerade dies ist aber im Rahmen von bwIDM eine zentrale Anforderung *(Anforderungsblock A-1)*.

Des Weiteren wird der Bereich der Provisionierung und Deprovisionierung von Nutzerkonten durch Moonshot bislang nicht betrachtet. Jedoch sind diese elementaren Prozesse eines FV Anforderungen im Rahmen von bwIDM *(Anforderungsblock A-2)*.

Schließlich setzt Moonshot voraus, dass beim Rechner des Nutzers angepasste Programme beziehungsweise Bibliotheken vorliegen. Dies ist derzeit im Allgemeinen nicht der Fall und auch frühestens mittelfristig zu erwarten, so dass zunächst auf jedem Client zusätzliche Software installiert werden muss. Dies bedeutet einen nicht unerheblichen Wartungs- und Supportaufwand *(Anforderungsblock A-3)*.

[6] http://project-moonshot.org/

4.2 PAM/ECP

Enhanced Client or Proxy (ECP) ist ein SAML-Profil [HCH+05], das - im Gegensatz zum sonst oft üblichen WebSSO-Profil - für nicht-webbasierte Zugangssysteme Verwendung finden kann. Dabei wird entweder ein erweiterter *Client* benötigt, der dieses Profil nativ unterstützt, oder ein *Proxy*, der die Funktionalität vor dem Client verbergen kann.

Der SAML-Standard spezifiziert *Assertions* zur Übertragung von Attributen, um den Diensten AuthZ-Merkmale bereit zu stellen *(Anforderungsblock A-1)*. Bezüglich der Datensicherheit sollte bei der Übertragung der Attribute jedoch auf einen sicheren Kanal geachtet werden. Dies ist bei SAML nicht verpflichtend, wird aber empfohlen und üblicherweise durch HTTPS als Übertragungsstandard umgesetzt. Bei gängigen SAML-Implementierungen - wie etwa Shibboleth - kann festgelegt werden, welchem Dienst welche Attribute übermittelt werden*(Anforderungsblock A-1)*. Dadurch werden die Anforderungen bzgl. der Datensparsamkeit erfüllt. Derzeit ist jedoch das Einholen einer expliziten Erlaubnis zur Übermittlung von personenbezogenen Daten innerhalb eines PAM-Moduls nicht möglich. Es können auch keine Änderungen an personenbezogenen Daten ohne erneute Interaktion des Nutzers übermittelt werden *(Anforderungsblock A-2)*. Zusätzlich ist bei der einfachsten Variante von PAM ein Vertrauen gegenüber dem Dienst notwendig, da dieser in seiner Funktion als Proxy Kenntnis der Zugangsdaten bekommen kann.

Ferner unterstützt die aktuelle Shibboleth-Implementierung eines Identity Provider ECP „nur“ mit der sogenannten *HTTP Basic Authentication*. Das Anmelden ist demnach jedoch wie gefordert mit einer Nutzername-/Passwort-Kombination möglich. Durch die Verwendung von PAM werden zudem nachgelagerte oder vorausgehende AuthN-Mechanismen eines Dienstes nicht beschränkt. Effektiv kann also bei einem SSH-Dienst weiterhin zusätzlich eine Public-/Private-Key-AuthN vom SSH-Server direkt vorgenommen werden, wie es dem heutigen Vorgehen entspricht. Grundsätzlich wird die Kombination von PAM und ECP implizit von allen SAML-Föderationen mitgetragen. Bei SAML handelt es sich ferner um einen etablierten Standard, den bereits viele Hochschulen beispielsweise im Rahmen der Teilnahme an der DFN-AAI unterstützen. Wird bei den Institutionen ein Shibboleth IdP in einer neueren Version verwendet, ist ECP durch eine Änderung der Konfiguration aktivierbar. *(Anforderungsblock A-3)*

5 Notwendige Erweiterungen: Konzept und Implementierung

Wie in den vorangegangenen Abschnitten erläutert wurde, kann mit Hilfe von PAM/ECP auch auf unixoiden Systemen die AuthN von Nutzern über Shibboleth-Mechanismen erfolgen. Allerdings werden hierbei weitere Fragen und Probleme aufgeworfen, die nicht unmittelbar die AuthN als solche betreffen. Da für die Nutzung eines unixoiden Systems ein Eintrag in einer Name Switching Service-Datenbank (NSS) notwendig ist, kann auf einen Registriervorgang nicht verzichtet werden. Dieser Registriervorgang kann in einer Webanwendung realisiert werden. Dort ist es möglich das Einverständnis des Nutzers zur Datenübermittlung einzuholen. Da die aktuellen Daten bei jedem weiteren Lo-

gin übermittelt werden, besteht die Möglichkeit, dass personenbezogene Daten, die sich geändert haben, ohne erneutes Einverständnis des Nutzers an den Service Provider gemeldet werden. Konkret handelt es sich hier demnach um die vor- beziehungsweise nachgelagerten notwendigen Vorgänge, also die Provisionierung und Deprovisionierung von Dienst-lokalen Nutzerkonten. Zusätzlich ist es möglich und wünschenswert, eine explizite Bestätigung der Kenntnisnahme der Acceptable Use Policy (AUP) durch den Nutzer zu fordern. Diese drei Aspekte erfordern zusätzliche Infrastruktur, wie in den folgenden Abschnitten erläutert wird. Im Rahmen von bwIDM wurden diese Ergänzungen in Form eines Proof-of-Concept-Prototypen implementiert.

