Geschäftsstelle | Secrétariat Edulog
Erlachstrasse 21
3012 Bern

+41 31 300 55 99
info@edulog.ch

TECHNISCHES

Sicherheitsanforderungen für IdP und SP für die Integration in Edulog

20.5.2021 – Version 1.1

Föderation der Identitätsdienste im Bildungsraum Schweiz
Fédération des services d'identité de l'espace suisse de formation
Federazione dei servizi di identità nello spazio formativo svizzero
Federaziun dals servetschs d'identitad en il spazi da furmaziun svizzer

1. Ziel des Dokuments

Die Mitgliedschaft in der Föderation setzt voraus, dass Dienstleistungsanbieter (SP) und Identitätsanbieter (IdP) bei der Konfiguration ihres SAML-endpoints ein Minimum an Best Practices einhalten. Die Geschäftsstelle überprüft, ob die folgenden Elemente validiert sind und den hier veröffentlichten Mindestwerten entsprechen:

- Sicherheit des Kommunikationskanals (HTTPS)
- Die Sicherheit des SAML2-Protokolls
- Die Sicherheit des Webdienstes

Es ist zu beachten, dass für den normalen Betrieb der Föderation zwei X509v3-Zertifikate von Seiten der SP und IdP erforderlich sind. Ein Zertifikat sichert den Kommunikationskanal (HTTPS), das andere wird zur Unterzeichnung von Assertions verwendet. Die Anforderungen für die Zertifikate sind unten aufgeführt.

2. Sicherheit des Kommunikationskanals (HTTPS)

Die Sicherheit der Benutzerdaten bei der Übertragung zwischen dem SP, der Föderation und dem IdP, zu dem der Benutzer gehört, wird durch die Verwendung des TLS-Protokolls geschützt. Die Teilnehmenden der Föderation (SP, IdP) müssen die Verbindungen anhand der nachstehenden Kriterien überprüfen.

2.1 Protokolle

In Anlehnung an die OWASP-Empfehlungen EMPFIEHLT die Föderation die Verwendung der folgenden TLS-Protokolle:

Protokoll	Zulässig
TLSv1.3	Ja
TLSv1.2	Ja
TLSv1.1	Nein
TLSv1.0	Nein
SSLv2 ou SSLv3	Nein

Wenn die Verwendung der Protokolle TLSv1.0 und TLSv1.1 nicht empfohlen wird (obwohl auf einigen Cloud-Plattformen immer noch erlaubt), ist die Verwendung der Protokolle SSLv2 und SSLv3 verboten.

2.2 Verschlüsselungsalgorithmen

Für jeden Webserver empfiehlt die Föderation die Verwendung des *Intermediate grade* "TLS Cipher String" wie hier definiert https://wiki.mozilla.org/Security/Server_Side_TLS oder das Äquivalent dazu: *OWASP Cipher String 'B'*[1] . Ein Beispiel für den *Cipher String* mit fähigen Algorithmen für NGINX ist unten aufgeführt.

Diese Algorithmen können nur für TLS1.3 und TLS1.2 verwendet werden.

Wie im Dokument von Mozilla.org angegeben, *"it is the recommended configuration for the vast majority of services, considered highly secured and compatible with nearly every client released in the last five (or more) years"*.

2.3 Site Zertifikat X.509v3

Das digitale Website-Zertifikat (für SP und IdP) **MUSS ein gültiges Zertifikat sein, das von einer angesehenen kommerziellen Zertifizierungsstelle (CA) generiert wurde.**

- Vorgesehene Verwendung des Zertifikats: Digitale Signatur, Schlüsselverschlüsselung (= Digital Signature, Key Encipherment)
- Andere Verwendungen: Server Authentication, Client Authentication
- Kryptographische Merkmale (Schlüssellänge, Algorithmen, Hash-Funktionen): sollten die gleichen sein wie die, die für die Unterzeichnung des Zertifikats erforderlich sind, **siehe weiter unten.**
- Der *distinguished name* (oder *subject*) des Zertifikats muss mit dem FQN des Servers, der das Zertifikat vorlegt, identisch sein. Aus Kompatibilitätsgründen ist es am besten, sie in das `commonName` (CN)-Attribut des Zertifikats aufzunehmen UND in das Attribut `subjectAlternativeName` (SAN) (OWASP TLS Cheat Sheet).
- Vermeiden Sie wenn möglich die Verwendung von *wildcard*-Zertifikaten.
- Die Gültigkeit des Zertifikats sollte 2 Jahre nicht überschreiten (Unterschied zwischen *noAfter* und *notBefore* - *notBefore* KANN NICHT einem zukünftigen Datum entsprechen).

