Sakhr Al-absi

Security Assertion Markup Language 2.0 (SAML 2.0)

HTW Berlin

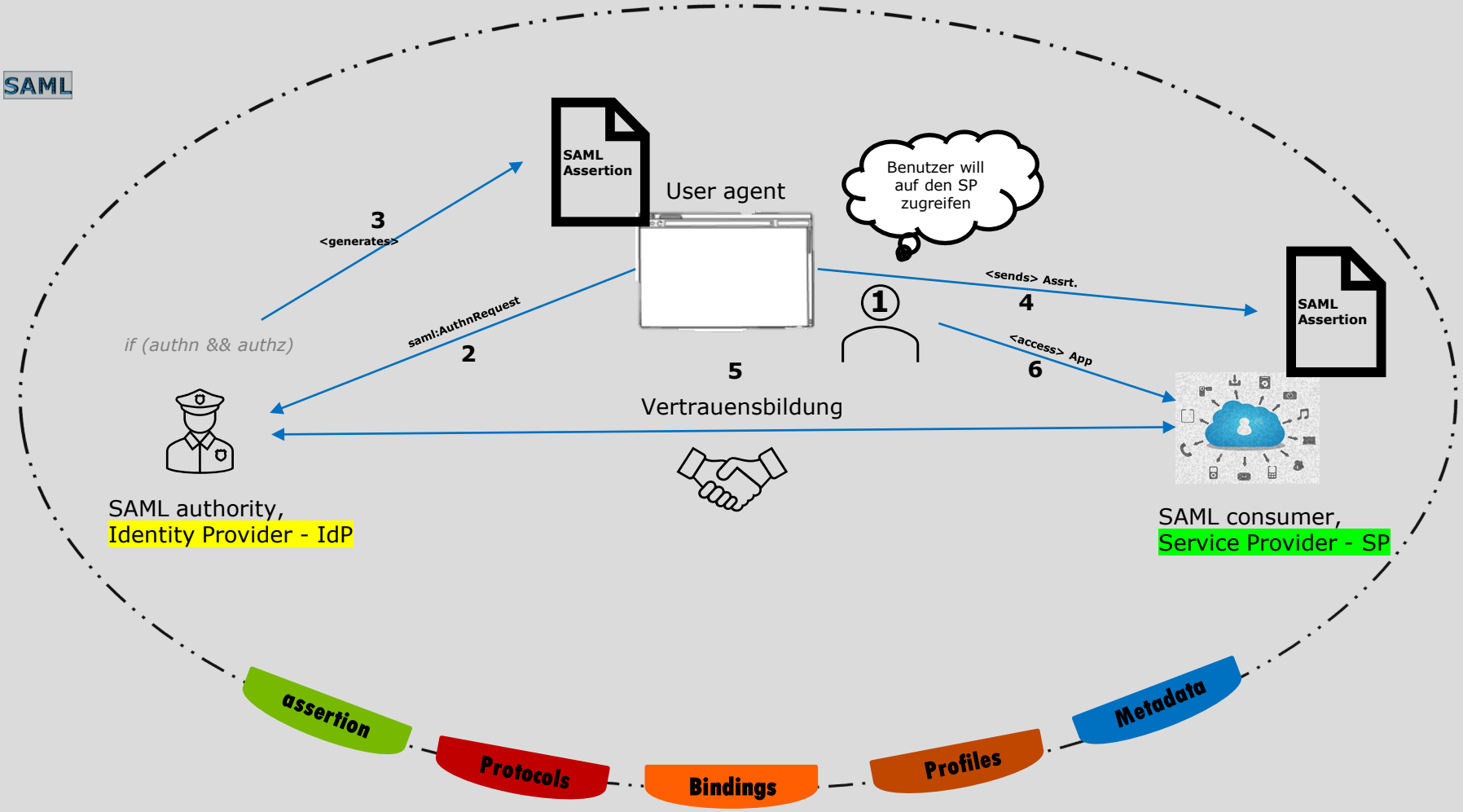
09.03.2021

Hochschule für Technik und Wirtschaft Berlin
University of Applied Sciences

Was ist SAML?

Security Assertion Markup Language, kurz SAML, ist ein quelloffenes Framework auf XML-Basis zum Austauschen von Authentifizierungs- und Autorisierungsinformationen.

Im Laufe der Zeit wurden diverse Anpassungen vorgenommen, die in den überarbeiteten Versionen SAML 2.0 und 2.1 (auch 1.1) wiederzufinden sind.


