

## TECHNIQUE

# Exigences de sécurité envers les IdP et SP pour l'intégration à Edulog

20.5.2021 – version 1.1

<b>1.</b>	<b>But du document</b> .....	<b>2</b>
<b>2.</b>	<b>Sécurité du canal de communication (HTTPS)</b> .....	<b>2</b>
2.1	Protocoles .....	2
2.2	Algorithmes de chiffrement .....	3
2.3	Certificat X.509v3 du site .....	3
<b>3.</b>	<b>Exigences minimales pour SAML2</b> .....	<b>4</b>
3.1	Principes généraux .....	4
3.2	Certificats pour la signature des assertions .....	5
3.3	Caractéristiques cryptographiques demandées .....	5
3.4	Fichiers de métadonnées (metadata files) .....	7
<b>4.</b>	<b>Sécurité du service web</b> .....	<b>11</b>
4.1	Exemple : ajout de caractéristiques web security à la configuration de nginx .....	13
	<b>Notes de version</b> .....	<b>14</b>

## 1. But du document

L'admission à la Fédération nécessite que les fournisseurs de service (SP) et fournisseurs d'identité (IdP) respectent un minimum de bonnes pratiques dans la configuration de leur endpoint SAML. Le Secrétariat vérifiera que les éléments suivants sont validés et correspondent aux valeurs minimales qui sont publiées ici :

- La sécurité du canal de communication (HTTPS)
- La sécurité du protocole SAML2
- La sécurité du service web

A noter que sont EXIGÉS deux certificats X509v3 de la part des SP et IdP pour un fonctionnement normal de la Fédération. L'un sécurise le canal de communication (HTTPS) et l'autre sert à signer les assertions. Les exigences des certificats sont indiquées plus loin.

## 2. Sécurité du canal de communication (HTTPS)

La sécurité des données des utilisateurs qui transitent entre le SP, la Fédération et l'IdP auquel l'utilisateur appartient, est protégée par l'utilisation du protocole TLS. Les participants à la Fédération (SP, IdP) doivent vérifier que les connexions se conforment aux critères ci-dessous.

### 2.1 Protocoles

Conformément aux recommandations OWASP, la Fédération RECOMMANDE l'utilisation des protocoles TLS suivants :

Protocole	Zulässig
TLSv1.3	Oui
TLSv1.2	Oui
TLSv1.1	Non
TLSv1.0	Non
SSLv2 ou SSLv3	Non

Si l'utilisation des protocoles TLSv1.0 et TLSv1.1 n'est pas recommandée (mais néanmoins toujours admise sur certaines plateformes cloud), l'utilisation des protocoles SSLv2 et SSLv3 est interdite.

## 2.2 Algorithmes de chiffrement

Pour tout serveur web, la Fédération recommande l'usage du niveau Intermediate grade « TLS Cipher String » comme défini ici [https://wiki.mozilla.org/Security/Server\\_Side\\_TLS](https://wiki.mozilla.org/Security/Server_Side_TLS) ou son équivalent : *OWASP Cipher String 'B'*. Un exemple de la Cipher String avec les algorithmes admissibles pour NGINX est inclus plus loin.

Ces algorithmes sont uniquement utilisables pour TLS1.3 et TLS1.2.

Comme indiqué dans le document de Mozilla.org, "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 Certificat X.509v3 du site

**Le certificat digital du site** (pour les SP et les IdP) **DOIT être un certificat valide, généré par une autorité de certification (CA) commerciale et renommée.**

- Usage prévu du certificat : Signature numérique, Chiffrement de la clé (= Digital Signature, Key Encipherment)
- Autres usages : Server Authentication, Client Authentication
- Caractéristiques cryptographiques (longueur de la clef, algorithmes, fonctions de hash) : elles devraient être les mêmes que celles demandées pour le certificat signant les assertions (**voir plus bas**).
- Le *distinguished name* (ou *subject*) du certificat doit être le même que le FQN du serveur qui présente le certificat. Pour des raisons de compatibilité, il vaut mieux aussi l'inscrire dans l'attribut `commonName` (CN) du certificat ET dans l'attribut `subjectAlternativeName` (SAN) ([OWASP TLS Cheat Sheet](#)).
- Éviter si possible l'usage de certificats type *wildcards*.
- La validité du certificat ne devrait pas dépasser 2 ans (différence entre `notAfter` et `notBefore` - `notBefore` NE PEUT correspondre à une date future).

---

<sup>1</sup> [OWASP TLS Cipher String Cheat Sheet](#)

Le contrôle s'effectue en particulier sur le certificat digital utilisé à l'endroit où pointe l'URL RelayState (là où sont envoyés les attributs), s'il diffère du site.