[1] OWASP TLS_Cipher_String_Cheat_Sheet

Die Prüfung wird insbesondere an dem digitalen Zertifikat durchgeführt, das für die RelayState-URL (an die die Attribute gesendet werden) verwendet wird, wenn es von der Website abweicht.

3. Mindestanforderungen für SAML2

3.1 Allgemeine Grundsätze

SP und IdP sollten Folgendes überprüfen:

- **SP müssen ihre SAML AuthnRequest und IdP müssen ihre SAML Response signieren** (der Endpunkt der Föderation Edulog wird dabei als IdP respektive als SP betrachtet):
    - Die Merkmale der X509-Zertifikate, die die Assertions unterzeichnen, sind in folgendem Punkt definiert
    - Die Algorithmen für die Unterzeichnung von Assertions (*request* und *response*) sind nachstehend definiert.
    - Zertifikate, die Assertions unterzeichnen, sind in den Metadaten-Dateien enthalten.
    - Die Föderation unterstützt die üblichen Standardmethoden der Kanonikalisierung, Transformation und Signaturen.
    - Im Falle von IdP bevorzugen wir die Signatur der Antwort gegenüber der Signatur der Assertion.
- Es wird erwartet, dass SP und IdP die Gültigkeit der SAML *AuthnRequest*- und SAML *Response*-Signaturen der Föderation überprüfen.
- Die Föderation validiert den SAML *AuthnRequest* von SP sowie die SAML-*Response* von IdP
- SAML Bindings: Nur zwei werden akzeptiert: HTTP-POST und HTTP-Redirect.
- SP müssen den AssertionConsumerService definieren.
- IdP müssen den SSO Dienst definieren.
- IdP müssen den SLO Dienst definieren.

3.2 Zertifikate zum Unterzeichnen von Assertions

- Das für die Signatur verwendete Zertifikat MUSS sich von dem auf der öffentlichen Website (für TLS) unterscheiden.
- Es kann ein selbstsigniertes Zertifikat sein (*self-signed*). Die minimal notwendigen kryptographischen Eigenschaften werden im folgenden Abschnitt beschrieben, insbesondere:
  - die Länge des Schlüssels (z.B.: *RSA Public-Key: (2048 bits)*)
  - seine Art (z.B.: *Public Key Algorithm: rsaEncryption*)
  - und den Signatur-Algorithmus (z.B.: *Signature Algorithm: sha256WithRSAEncryption*).
- **Die Gültigkeit des Zertifikats darf 3 Jahre nicht überschreiten** (Unterschied zwischen *notAfter* und *notBefore* - *notBefore* kann NICHT einem zukünftigen Datum entsprechen). → Obligatorisches Kriterium
- Der *distinguished name* (oder *subject*) des Zertifikats muss mit dem FQN des Servers, der das Zertifikat vorlegt, identisch sein. Aus Kompatibilitätsgründen ist es auch am besten, sie in das `commonName` (CN)-Attribut des Zertifikats aufzunehmen UND in das Attribut `subjectAlternativeName` (SAN). Zertifikate DÜRFEN NICHT vom Typ *wildcard* sein.
- **Der SP muss die Föderation unverzüglich über Verstöße gegen private Schlüssel benachrichtigen und unverzüglich Maßnahmen ergreifen, um neue Schlüssel zu generieren und zu zertifizieren.**

3.3 Geforderte kryptographische Merkmale[2]

Verschlüsselungsstärke

Da eine digitale Signatur die Verschlüsselung mittels eines privaten Schlüssels eines kryptographischen Hashs ist, müssen die Hash- und Verschlüsselungsalgorithmen angegeben werden:

- Der kryptografische Hash muss der SHA-2- oder SHA-3-Familie mit einer Schlüssellänge von mindestens 256 Bit angehören. In Fällen, in denen die Ausgabelänge unterschiedlich ist, wird dies nach dem Schrägstrich vermerkt.