SAML Assertion
User agent
Benutzer will auf den SP zugreifen
3
<generates>
saml:AuthnRequest
2
if (authn && authz)
1
<sends> Assrt.
4
SAML Assertion
<access> App
6
5
Vertrauensbildung
SAML authority,
Identity Provider - IdP
SAML consumer,
Service Provider - SP
assertion
Protocols
Bindings
Profiles
Metadata

Assertion

Assertionen

Assertionen sind Aussagen des Identitätsanbieters (engl.Identity Provider - IdP) über den Principal (User). Zum Beispiel die E-Mail-Adresse des Principals und / oder Gruppen / Rollen, denen der Principal zugeordnet sein kann. Assertions werden vom Dienstanbieter (engl. Service Provider – SP) verwendet, um Sitzungen für einen Principal zu erstellen und zu konfigurieren[1].

Mit anderen Worten, assertions definieren, welche Identitätsinformationen über den Principal vorliegen von einem Identitätsanbieter an einen Dienstanbieter übermittelt werden.

Über Authentication Statements informiert der Identity-Provider die Anwendung darüber, dass der Benutzer authentifiziert wurde. Ferner liefert dieser Aussagentyp in einer Assertion auch Informationen darüber, wann die Authentifizierung stattfand und welche Methode hierfür zur Anwendung kam.

Bei Attribute Statements handelt es sich um Attribute, die mit dem jeweiligen Benutzer verknüpft sind und so der Anwendung über das entsprechende SAML-Token mitgeteilt werden können.

Wenn Authorization Decision Statements in einer SAML-Assertion enthalten sind, hat der jeweilige Benutzer entweder Zugriff auf spezifische Ressourcen erhalten oder ihm wurde der Zugriff auf spezifische Ressourcen verweigert.

- SAML 2.0 definiert 3 Assertion Aussagen:
  - Authentication Assertion
  - Attribute Assertion
  - Authorization Decision Assertion
- Alle SAML-definierte Aussagen sind mit einem Subject assoziert.
- Bearer Assertion – **am wichtigsten**

```
<saml:Assertion
   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   ID="_d71a3a8e9fcc45c9e9d248ef7049393fc8f04e5f75"
   Version="2.0"
   IssueInstant="2004-12-05T09:22:05Z">
   <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
   <ds:Signature
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
   <saml:Subject>
     <saml:NameID
       Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
       3f7b3dcf-1674-4ecd-92c8-1544f346baf8
     </saml:NameID>
     <saml:SubjectConfirmation
       Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
       <saml:SubjectConfirmationData
         InResponseTo="aaf23196-1773-2113-474a-fe114412ab72"
         Recipient="https://sp.example.com/SAML2/SSO/POST"
         NotOnOrAfter="2004-12-05T09:27:05Z"/>
     </saml:SubjectConfirmation>
   </saml:Subject>
   <saml:Conditions
     NotBefore="2004-12-05T09:17:05Z"
     NotOnOrAfter="2004-12-05T09:27:05Z">
     <saml:AudienceRestriction>
       <saml:Audience>https://sp.example.com/SAML2</saml:Audience>
     </saml:AudienceRestriction>
   </saml:Conditions>
   <saml:AuthnStatement
     AuthnInstant="2004-12-05T09:22:00Z"
     SessionIndex="b07b804c-7c29-ea16-7300-4f3d6f7928ac">
     <saml:AuthnContext>
       <saml:AuthnContextClassRef>
         urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
       </saml:AuthnContextClassRef>
     </saml:AuthnContext>
   </saml:AuthnStatement>
   <saml:AttributeStatement>
     <saml:Attribute
       xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
       x500:Encoding="LDAP"
       NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
       Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
       FriendlyName="eduPersonAffiliation">
       <saml:AttributeValue
         xsi:type="xs:string">member</saml:AttributeValue>
       <saml:AttributeValue
         xsi:type="xs:string">staff</saml:AttributeValue>
     </saml:Attribute>
   </saml:AttributeStatement>
</saml:Assertion>
```

Element, which contains the unique identifier of the IDP

Element, which contains an integrity-preserving digital signature (not shown) over the <saml:Assertion> element

Element, which identifies the authenticated principal (but in case the identity of the principal is hidden behind an opaque transient identifier, for reasons of privacy)

Element, which gives the conditions under which assertion is to be considered valid.

Element, which describes the act of authentication at the identity provider.

Element, which asserts a mutli-valued attribute associated with the authenticated principal.

