FUDIS-Plugins
für den Shibboleth Identity Provider

08.10.2024 – 81. DFN Betriebstagung – AAI-Forum

Steffen Hofmann, Freie Universität Berlin, FUB-IT

Gliederung

- **FUDIS, was ist das?**
- **Welche Plugins gibt es?**
- **fudiscr-Plugin im Detail**
- **Zukünftige Plugins**
- **Fragen und Diskussion**

FUDIS, was ist das?

- FUDIS - FU Directory and Identity Service
- Identitätsmanagementsystem der Freien Universität Berlin
- 2005 als Dienst aus SQL-Datenbank mit PHP-Verwaltungsoberfläche und LDAP-Verzeichnisdienst gestartet
- Inzwischen große Software-Architektur für diverse personenbezogene Identitäten
- Ein paar wenige Zahlen:
  - ca. 60 Tausend aktive Accounts
  - ca. 10 Millionen LDAP-Anfragen am Tag
  - ca. 120 interne SAML Service Provider

FUDIS-Team

- ca. 5,5 VZÄ
- zuständig auch für
  - Bewerbungssysteme der Studierendenverwaltung
  - Campuscard
  - Deutschlandstipendium
  - … u.a.
- Diverse Projekte
  - Landes-IDM-Projekt
  - eduMFA
  - … u.a.

Motivation für Shibboleth IdP FUDIS-Plugins

- Eigener Bedarf
- Schulungsunterstützung bei DFN-Workshops
- Amtshilfe
- Wenn andere sagen, was sie brauchen, bekomme ich eventuell neue Ideen für meine eigene Implementierung
- Vision: Geld sparen und Sicherheit erhöhen mit guter Open Source Software im Wissenschaftsumfeld
  Jeder sollte seinen Beitrag für die Community leisten
  … wenn möglich!

Übersicht FUDIS-Plugins

Für Authentifizierungen am Bibliothekssystem von ExLibris:

- fudis-shibboleth-idp-plugin-alma

Für DSGVO-Informationen in Audit-Einträgen:

- fudis-shibboleth-idp-plugin-dsgvo

Für die tokenbasierte Authentifizierungen:

- fudis-shibboleth-idp-plugin-fudiscr

Plugin: ‚fudis-shibboleth-idp-plugin-alma‘

End of life

→ Wurde aufgrund mangelnder Nachfrage wieder eingestellt.

Plugin: ‚fudis-shibboleth-idp-plugin-dsgvo‘

- Bestandteil des „Beispiels für eine EU-DSGVO-konforme Konfiguration des User Consent Moduls“
- siehe https://doku.tid.dfn.de/de:shibidp:config-consent-dsgvo von Wolfgang Pempe, DFN Verein
- Definition eines Attributs, welches die Rechtsgrundlage für die Datenweitergabe widerspiegelt
- Das Plugin extrahiert dieses Attribut und stellt es unter dem Pattern %DSGVO für das Audit-Log zur Verfügung.

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Funktion

- Weiterer Faktor für die Multifaktor-Authentifizierung (MFA)
- Token basierte Authentifizierungen nach dem Challenge-Response-Pattern
- Keine Funktionen zur Verwaltung von Token
- Keine speziellen Algorithmen zur Validierung von Token
- Über generisches `ChallengeResponseClientInterface` kann ein Token-System angebunden werden; aktuell:
  - eduMFA (Standard seit der Version 2.1.0) → https://edumfa.io
  - privacyIDEA
  - Fortinet - FortiAuthenticator (mit Plugin-Erweiterung der DAASI GmbH)

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Dokumentation und Verbreitung

Dokumentation:

- https://doku.tid.dfn.de/de:shibidp:plugin-fudiscr

Verbreitung:

- hohe Verwendung bei deutschen Hochschulen und wissenschaftlichen Einrichtungen (>200)
- steigende Verwendung bei europäischen Hochschulen (>20)

(genaue Zahlen sind schwer zu messen)

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Unterstützte Token-Typen

Aus eduMFA und privacyIDEA:

- Email
- HOTP Token
- Indexed Secret Token
- mOTP Token
- Paper Token (PPR)
- Questionnaire Token
- Registration
- Remote
- SMS Token
- Spass
- TAN Token
- TOTP
- WebAuthn
- Yubico
- Yubikey

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Konfigurationselemente im Details

- Filterung von Token
- Funktion fudiscr.UserHasTokenPredicate
- Vorselektierte Token
- Realm-Transformation
- Subject Customizers
- Subject Canonicalization

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Filterung von Token - Mögliche Szenarien

- Beim Single Sign-On sollen nur bestimmte Token-Typen zum Einsatz kommen → z.B. sind SMS-Token nur für spezielle Admin-Systeme gedacht.
- Recovery Codes sollen nur bei einem Service Provider für die Entsperrung von anderen Token-Verfahren funktionieren.
- Nutzer*innen sollen erst zur Zweifaktor-/Multifaktor-Authentifizierung gezwungen werden, wenn sie den TOTP-Token aus ihrem Brief (QR-Code) das erste Mal bestätigt haben.