Security Strength	Hash Algorithms
128	SHA-256, SHA 512/256, SHA3-256
192	SHA-384, SHA3-384
256 oder mehr	SHA-512, SHA3-512

[2] Basierend auf NIST 186-4, 800-51

- Die einzigen erlaubten asymmetrischen kryptographischen Protokolle sind: DSA, RSA et ECDSA.

Encryption Algorithm	Required Key Strength
DSA	2048 und 3072
RSA	2048 oder 3072
ECDSA	224-255, 256-383, 384-511 und alles über 512

Kommunikation in der Metadaten-Datei

Das Folgende gilt für die Unterzeichnung einer SAML-Authentifizierungsanfrage, die von einem SP für die Föderation ausgestellt wurde. Die Föderation muss diese Unterschrift vor der weiteren Bearbeitung des Authentifizierungsantrags überprüfen. Dementsprechend muss die Föderation im Besitz des öffentlichen Schlüssels sein, der dem in der digitalen Signatur verwendeten privaten Schlüssel entspricht, und muss diesen Schlüssel bei der Verifizierung verwenden. Dazu wird der öffentliche Schlüssel direkt in der Metadatendatei (*metadata file*) ausgegeben, entweder durch Aufnahme in ein in die digitale Signatur eingebettetes Zertifikat oder durch einen indirekten Verweis auf einen Zertifikatsspeicher (*store*), den die Föderation interpretieren kann. In SAML werden alle Angelegenheiten im Zusammenhang mit digitalen Signaturen an die XML-Signatur-Spezifikation delegiert.

Beispiel: XML Digital Signature

```
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" id="Example">
  ...
  <dsig:KeyInfo>
    <dsig:X509Data>
      <dsig:X509SubjectName>...</dsig:X509SubjectName>
      <dsig:X509Certificate>
      ...one possible way of transmitting the public key...
      </dsig:X509Certificate>
    </dsig:X509Data>
    <dsig:KeyValue>
      ...the digital signature itself...
    </dsig:KeyValue>
  </dsig:KeyInfo>
</dsig:Signature>
```

3.4 Metadaten-Dateien (metadata files)

In SAML2 ermöglichen die Metadaten-Dateien die Veröffentlichung der grundlegenden Elemente der Konfiguration der verschiedenen Teilnehmer eines Identitätsföderation. Das OASIS-Dokument: https://docs.oasis-open.org/security/saml/v2.0/saml-schema-metadata-2.0.xsd beschreibt die verschiedenen Syntax-Möglichkeiten, die eine Metadaten-Datei bilden kann.

Das Hauptelement, um das es hier geht, ist der «EntityDescriptor», der dank der IDPSSO-Descriptor-bzw. SPSSODescriptor-Elemente die Darstellung sowohl von IdP als auch von SP ermöglicht. Einige optionale Elemente können hinzugefügt werden, wie «Organization», «Contact».