Metadata

METADATA

URN:-ASIS:NAMES:SAML:1.1:NAMEID-FORMAT:UNSPECIFIED

URN:-ASIS:NAMES:SAML:1.1:NAMEID-FORMAT:EMAILADDRESS

URN:-ASIS:NAMES:SAML:2.0:NAMEID-FORMAT:PERSISTENT

URN:-ASIS:NAMES:SAML:2.0:NAMEID-FORMAT:TRANSIENT

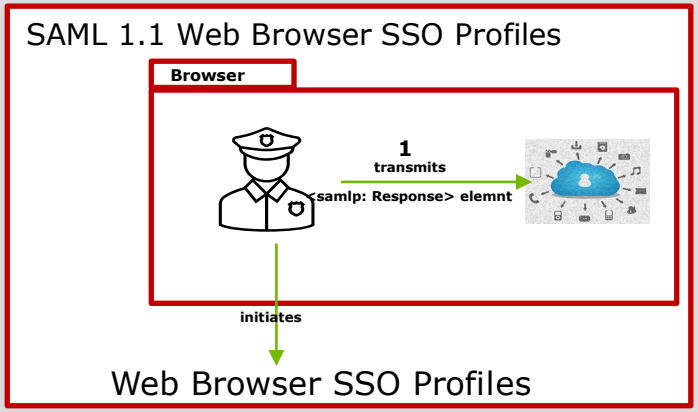
URN:-ASIS:NAMES:SAML:1.1:NAMEID-FORMAT:X509SUBJECTNAME

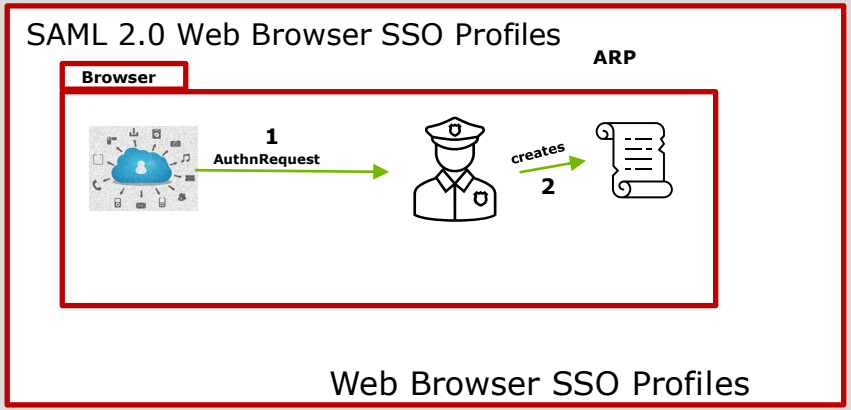
<DS:X509CERTIFICATE>
MIFFJCCAV7HONVNAO32BBOBVO22BBONNOOAONFOANO
HH3B0AFX92LAJDLF0221SLZKWHKSODJJ2JJ8LNFSK/LLDFF3E
</DS:X509CERTIFICATE>

ENTITYID="HTTPS://idp.example.com/metadata/sp.xml"

Protocols

1. Assertion Query and Request Protocol
2. Authentication Request Protocol (ARP1) → das wichtigste
3. Artifact Resolution Protocol
4. Name Identifier Management Protocol
5. Single Logout Protocol
6. Name Identifier Mapping Protocol






```
<samlp:AuthnRequest
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="aaf23196-1773-2113-474a-fe114412ab72"
  Version="2.0"
  IssueInstant="2004-12-05T09:21:59Z"
  AssertionConsumerServiceIndex="0"
  AttributeConsumingServiceIndex="0">
  <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
  <samlp:NameIDPolicy
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"/>
</samlp:AuthnRequest>
```

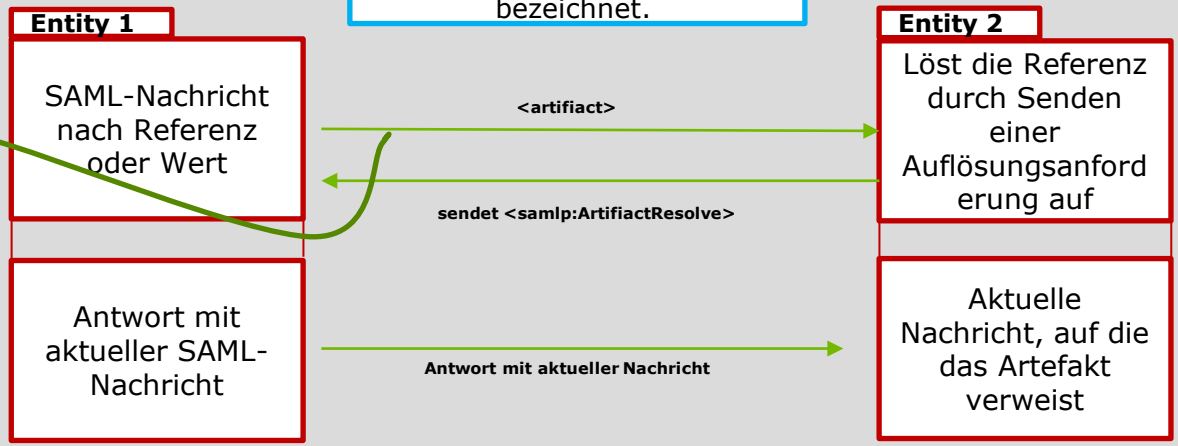
Protocols

1. Assertion Query and Request Protocol
2. Authentication Request Protocol (ARP1)
3. Artifact Resolution Protocol (ARP2) → Auch von Bedeutung
4. Name Identifier Management Protocol
5. Single Logout Protocol
6. Name Identifier Mapping Protocol

```
<samlp:ArtifactResolve
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="_cce4ee769ed970b501d680f697989d14"
  Version="2.0"
  IssueInstant="2004-12-05T09:21:58Z">
  <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
  <!-- an ArtifactResolve message SHOULD be signed -->
  <ds:Signature
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
  <samlp:Artifact>AAQAAMh48/1oXIM+sDo7Dh2qMp1HM4IF5DaRWmDj6RdUmllwn9jJHyEgIi8=</samlp:Artifact>
</samlp:ArtifactResolve>
```

Ein Referenz auf eine SAML-Nachricht wird als Artefakt bezeichnet.