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Filterung von Token - Varianten

- anhand von Token-Eigenschaften, z.B.
  - ID/Seriennummer
  - Typ
  - Status

- anhand von Eigenschaften der SAML- oder OpenID-Anfrage, z.B.
  - entityID des Service Providers/der Relying Party
  - Authentication Context Class
  - Client-IP-Adresse

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Filterung von Token via Regex

Filterung von Token-Typen:

```
fudiscr.filter_tokens.type_exclude_regex= \
^(sms|web_authn)$
```

Filterung von IDs/Seriennummern:

```
fudiscr.filter_tokens.id_exclude_regex= \
^(OATH|TOTP).+$
```

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Filterung von Token via Scripted-Bean

```
<![CDATA[
   // default is to keep the token
   exclude = false;

   relyingPartyId = profileContext.
   getSubcontext("net.shibboleth.profile.context.RelyingPartyContext").
   getRelyingPartyId();

   // if the service provider is not 'https://portal.local/shibboleth‘,
   // tokens that are of type 'registration_code' are excluded
   if (!relyingPartyId.equals("https://portal.local/shibboleth") &&
       token.getType().toString().equalsIgnoreCase("registration_code")) {
      exclude = true;
   }
   exclude;
]]>
```

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Filterung von Token via Regex-Bean

```
<bean class="de.zedat.fudis.shibboleth.idp.plugin.authn.
            fudiscr.function.impl.
            RegexExcludeTokenPropertiesPredicate"
      c:propertyName="rollout_state"
      c:excludePattern="^(clientwait|verify)$"/>
```

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Funktion fudiscr.UserHasTokenPredicate

- kann genutzt werden, um zu entscheiden, ob Nutzer*innen MFA-enabled sind
- Beliebt für MFA-Rollout-Phase
- kann auch Token-Filter nutzen

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr' Vorselektierte Token

- Einrichtungen und/oder Nutzer*innen bevorzugen in der Regel ein Token-Verfahren
- Auf Wunsch können bevorzugte Token-Verfahren immer direkt angesteuert werden
- Ermittlung bevorzugter Token-Verfahren durch Token-Filterung
- Drei Resultate der Filterung:
  1. Token-Liste ist leer → dann Ausgangsliste
  2. Filterung ergibt (reduzierte) Liste mit mehreren Möglichkeiten → Auswahlliste
  3. Filterung reduziert Auswahlmöglichkeiten auf genau eine → direkte Ansteuerung des Verfahrens

→ Demo

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Realm-Transformation

- Nutzer*innen eventuell über mehrere Datenquellen verteilt.
- Bei API-Anfrage gegen eduMFA/privacyIDEA muss Realm-Zuordnung mitgegeben werden, wenn es nicht der Default-Realm ist.
- Abfolge zur Realm-Ermittlung:
  - `CanonicalUsernameLookupStrategy` ermittelt Username
  - Mit `fudiscr.username_realm_split_regex` (@) Realm-Extraktion
  - Mit Strategien (scripted) in `fudiscr.xml` können anhand bisheriger Informationen aus vorherigen Authentifizierungen Realms ermittelt und/oder transformiert werden.
  - Wenn Realm noch `null`, dann `fudiscr.default_users_realm`

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr' Subject Customizers

- REFEDS Assurance Framework, siehe
  - https://doku.tid.dfn.de/de:aai:assurance_idp
  - https://download.aai.dfn.de/ws/2022/refeds_assurance_suite.pdf von Wolfgang Pempe, DFN Verein
- Transport von MFA-Profilen über AuthnContextClassRef in SAML Assertion oder OIDC acr claim
- Abbildung als `Principals` des `Subject` … und jetzt wird es kompliziert
- Ausführliche Beschreibung in der Dokumentation: https://doku.tid.dfn.de/de:shibidp:plugin-fudiscr#subject_customizers

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr' Subject Canonicalization

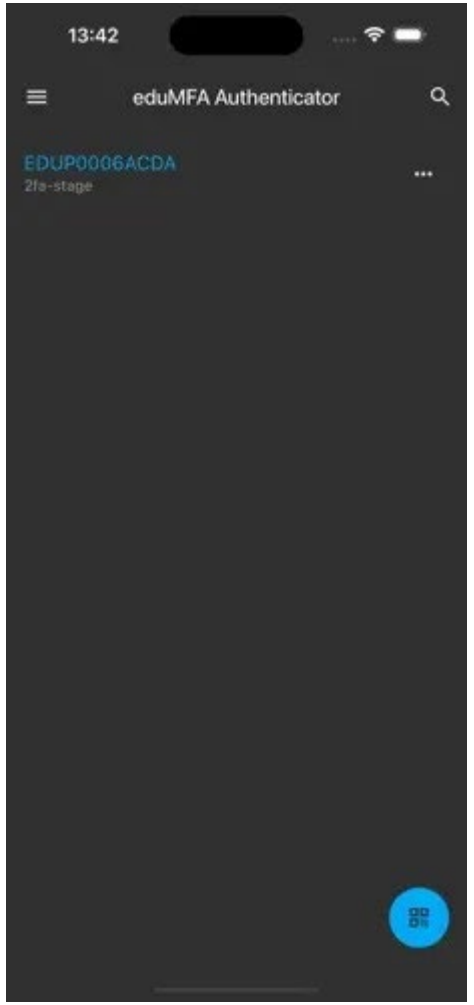
- Siehe https://shibboleth.atlassian.net/wiki/spaces/IDP5/pages/3199512211/SubjectCanonicalization
- In `subject-c14n.xml` kann die Referenz `<ref bean="c14n/fudiscr"/>` aufgenommen werden
- Spezieller `Principal` enthält ID/Seriennummer des Token
- `idp.c14n.fudiscr.lowercase=false`
- `idp.c14n.fudiscr.uppercase=false`
- `idp.c14n.fudiscr.trim=true`