Detail des EntityDescriptor Typs:

```
<complexType name="EntityDescriptorType">
   <sequence>
      <element ref="ds:Signature" minOccurs="0"/>
      <element ref="md:Extensions" minOccurs="0"/>
      <choice>
         <choice maxOccurs="unbounded">
            <element ref="md:IDPSSODescriptor"/>
            <element ref="md:SPSSODescriptor"/>
            ... other descriptors non-used in Edulog: RoleDescriptor, AuthnAuthori-
tyDescriptor, AttributeAuthorityDescriptor, PDPDescriptor
         </choice>
         <element ref="md:AffiliationDescriptor"/>
      </choice>
      <element ref="md:Organization" minOccurs="0"/>
      <element ref="md:ContactPerson" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="md:AdditionalMetadataLocation" minOccurs="0" maxOccurs="un-
bounded"/>
   </sequence>
   <attribute name="entityID" type="md:entityIDType" use="required"/>
   <attribute name="validUntil" type="dateTime" use="optional"/>
   <attribute name="cacheDuration" type="duration" use="optional"/>
   <attribute name="ID" type="ID" use="optional"/>
   <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

Elemente, die in der Metadatendatei vorhanden sein müssen:

- EntityDescriptor: MUSS eine EntityID sowie die erforderlichen Namensräume (metadata und XML Signature namespace) enthalten. Eine ID ist optional.
- Extensions: für *signature* und *digest*. **In Übereinstimmung mit dem vorherigen Punkt sind nur bestimmte Algorithmen zulässig**. Siehe Liste unten.
- IDPSSODescriptor/SPSSODescriptor:
    - MUSS SAML v2.0 unterstützen
    - MUSS einen KeyDescriptor mit einer KeyInfo enthalten, sowie das öffentliche Zertifikat zum Signieren von *requests*.
    - MUSS mindestens eine der Bindings HTTP-POST und/oder HTTP-Redirect verwenden.
    - Der SP MUSS den AssertionConsumerService definieren und verwenden.
    - Der IdP IdP MUSS den SingleSignOnService definieren und verwenden.
    - Der IdP IdP MUSS den SingleLogoutService definieren und verwenden.
- Ein *Organization*-Block (optional)
- Ein *Contact*-Block (optional)

Beachten Sie, dass der Block «Extensions» entweder global in EntityDescriptor (siehe SP-Beispiel) oder in den IDPSSODescriptor/SPSSODescriptor-Elementen definiert werden kann. Die Metadatendatei kann signiert werden, und es kann eine Gültigkeit festgestellt werden.

Liste der zulässigen Unterschriften und Digests für Edulog:

- `<alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha512"/>`
- `<alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha384"/>`
- `<alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>`
- `<alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha512"/>`
- `<alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha384"/>`
- `<alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>`
- `<alg:SigningMethod Algorithm="http://www.w3.org/2009/xmldsig11#dsa-sha256"/>`

Beispiel: http://www.w3.org/2009/xmldsig11#rsa-sha1 oder http://www.w3.org/2009/xmldsig11#rsa-md5 sind nicht erlaubt.

Die Föderation unterstützt Standard-Kanonikalisierungsmethoden (CanonicalizationMethod) und Transformationsmethoden (Transform).

Weitere Informationen zur XML-Sicherheit finden Sie unter https://www.w3.org/TR/xmlsec-algorithms/

Beispielmetadaten-Datei für einen SP:

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" entityID="https://unexemple.service-
provider.com/saml/..." ID="https_saml_un_sp">
   <md:Extensions xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport">
      <alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha512"/>
      <alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha384"/>
      <alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
      <alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha512"/>
      <alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha384"/>
      <alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
      <alg:SigningMethod Algorithm="http://www.w3.org/2009/xmldsig11#dsa-sha256"/>
   </md:Extensions>
   <SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:proto-
col">
      <KeyDescriptor use="signing">
         <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:KeyName>unexemple.serviceprovider.com</ds:KeyName>
               <ds:X509Data>
                  <ds:X509SubjectName>
                     emailAddress=info@serviceprovider.com,
                     CN=unexemple.serviceprovider.com,OU=IT,
                     O=serviceprovider,L=Bern,ST=Bern,C=CH
                  </ds:X509SubjectName>
                  <ds:X509Certificate>
                     MIIFuz....5MIA==
                  </ds:X509Certificate>
               </ds:X509Data>
         </ds:KeyInfo>
      </KeyDescriptor>
      <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect" Location="https://unexemple.serviceprovider.com/Shibboleth.sso/SLO/Redi-
rect"/>
      <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST" Location="https://unexemple.serviceprovider/Shibboleth.sso/SLO/POST"/>
      <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bind-
ings:HTTP-POST" Location="https://unexemple.servicepro-
vider.com/plugins/servlet/samlsso" index="0"/>
      <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bind-
ings:HTTP-Redirect" Location="https://unexemple.servicepro-
vider.com/plugins/servlet/samlsso" index="1"/>
   </SPSSODescriptor>

   <md:Organization>
      <md:OrganizationName xml:lang="en">serviceprovider.com</OrganizationName>
      <md:OrganizationDisplayName  xml:lang="en">serviceprovider.com</Organiza-
tionDisplayName>
      <md:OrganizationURL  xml:lang="en">http://www.serviceprovider.com/</Organiza-
tionURL>
   </md:Organization>

   <md:ContactPerson contactype="technical">
      <md:SurName>Muster</md:SurName>
      <md:EmailAddress>Erika.Muster@serviceprovider.com</md:EmailAddress>
   </md:ContactPerson>

</EntityDescriptor>
```

Beispielmetadaten-Datei für einen IdP:

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" entityID="https://unexemple.identityprovi-
der.com/XXXXX-XXXXX-XXXXX/saml/..." ID="https_saml_un_idp">
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
         <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
            <Reference URI="#_c4910a01-f63f-48c2-851f-5fae0e3770bf">
         <Transforms>
            <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
         </Transforms>
         <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
         <DigestValue>6MEvGDbdl...ZBAO+aA=</DigestValue>
         </Reference>
      </SignedInfo>
      <SignatureValue>ijbL...Xl0Rg==</SignatureValue>
      <KeyInfo>
         <X509Data>
            <X509Certificate>MIID...i8</X509Certificate>
         </X509Data>
      </KeyInfo>
   </Signature>

   <IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
      <md:Extensions xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport">
         <alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha512"/>
         <alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha384"/>
         <alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
         <alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha512"/>
         <alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha384"/>
         <alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
         <alg:SigningMethod Algorithm="http://www.w3.org/2009/xmldsig11#dsa-sha256"/>
      </md:Extensions>
      <KeyDescriptor use="signing">
         <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <X509Data>
               <X509Certificate>MII...5i8</X509Certificate>
            </X509Data>
         </KeyInfo>
      </KeyDescriptor>
      <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Loca-
tion="https://unexemple.identityprovider.com/XXXXX-XXXXX-XXXXX/saml2"/>
      <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Loca-
tion="https://unexemple.identityprovider.com/XXXXX-XXXXX-XXXXX/saml2"/>
      <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Loca-
tion="https://unexemple.identityprovider.com/XXXXX-XXXXX-XXXXX/saml2"/>
   </IDPSSODescriptor>
   <md:Organization>
      <md:OrganizationName xml:lang="en">identityprovider.com</OrganizationName>
      <md:OrganizationDisplayName  xml:lang="en">identityprovider.com</OrganizationDisplayName>
      <md:OrganizationURL  xml:lang="en">http://www.identityprovider.com/</OrganizationURL>
   </md:Organization>

   <md:ContactPerson contactype="technical">
      <md:SurName>Muster</md:SurName>
      <md:EmailAddress>Peter.Muster@identityprovider.com</md:EmailAddress>
   </md:ContactPerson>
</EntityDescriptor>
```

Für weitere Erklärungen oder Beispiele siehe: https://www.oasis-open.org/committees/download.php/51890/SAML%20MD%20simplified%20overview.pdf

4. Sicherheit des Webdienstes

Die Föderation EMPFIEHLT, dass in der Antwort, die von einer Website bereitgestellt wird, wenn sich ein Client verbindet, eine Reihe von Sicherheits-header vorhanden sind. Dies betrifft sowohl die Client- als auch die Server-Sicherheit.

Die hier aufgeführten Vorschläge für jeden der *header*, sowie die NGINX-Konfiguration **sind als Beispiele angegeben**. Die *header* müssen für jede Implementierung mit den notwendigen Parametern und Modifikationen angepasst werden (z. B. die erlaubten Quellen im Content-Security-Policy *header* definieren etc.)

Web header	Schutz gegen	Nginx Implementation	Kommentar
X-XSS-Protection	XSS (cross-site-scripting) reflection attacks	add_header X-XSS-Protection "1; mode=block";	Aktiviert den XSS-Filter, der in den meisten Browsern vorhanden ist.
Content-Security-Policy	XSS and Code Injection attacks	add_header Content-Security-Policy "default-src 'self';";	Definiert, welche Inhaltsquellen freigegeben sind und erlaubt dem Browser, sie zu lesen. Mehr Informationen hier: https://content-security-policy.com/
X-Frame-Options	Clickjacking protection	add_header X-Frame-Options "SAMEORIGIN" always;	Verhindert das Laden von iframes auf die Website.
X-Content-Type_options	MIME type sniffing vulnerabilities	add_header X-Content-Type-Options "nosniff";	Verhindert, dass der Browser Dateien mit einem anderen MIME-Typ als dem im HTTP Content-Type-Header angegebenen interpretiert (z.B. Behandlung von text/plain als text/css).
HSTS	MiTM type attacks	add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;" always; max age: Zeit in s, in der sich der Client nur über HTTPS wieder mit dem Server verbindet. Andere Optionen: includeSubDomains; preload	Mechanismus, der den Zugriff von Web-Browsern auf einen Server nur über HTTPS einschränkt. Mehr Informationen hier: OWASP_HSTS_Cheat_Sheet

Feature-Policy	Ermöglicht es Entwicklern, die Verwendung verschiedener Navigations- und API-Funktionen selektiv zu aktivieren und zu deaktivieren.	add_header Feature-Policy "geolocation 'none';midi 'none'; sync-xhr 'self';microphone 'none';camera 'none'; magnetometer 'none';gyroscope 'none';speaker 'self'; vibrate 'none';fullscreen 'self';payment 'none';" always;	Kann nützlich sein, um die Auswirkungen der Website auf den Datenschutz zu überprüfen. Sperrt den Zugriff Dritter auf die Browser-Fähigkeiten des Benutzers. Verletzungen der Feature-Richtlinien können über eine Reporting-API gemeldet werden. Wird ersetzt durch «Permissions-Policy» und «Document-Policy»
Permissions-Policy	Ermöglicht es Entwicklern, die Verwendung verschiedener Navigations- und API-Funktionen selektiv zu aktivieren und zu deaktivieren.	add_header Permissions-Policy "geolocation=();midi=(); sync-xhr=(self);microphone=();camera=();magnetometer=(); gyroscope=();speaker=(self); vibrate=();fullscreen=(self);payment=();";	Es ist möglich, «Feature-Policy» und diesen *header* gleichzeitig in die Konfigurationsdatei zu setzen.
Expect-CT	MiTM type Attacks	add_header Expect-CT "enforce, max-age=604800, report-uri='https://www.your-report-website.com/'"; *max-age*: teilt dem Browser mit, wie lange die Richtlinie zwischengespeichert werden soll - in s. *report-uri:* (option) Der Browser sendet einen Bericht an diese URL, wenn ein Zertifikat identifiziert wird, das die Richtlinie nicht verifiziert.	Verifiziert SSL-Zertifikate, die den Google-Richtlinien zur Transparenz von Zertifikaten entsprechen. Der Expect-CT wird voraussichtlich im Juni 2021 obsolet werden. Weist den Browser an, das Zertifikat gegen ein CA-log zu prüfen. Wenn es nicht den Anforderungen entspricht, handelt es sich um eine Fälschung und die Website ist nicht vertrauenswürdig.

4.1 Beispiel: Hinzufügen von Web-Sicherheitsfunktionen zur nginx-Konfiguration

1. Erstellen eines verbesserten Diffie-Hellman-Schlüssels - Mindestlänge 2048 Bits.

   #openssl dhparam -out dhparam.pem 2096

   Kopieren Sie es unter: /etc/nginx

2. Fügen Sie der Datei nginx.conf die folgenden Zeilen hinzu:

```
##
# Security settings v1.0 20201105
##
server_tokens off;
add_header X-XSS-Protection "1; mode=block";
add_header Content-Security-Policy "frame-ancestors 'self';";
add_header X-Frame-Options "SAMEORIGIN" always;
add_header X-Content-Type-Options "nosniff";

# Added HSTS to avoid MiTM attacks on SSL
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;" always;

# Added Expect-CT
add_header Expect-CT "enforce, max-age=604800";

# Added Feature-policy
add_header Feature-Policy "geolocation
'none';midi 'none';sync-xhr 'self';microphone 'none';camera
'none';magnetometer 'none';gyroscope 'none';speaker 'self';vibrate
'none';fullscreen 'self';payment 'none';" always;

add_header Permissions-Policy "geolocation=();midi=();sync-xhr=(self);micro-
phone=();camera=();magnetometer=();gyroscope=();speaker=(self);vibrate=();full-
screen=(self);payment=();";

# Referrer Policy
add_header Referrer-Policy same-origin;

# TLS, and Algos
ssl_dhparam           /etc/nginx/dhparam.pem;
ssl_protocols         TLSv1.2 TLSv1.3;
ssl_ciphers
TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:ECDHE-
ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-
CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:HIGH:!aN-
ULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4:!LOW:!kECDH:!DSS:!SRP:!CAMELLIA:!SEED;

# List of server side ciphers is preferred
ssl_prefer_server_ciphers on;
ssl_session_cache shared:SSL:10m;
```

3. Unnötige Informationen entfernen

Entfernen Sie zum Beispiel den *header* X-Powered-By

Versionshinweise

Datum	Version	Änderungen
20.5.2021	1.1	enthält eine Änderung bezüglich der Verpflichtung, einen SLO für IdPs einzurichten (siehe Punkte 3.1 und 3.4). Erstellung «Versionenkontrolle»