Eine SAML-Nachricht wird von einer Entität zu einer anderen als Referenz oder Wert übertragen



Bindings

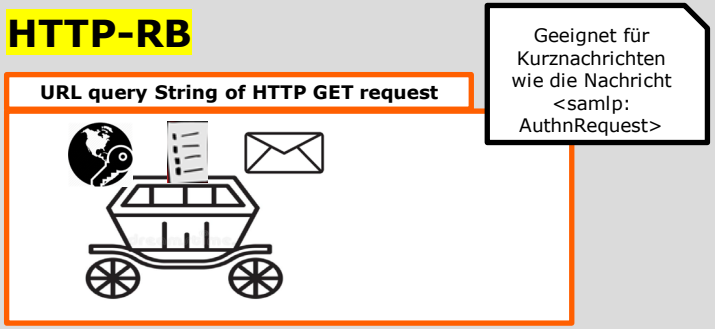
Bindungen sind das Format, in dem Daten zwischen SP und IdP übertragen werden.

Arten von Bindungen:

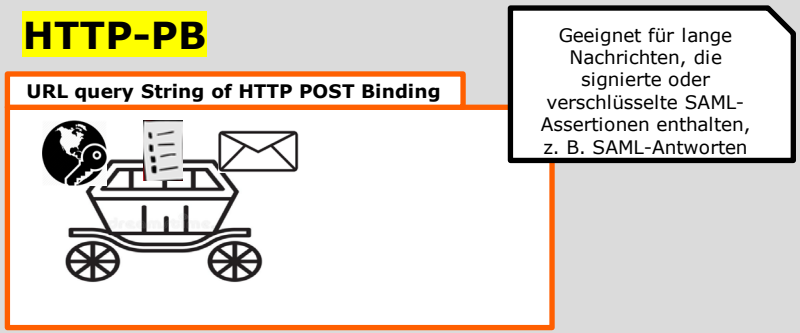
1. SAML SOAP Binding (based on SOAP 1.1)
2. Reverse SOAP (PAOS) Binding
3. HTTP Redirect Binding (HTTP-RB)
4. HTTP POST Binding (HTTP-PB)
5. HTTP Artifact Binding
6. SAML URI Binding

Die beiden beliebtesten sind HTTP Redirect Binding und HTTP POST Binding.

HTTP-RB



HTTP-PB



SAML req or res transmitted via HTTP redirect have a SAMLRequest or SAMLResponse query string parameter.