5.1 Provisionierung von Nutzern durch einmaliges „Registrieren“ (Web)

Einer der Vorteile einer dezentralen AuthN, wie sie mit Shibboleth zur Verfügung gestellt wird, besteht darin, dass neue Nutzer mit relativ wenig Aufwand angelegt werden können. Es ist nicht notwendig, alle potentiell betroffenen Systeme von dieser Änderung in Kenntnis zu setzen; vielmehr ist es ausreichend, dem zuständigen Identity Provider gegenüber die Daten des neuen Nutzers bekannt zu geben. Dies hat aber andererseits zur Folge, dass ein Dienst, der Shibboleth zur AuthN von Nutzern verwendet, per definitionem keine A-priori-Kenntnis haben kann, welcher Nutzer potentiell den Dienst in Anspruch nehmen wird. Dementsprechend ist es dem Dienst unmöglich, von sich aus etwaig notwendige Vorbereitungen zu treffen, um jedem potentiellen Nutzer Zugang zum Dienst zu gewähren.

Um beispielsweise auf einem Linux-Rechner, der als Dienst oder Dienstzugang genutzt wird, sinnvoll arbeiten zu können, ist es häufig notwendig, dass der Nutzer über ein lokales Nutzerkonto verfügt, das über längere Zeit hinweg unverändert zur Verfügung steht. Insbesondere sind eine wohldefinierte Unix-UID und ein wohldefiniertes Home-Verzeichnis notwendig, wenn nicht bei jedem Zugriff auf den Dienst alle notwendigen Daten erneut auf die Linux-Maschine kopiert und nach erfolgter Benutzung des Dienstes alle Ergebnisdaten wieder auf einen anderen, permanenten Speicher zurückkopiert werden sollen. Genau dieses Modell findet beispielsweise im Grid-Umfeld Anwendung und erlaubt es mangels der Notwendigkeit, Daten für längere Zeit aufzubewahren, Dienstnutzer für die Dauer der einzelnen Nutzung dynamisch auf einen im Prinzip beliebigen lokalen Account abzubilden. Die damit verbundenen Nachteile sollen jedoch im Projekt bwIDM vermieden werden, so dass es insbesondere notwendig ist, für jeden Dienstnutzer ein permanentes (oder wenigstens mittelfristig stabiles) lokales Nutzerkonto einzurichten.

Dies erfolgt im vorgestellten Prototyp mit Hilfe eines webbasierten Selbstbedienungsportals, das der Nutzer einmalig vor dem allerersten Dienst-Zugriff besuchen muss. Das Webportal erlaubt es dem Anwender, nach erfolgter Shibboleth-Webauthentifikation auf „Knopfdruck“ ein lokales Nutzerkonto erzeugen zu lassen, das mit seiner Shibboleth-Identität verknüpft wird. Durch diese „Registrierung“ des Nutzers wird Dienst-seitig eine (aus Nutzersicht zunächst zufällige) UID gewählt und damit ein lokales Konto eingerichtet sowie das zugehörige Home-Verzeichnis angelegt. Nach dieser Registrierung steht das Konto permanent und stabil zur Verfügung, so dass der Nutzer einen „normalen“ SSH-Zugang zum Dienst bereitgestellt bekommt. Insbesondere können aus technischer Sicht

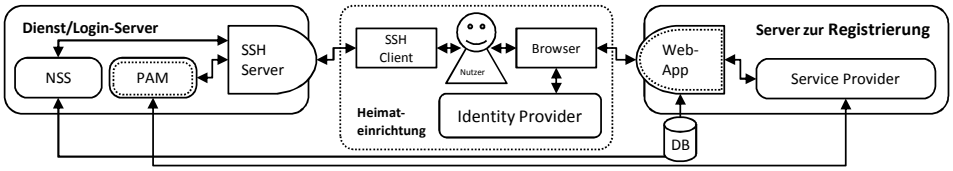

Abbildung 1: Proof-of-Concept: PAM/ECP-Ansatz mit Erweiterung (WebApp und PAM)

beispielsweise SSH-Schlüssel in das Home-Verzeichnis abgelegt werden, so dass sich der Zugang aus Nutzersicht wie jeder andere SSH-Zugang verhält.