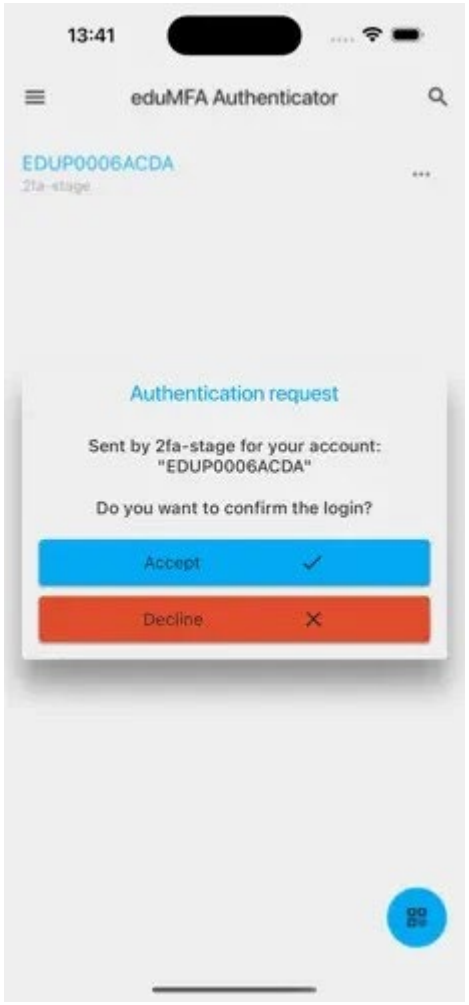
Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ fudispasskeys

- Eigener Authn-Flow
- Direkte Authentifizierung mit einem Sicherheitsschlüssel/Passkeys
- Lässt sich in die Login-Seite für die Authentifizierung mit Benutzernamen und Passwort integrieren
- Nur für eduMFA!

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Zukünftige Entwicklung

- eduMFA Authenticator
  - push Verfahren
- Ab fudiscr Version 2.2.0
  - Aktuell Testphase
  - Release bis 15.11.2024





Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr‘ Quellcode

- Hochschulen und andere öffentliche Einrichtungen erhalten auf Anfrage Zugriff auf den Quellcode
  - E-Mail an fudis@fu-berlin.de
- Apache 2.0 Lizenz wird es werden!
- Wir bitten um Geduld – es gibt Gründe

Plugin: ‚fudis-shibboleth-idp-plugin-fudiscr' Fortinet-Erweiterung

- Unterstützte Token-Typen
  - FortiToken (Mobile und Hardware)
  - SMS
  - E-Mail
- kein FIDO2, da die API das aktuell nicht unterstützt
- Apache 2.0 Lizenz - https://gitlab.daasi.de/shibboleth-identity-provider/shibboleth-idp-plugin-authn-fudiscr-fortinetclient
- Bereitgestellt von DAASI International mit der Option eines Softwarepflegevertrages
  - Anfragen bitte an info@daasi.de

Zukünftige Plugins
MFA-Rollout

- Grundsätzlich wird Gerätebindung bei Sicherheitsschlüssel/Passkeys nicht verstanden.
- MFA aktuell nur für neue Studierende der FU Berlin seit dem Sommersemester 2024 (separate Rollout-Strategie für Mitarbeitende)
- Für früher immatrikulierte Studierende im Service Portal Einrichtung von Token auf Basis des ersten Faktors
- Schwierige SAML-Abfolge, wenn beim Identity Provider für die MFA-Einrichtung auf einen anderen Service Provider verwiesen wird
- Plugin (POST Authn Flow), das eine Token-Einrichtung für bestimmte Token-Typen direkt im IdP erlaubt, könnte Situation verbessern

Zukünftige Plugins
SSF

- Kooperation mit Apple Deutschland begonnen (erste Gespräche)
- Anbindung des Shibboleth Identity Providers mittels OpenID für Managed Apple Ids
  - z.B. im Apple School Manager

- Aktuelle Herausforderung:
  - Apple verlangt in Teilen die Unterstützung das Shared Signals Framework (SSF), https://openid.net/wg/sharedsignals/
  - der Shibboleth IdP unterstützt SSF nicht und es gibt auch keine aktuellen Planungen dazu
  - neues Plugin könnte diese Lücke schließen

Fragen und Diskussion

Vielen Dank!

Fragen?

Kontakt: steffen.hofmann@fu-berlin.de