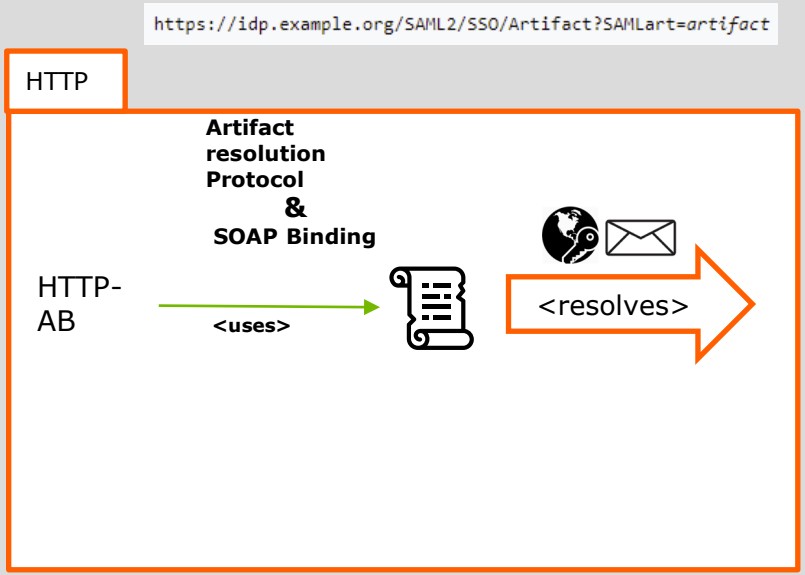
Bindings

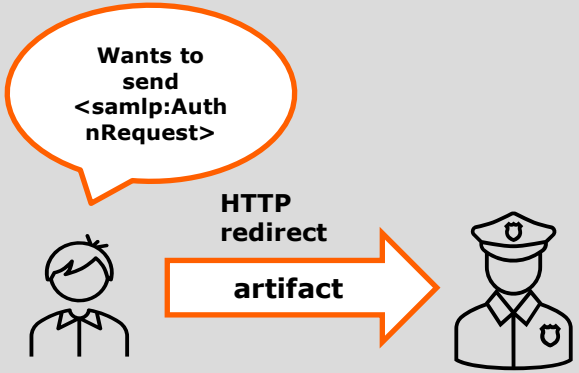
== transmits

HTTP-AB – Artifact Binding

Der HTTP-AB verwendet das Artifact Resolution Protocol und die SAML SOAP Binding (über HTTP), um eine SAML-Nachricht per Referenz aufzulösen. Betrachten Sie das folgende spezifische Beispiel:

- **Angenommen, ein Dienstanbieter möchte eine *<samlp: AuthnRequest>* -Nachricht an eine IdP senden.**
- **Zunächst überträgt der SP ein Artefakt über eine HTTP-Redirect an die IdP**





https://idp.example.org/SAML2/SSO/Artifact?SAMLart=artifact

Bindings

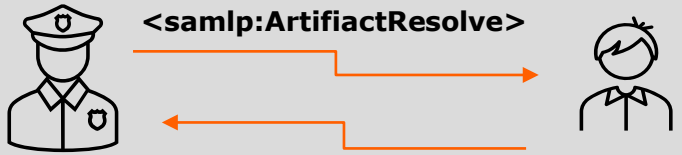
== sends

== returns

HTTP-AB – Artifact Binding

Als nächstes sendet die IdP eine *<samlp: ArtifactResolve>* -Anforderung (wie die zuvor gezeigte ArtifactResolveRequest) über einen Rückkanal direkt an den SP. Schließlich gibt der SP ein *<samlp: ArtifactResponse>* -Element zurück, das die referenzierte *<samlp: AuthnRequest>* -Nachricht enthält:

```
<samlp:ArtifactResponse
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  ID="_d84a49e5958803dedcff4c984c2b0d95"
  InResponseTo="_cce4ee769ed970b501d680f697989d14"
  Version="2.0"
  IssueInstant="2004-12-05T09:21:59Z">
  <!-- an ArtifactResponse message SHOULD be signed -->
  <ds:Signature
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
  <samlp:Status>
    <samlp:StatusCode
      Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>
  <samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="_306f8ec5b618f361c70b6ffb1480eade"
    Version="2.0"
    IssueInstant="2004-12-05T09:21:59Z"
    Destination="https://idp.example.org/SAML2/SSO/Artifact"
    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
    AssertionConsumerServiceURL="https://sp.example.com/SAML2/SSO/Artifact">
    <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
    <samlp:NameIDPolicy
      AllowCreate="false"
      Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"/>
  </samlp:AuthnRequest>
</samlp:ArtifactResponse>
```



<samlp:ArtifactResponse> enthält <samlp:AuthnRequest>

Notiz

Natürlich kann der Fluss auch in die andere Richtung gehen, dh die IdP kann ein Artefakt ausgeben, und tatsächlich ist dies häufiger.

Profiles

Die Hauptverwendung in saml 2.0 ist immer noch Webbrowser-SSO, aber der Umfang von SAML 2.0 ist breiter als in früheren Versionen von SAML, wie in den folgenden Profilen:

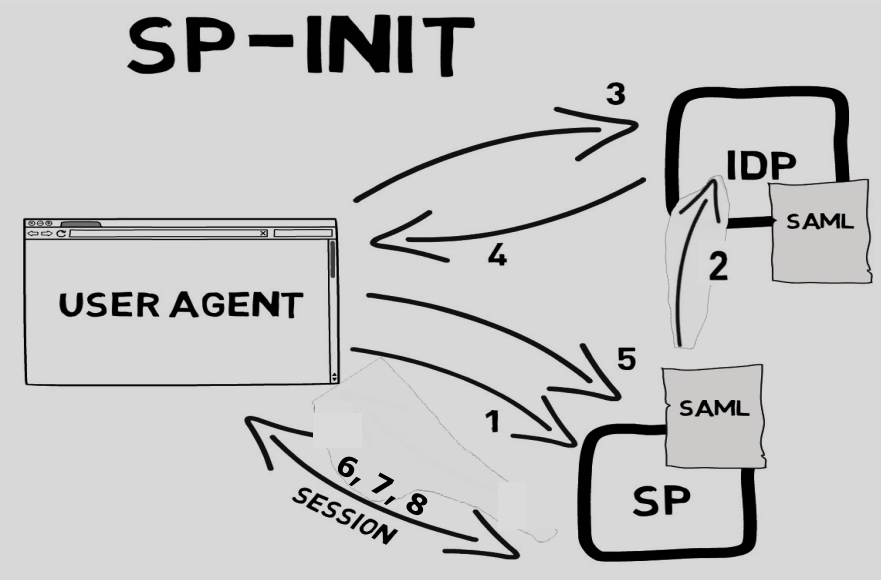
1. SSO Profiles
    - Web Browser SSO Profile → das wichtigste
    - Enhanced Client or Proxy (ECP) Profile
    - Identity Provider Discovery Profile
    - Single Logout Profile
    - Name Identifier Management Profile