5.2 AUP-Unterstützung durch getrennten Provisionierungsschritt (Web)

Als weiteren Vorteil der im vorigen Abschnitt beschriebenen Registrierung mittels eines Webportals erweist es sich, dass durch diese Dienstaktivierung durch den Nutzer eine direkte Interaktionsmöglichkeit hergestellt wird. An dieser Stelle ist es beispielsweise möglich, der Registrierung eine Anerkennung der für den konkreten Dienst gültigen AUP voranzustellen, ohne die keine Aktivierung des Dienstzugangs erfolgt. In ähnlicher Weise kann auch eine erneute Anerkennung der AUP durch den Nutzer erzwungen werden, indem das lokale Nutzerkonto mit einem entsprechenden Hinweis deaktiviert wird, der den Nutzer auffordert, sich erneut zu registrieren. Dies ist beispielsweise dann wünschenswert beziehungsweise notwendig, wenn sich die AUP seit der letzten Zustimmung des Nutzers geändert haben und dies dem Nutzer explizit zur Kenntnis gegeben werden soll.

5.3 Realisierung als Proof-of-Concept

Um diese Thematik weiter zu evaluieren, wurde ein Proof-of-Concept erstellt (vgl. Abbildung 1). Teil des Proof-of-Concept ist ein PAM, das als Enhanced Proxy agiert, um so einen transparenten Zugang mit einem unmodifizierten SSH-Client auf einem unmodifizierten SSH-Server zu ermöglichen. Es kommt das pam_python Modul zum Einsatz und die AuthN wird über ein Python Script realisiert. Das Script richtet eine PAOS-Anfrage (Reverse SOAP[7]) an einen Shibboleth SP, der für die ECP Nutzung konfiguriert ist. Auf diese Anfrage antwortet der SP mit einer SOAP-Anfrage, die das Script an den zuständigen ECP-Endpunkt des IdP schickt. Dabei kommt ein Suffix ähnlich einem Radius-Realm zum Einsatz. Dieses Suffix muss in einer Mapping-Tabelle mit einem dazugehörenden ECP-Endpunkt vorhanden sein. Aktuell unterstützt der Shibboleth IdP nur die ECP-Variante mit HTTP Basic Authentifikation. Bei dieser Variante wird der ECP-Endpunkt beim IdP mittels HTTP Basic geschützt. Es ist also Aufgabe des Scripts die vom Nutzer übermittelten Zugangsdaten mit der SOAP Anfrage an den ECP-Endpunkt zu übermitteln. Ist der Login erfolgreich, antwortet der IdP mit einer Assertion, die das Script wiederum an den SP

[7] http://www.w3.org/TR/soap/

weiterleitet. An dieser Stelle können nun vom SP evtl. notwendige AuthZ-Merkmale entgegengenommen und überprüft werden. Für diesen Proof-of-Concept musste lediglich ein PAM sowie eine Web-Anwendung implementiert werden. SSH-Clients und -Server, NSS sowie Shibboleth Identity und Service Provider können ohne Anpassung des jeweiligen Programmcodes in die vorgeschlagene Infrastruktur aufgenommen werden. Dies unterstreicht die Betriebsfähigkeit des Ansatzes.

5.4 Deprovisionierung: Aufstellen von Regeln (dienstabhängig)

Die Provisionierung von permanenten Nutzerkonten bringt jedoch auch Probleme mit sich. Insbesondere ist zunächst unklar, wie zu verfahren ist, wenn ein Nutzer das Zugangsrecht zum Dienst verliert, sei es etwa durch Ausscheiden aus der Hochschule oder beispielsweise aufgrund der Sperrung seines Accounts aus Sicherheitsgründen. Ist ein permanentes Nutzerkonto eingerichtet, das es zum Beispiels erlaubt, sich mit Hilfe von SSH-Schlüsseln zu authentifizieren, so ist es dem Dienst nicht ohne weiteres möglich, von entsprechenden Änderungen des Shibboleth-Nutzerprofils Kenntnis zu erlangen, da in diesem Fall überhaupt keine AuthN gegenüber dem Shibboleth IdP mehr erfolgt. Wie der Dienst möglichst zeitnah über diese Änderungen informiert wird, ist unklar. Denkbar wäre einerseits ein Push-Ansatz, bei dem der IdP alle ihm bekannten Dienste aktiv mit Änderungsmeldungen versorgt; ein derartiger Ansatz scheint aber bereits aus Komplexitätsgründen nicht ratsam, würde aber jedenfalls das lose gekoppelte föderale Konzept brechen, da jeder IdP doch wieder Informationen an alle Dienste verteilen müsste. Alternativ wäre ein Pull-Ansatz vorstellbar, bei dem jeder Dienst regelmäßig beim IdP anfragt, um Änderungen an den für ihn relevanten Identitäten zu erfahren. Hierbei ist derzeit noch unklar, ob Shibboleth/ECP bereits alle notwendigen technischen Mittel bereitstellt.