2. Artifact Resolution Profile
3. Assertion Query/Request Profile
4. Name Identifier Mapping Profile
5. SAML Attribute Profile
    - Basic Attribute Profile
    - X 500/LDAP Attribute Profile
    - UUID Attribute Profile
    - DCE PAC Attribute Profile
    - XACML Attribute Profile

Profiles

So kann der Benutzer den Authentifizierungsablauf initiieren.

Wir haben zwei Hauptmethoden für die Initiierungder Fluss der Authentifizierung:
1. Der erste wird als SP-initiierter (SP-INIT) Fluss bezeichnet:



6. Redirect to the target Resource
7. Request the target Resource at the SP again
8. Respond with requested resource

1. Request the target resource at the SP

```
https://sp.example.com/myresource
```

2. Redirect to IdP SSO Service

```
302 Redirect
Location: https://idp.example.org/SAML2/SSO/Redirect?SAMLRequest=request&RelayState=token
```

```
<samlp:AuthnRequest
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="identifier_1"
  Version="2.0"
  IssueInstant="2004-12-05T09:21:59Z"
  AssertionConsumerServiceIndex="0">
  <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
  <samlp:NameIDPolicy
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"/>
</samlp:AuthnRequest>
```

3. Request the SSO Service at the IdP

```
GET /SAML2/SSO/Redirect?SAMLRequest=request&RelayState=token HTTP/1.1
Host: idp.example.org
```

4. Respond with an XHTML Form

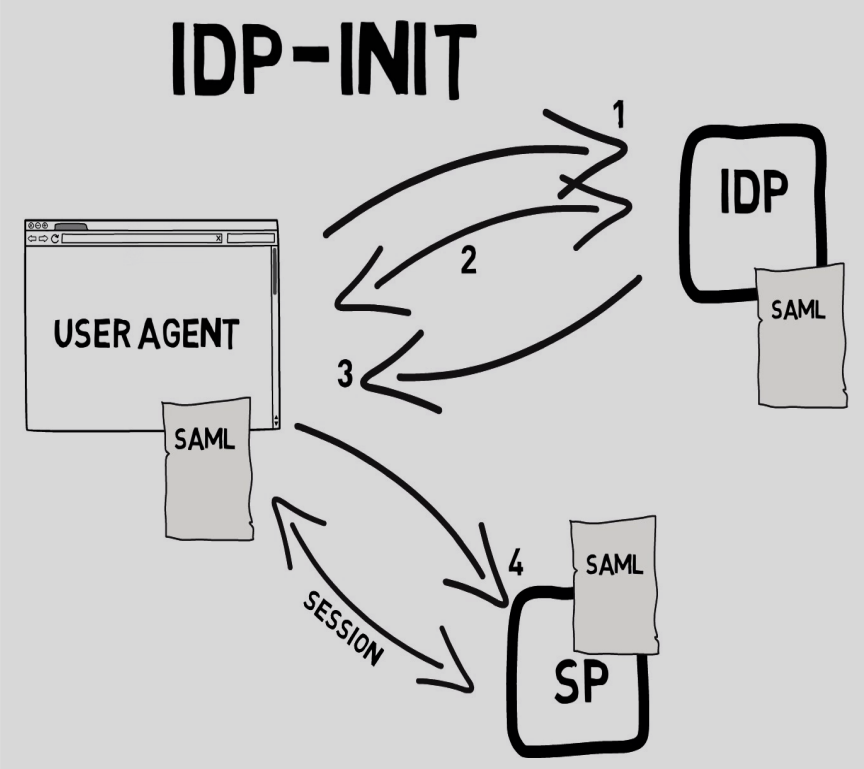
```
<form method="post" action="https://sp.example.com/SAML2/SSO/POST" ...>
  <input type="hidden" name="SAMLResponse" value="response" />
  <input type="hidden" name="RelayState" value="token" />
  ...
  <input type="submit" value="Submit" />
</form>
```

5. Request the Assertion Consumer Service at the SP

```
POST /SAML2/SSO/POST HTTP/1.1
Host: sp.example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnn

SAMLResponse=response&RelayState=token
```

Profiles



2. **Der zweite wird als IDP-initiierter (IDP-INIT) Fluss bezeichnet:**

   1. Benutzer greift auf den IdP
   2. Benuzter authentifiziert
   3. Nach 2) eine SAML-Assertion ist generiert mit dem User agent
   4. Die Assertion wird an SP gesendet (POST-MESSAGE)

Metadata

METADATA

URN:-ASIS:NAMES:SAML:1.1:NAMEID-FORMAT:UNSPECIFIED

URN:-ASIS:NAMES:SAML:1.1:NAMEID-FORMAT:EMAILADDRESS

URN:-ASIS:NAMES:SAML:2.0:NAMEID-FORMAT:PERSISTENT

URN:-ASIS:NAMES:SAML:2.0:NAMEID-FORMAT:TRANSIENT

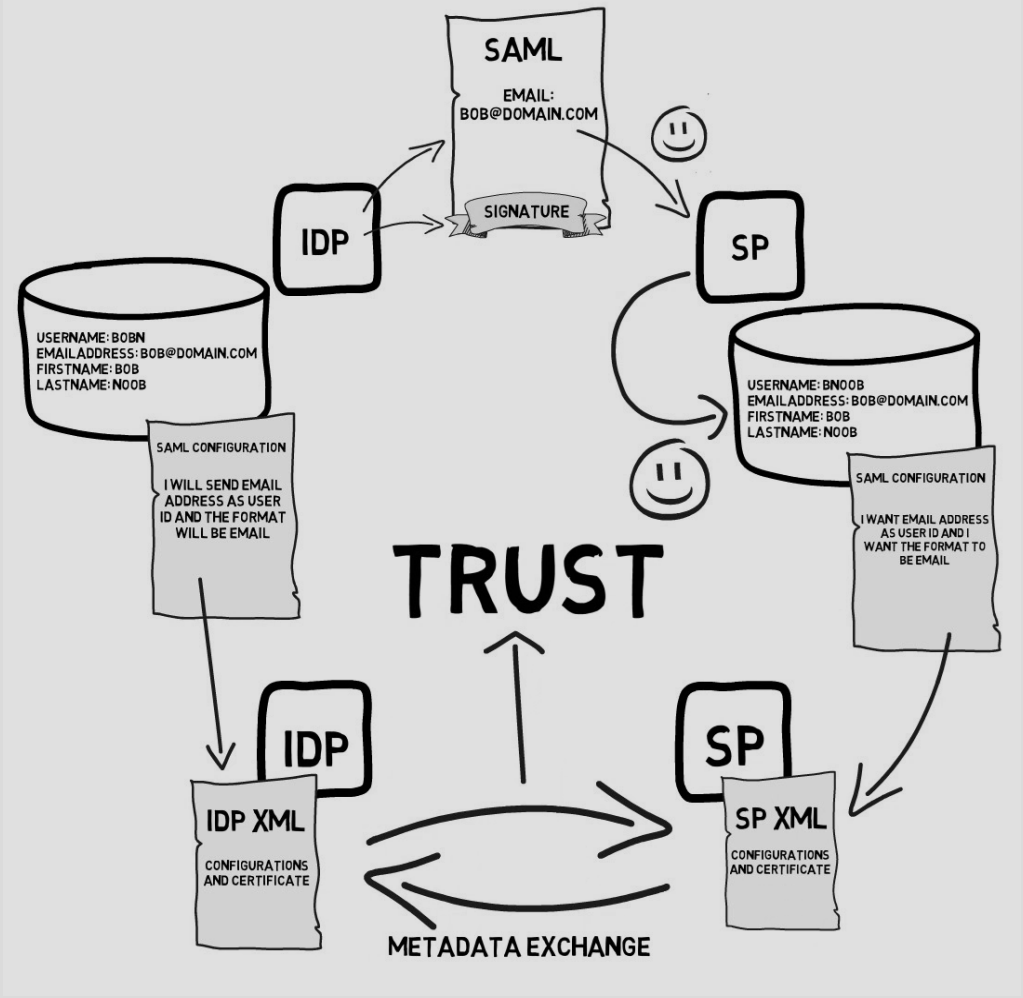
URN:-ASIS:NAMES:SAML:1.1:NAMEID-FORMAT:X509SUBJECTNAME

<DS:X509CERTIFICATE>
MIFFJCCAV7HONVNAO32BBOBVO22BBONNOOAONFOANO
HH3B0AFX92LAJDLF0221SLZKWHKSODJJ2JJ8LNFSK/LLDFF3E
</DS:X509CERTIFICATE>

ENTITYID="HTTPS://idp.example.com/metadata/sp.xml"

URN: 0 ASIS:NAMES:TC:SAML:2.0:BINDING:HTTP-POST
URN: 0 ASIS:NAMES:TC:SAML:2.0:BINDING:HTTP-REDIRECT
URN: 0 ASIS:NAMES:TC:SAML:2.0:BINDING:HTTP-ARTIFACT

- **Mit diesem neuen Wissen können wir zu unserer Metadatendatei zurückkehren. Wir müssen uns darauf einigen, wie wir Nachrichten zwischen SP und IdP austauschen sollen. Jetzt haben wir also eine ziemlich vollständige Metadatendatei. Damit können wir eine SAMLFederation zwischen zwei Entitäten herstellen.**