6 Zusammenfassung und Ausblick

Nach der Integration zahlreicher webbasierter IT-Dienste in föderative Verbünde wächst der Wunsch nach betriebsfähigen Integrationslösungen für nicht-webbasierte Dienste. Im vorliegenden Papier wurden Anforderungen an föderative Verfahren aufgestellt, die erfüllt werden müssen, um auch diese Dienste einbinden zu können. Moonshot, als ein derzeit entstehender, Radius-basierter Ansatz, und ein Ansatz zur Dienstintegration via PAM und ECP wurden vorgestellt und bewertet. Kontrastierend zu den zuvor aufgestellten Anforderungen wurden notwendige Erweiterungen für den für die Integrationsvorhaben vielversprechenderen zweiten Ansatz (PAM/ECP) identifiziert. Die technische Ausgestaltung dieser Erweiterungen wurde im Folgenden diskutiert und die Umsetzbarkeit in einem Proof-of-Concept belegt. Das so skizzierte bwIDM-Konzept passt mit der bisherigen Planung und prototypischen Umsetzung auf die obligatorischen Anforderungen.

Auch wenn die Anforderungen durch das skizzierte PAM/ECP-Konzept mit den diskutierten Erweiterungen weitestgehend erfüllt werden können, sind weitere Schritte in Rich-

tung einer betriebsfähigen Lösung zu gehen. Neben der zu führenden Diskussion über die Deprovisionierung (Push- oder Pull-Ansätze), sind Richtlinien zur Nutzung der bwIDM-Föderation zu definieren. Ferner ist der derzeit auf SSH-Zugänge beschränkte Proof-of-Concept auf die in Kapitel 1 genannten CIFS und NFS zu erweitern. Abschließend sollte diskutiert werden, wie dem SAML-Paradigma Rechnung getragen werden kann, so dass einem SP der Zugriff auf die Credentials eines Nutzers stets verwehrt werden kann.

7 Weitere Autoren und Danksagung

An der Entstehung dieses Papiers waren neben den oben genannten Personen folgende Autoren beteiligt: Tobias Dussa und Sebastian Labitzke (Editor) vom Karlsruher Institut für Technologie (KIT), Jacob Becker, Markus Grandpre, Michael Längle und Daniel Scharon von der Universität Konstanz, Harald Däubler und Vladimir Nikolov von der Universität Ulm sowie Markus Klein von der Universität Freiburg. Ein besonderer Dank gebührt dem Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg (MWK), dem Projektverantwortlichen Prof. Dr. Hannes Hartenstein sowie allen Projektteilnehmern aller Landesuniversitäten in Baden-Württemberg.

Literatur

[ABV+04] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson und H. Levkowetz. Extensible Authentication Protocol (EAP) - RfC 3748, 2004.

[DW98] J. Danielsson und A. Westerlund. Heimdal: an independent implementation of Kerberos 5. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, ATEC '98, pages 34–34, Berkeley, CA, USA, 1998. USENIX.

[Eck09] C. Eckert. *IT-Sicherheit - Konzepte, Verfahren, Protokolle (6.A.)*. Oldenbourg, 2009.

[HCH+05] J. Hughes, S. Cantor, J. Hodges, F. Hirsch, P. Mishra, R. Philpott und E. Maler. Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, 2005.

[HVS10] J. Howlett, Nordh V. und W. Singer. Deliverable DS3.3.1: eduGAIN service definition and policy Initial Draft. Technical report, GÉANT, 2010.

[Lin93] J. Linn. Generic Security Service Application Program Interface - RfC 1508, 1993.

[Lin96] J. Linn. The Kerberos Version 5 GSS-API Mechanism - RfC 1964, 1996.

[MRW+08] M. Milinovic, J. Rauschenbach, S. Winter, L. Florio, D. Simonsen und J. Howlett. Deliverable DS5.1.1: eduroam Service Definition and Implementation Plan. Technical report, GÉANT2, 2008.

[NYHR05] C. Neuman, T. Yu, S. Hartman und K. Raeburn. The Kerberos Network Authentication Service (V5) - RfC 4120, 2005.

[RWRS00] C. Rigney, S. Willens, A. Rubens und W. Simpson. Remote Authentication Dial In User Service (RADIUS) - RfC, 2000.

[Sam96] V. Samar. Unified login with pluggable authentication modules (PAM). In *Proceedings of the 3rd ACM conference on Computer and communications security*, CCS '96, pages 1–10, New York, NY, USA, 1996. ACM.

[WMVW11] S. Winter, M. McCauley, S. Venaas und K. Wierenga. TLS encryption for RADIUS - draft-ietf-radext-radsec-09, 2011.