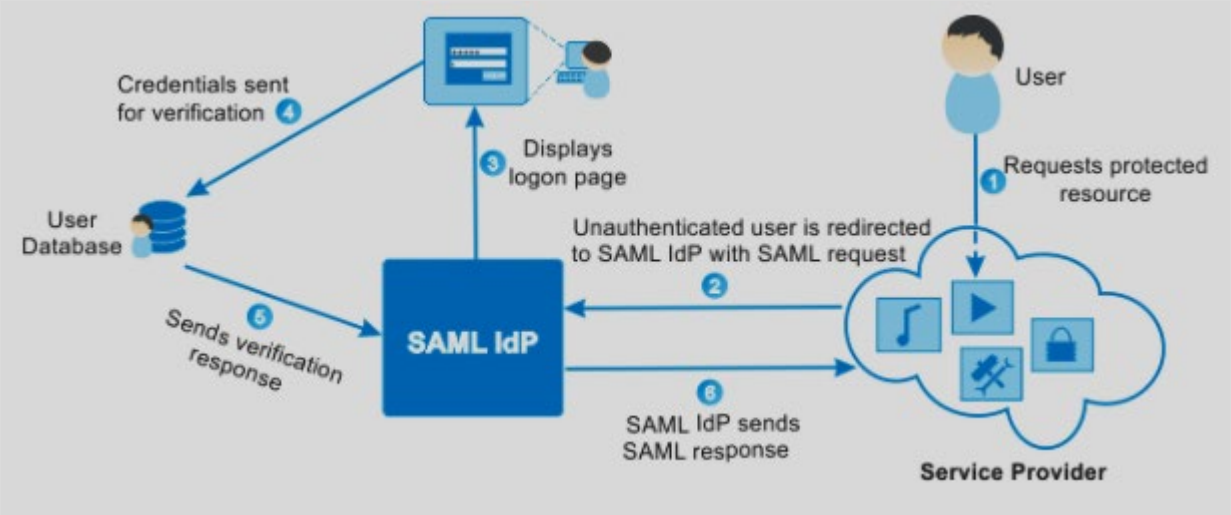


Ein Schritt tiefer in der SAML-Flow:

- Der IdP kennt die Benutzer und deren Attribute.
- Der SP hat sein eigenes Wissen über die Benutzer
- Wenn der IdP eine Assertion generiert, füllt es diese mit einer Benutzerkennung und sendet sie über den SP.
- Der SP validiert jetzt die Assertion, aber wir können sie nicht einfach im Klartext oder völlig ungeschützt senden.
- Der IdP muss also zuerst die Zusicherung unterschreiben. Auf diese Weise kann der SP den Aussteller der Assertion validieren und dadurch vertrauen.
- Als nächstes liest der SP die Benutzerkennung und versucht, sie einem Benutzer in seinem eigenen Benutzerspeicher zuzuordnen. In diesem Fall schlägt dies fehl, da das Benutzerattribut nicht gefunden wird.

- Damit Federation funktionieren kann, müssen einige Integrationsregeln festgelegt werden. Beispielsweise kann der SP vorschreiben, dass die Benutzer-ID und das Format die E-Mail-Adresse sein sollen. Dann muss der IdP zustimmen und so konfiguriert sein, dass diese übereinstimmen, wenn beide Seiten dieselbe Konfiguration haben, die die SAML-Assertion kann einem Benutzerobjekt am SP zugeordnet werden und dadurch kann der SP den Zugriff ermöglichen.

Die Konfigurationen oder Regeln für die Integration sind für die erfolgreiche Einrichtung einer SAML- Federation von entscheidender Bedeutung. Diese Konfigurationen können manuell in Ihren SP oder IdP eingegeben werden. Oft sammeln Sie die Anforderungen und Funktionen jedoch in einer XML-Metadatendatei. Diese Datei enthält die Einstellungen und das Zertifikat des Systems. Jetzt können Sie diese Dateien austauschen, um den Verbund zu konfigurieren. Durch diesen Metadatenaustausch wird die Vertrauenswürdigkeit hergestellt.


Credentials sent
for verification
4
Displays
logon page
3
User
Requests protected
resource
1
User
Database
Unauthenticated user is redirected
to SAML IdP with SAML request
2
5
Sends verification
response
SAML IdP
6
SAML IdP sends
SAML response
Service Provider

Qullenverzeichnis

- https://en.wikipedia.org/wiki/SAML_2.0
- https://www.youtube.com/watch?v=SvppXbpv-5k
- https://goteleport.com/blog/how-saml-authentication-works/
- https://bigdataanalyticsnews.com/how-does-saml-work/
- https://www.ionos.de/digitalguide/server/sicherheit/saml/

Danke
für Ihre
Aufmekrsamkeit

www.htw-berlin.de