### 3. Exigences minimales pour SAML2

#### 3.1 Principes généraux

S Les SP et les IdP doivent vérifier ce qui suit :

- **Les SP doivent signer leurs SAML AuthnRequest et les IdP doivent signer leurs SAML Response** (l'endpoint de la Fédération Edulog est considéré comme un IdP respectivement comme un SP pour cela).
  - Les caractéristiques des certificats X.509 signant les assertions sont définies au point suivant.
  - Les algorithmes de signature des assertions (request et response) sont définis ci-dessous.
  - Les certificats signant les assertions sont inclus dans les fichiers de métadonnées.
  - La Fédération supporte les méthodes standard usuelles de canonicalisation, transformations et de signatures.
  - Dans le cas de l'IdP, on préférera la signature de la réponse, face à celle de l'assertion.
- Il est attendu que les SP et les IdP vérifient respectivement la validité des signatures de l'assertion SAML Response et de la SAML AuthnRequest, provenant de la Fédération.
- La Fédération valide les SAML AuthnRequest provenant des SP, ainsi que les assertions SAML provenant des IdP.
- SAML Bindings seuls : HTTP-POST et HTTP-Redirect sont acceptés.
- Les SP doivent définir le service AssertionConsumerService.
- Les IdP doivent définir le service SSO.
- Les IdP doivent définir le service SLO.

### 3.2 Certificats pour la signature des assertions

- Le certificat utilisé pour la signature DOIT ÊTRE différent de celui du site web public (pour TLS).
- Il peut s’agir un certificat auto-signé (self-signed). Les caractéristiques cryptographiques minimales nécessaires sont décrites dans le point suivant, en particulier :
  - la longueur de la clef (ex : *RSA Public-Key : (2048 bits)*)
  - son type (ex : *Public Key Algorithm : rsaEncryption*)
  - et l’algorithme de signature (ex : *Signature Algorithm : sha256WithRSAEncryption*).
- **La validité du certificat ne doit pas excéder 3 ans** (différence entre *notAfter* et *notBefore* - *notBefore* NE PEUT correspondre à une date future) → **Critère obligatoire**
- Le *distinguished name* (ou *subject*) du certificat doit être le même que le FQN du serveur qui présente le certificat. Pour des raisons de compatibilité, il vaut mieux aussi l’inscrire dans l’attribut *commonName* (CN) du certificat ET dans l’attribut *subjectAlternativeName* (SAN). Les certificats NE DOIVENT PAS ÊTRE du type *wildcards*.
- **Le SP doit informer immédiatement la Fédération de toute violation de la clé privée et prendre rapidement des mesures pour générer et certifier de nouvelles clés.**

### 3.3 Caractéristiques cryptographiques demandées<sup>2</sup>

#### Force de chiffrement

Étant donné qu’une signature numérique est le chiffrement à l’aide d’une clé privée d’un hachage cryptographique, les algorithmes de hachage et de cryptage doivent être spécifiés :

- Le hachage cryptographique doit appartenir aux familles SHA-2 ou SHA-3 ayant une longueur de clé d’au moins 256 bits. Dans les cas où la longueur de sortie diffère, cela est noté après la barre oblique.

Security Strength	Hash Algorithms
128	SHA-256, SHA 512/256, SHA3-256
192	SHA-384, SHA3-384
256 ou plus	SHA-512, SHA3-512

<sup>2</sup> Basé sur NIST 186-4, 800-51

- Les seuls protocoles cryptographiques asymétriques autorisés sont : DSA, RSA et ECDSA.

Encryption Algorithm	Required Key Strength
DSA	2048 et 3072
RSA	2048 ou 3072
ECDSA	224-255, 256-383, 384-511 et tout ce qui est au-dessus de 512

### Communication dans le fichier de métadonnées

Ce qui suit s'applique à la signature d'une demande d'authentification SAML qui a été émise par un fournisseur de services à la Fédération. La Fédération doit vérifier cette signature avant de poursuivre le traitement de la demande d'authentification. En conséquence, la Fédération doit être en possession de la clé publique correspondant à la clé privée utilisé dans la signature numérique et doit utiliser cette clé lors de la vérification. Pour ce faire, la clé publique est délivrée directement dans le fichier de métadonnées (*metadata file*), en l'incluant dans un certificat intégré à la signature numérique ou via une référence indirecte à un magasin (*store*) de certificats que la Fédération est capable d'interpréter. Dans SAML, toutes les questions relatives aux signatures numériques sont déléguées à la spécification de Signature XML.

### Exemple : 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 Fichiers de métadonnées (metadata files)

Dans SAML2 les fichiers de métadonnées permettent la publication des éléments fondamentaux de la configuration des différents participants à une fédération d'identités. Le document d'OASIS : <https://docs.oasis-open.org/security/saml/v2.0/saml-schema-metadata-2.0.xsd> décrit les différentes possibilités de syntaxe que peut former un fichier de métadonnées.

L'élément principal que nous regardons ici est « **EntityDescriptor** » celui-ci permet de représenter tant les IdP que les SP, grâce aux éléments **IDPSSODescriptor** et **SPSSODescriptor** respectivement. Certains éléments optionnels peuvent être rajoutés, comme « Organization », « Contact ».

#### Détail du type EntityDescriptor :

```
<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, AuthnAuthorityDescriptor, 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="unbounded"/>
  </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>
```

### Éléments qui doivent être présents dans le fichier de métadonnées :

- EntityDescriptor : DOIT contenir un entityID, ainsi que les namespaces nécessaires (metadata et XML Signature namespace). Et optionnellement un ID.
- Extensions : pour les signatures et les *digest*. **Conformément au point précédent, seuls certains algorithmes sont admis.** Voir liste dans ci-dessous.
- IDPSSODescriptor/SPSSODescriptor :
  - DOIT supporter SAML v2.0
  - DOIT contenir un KeyDescriptor avec un KeyInfo, ainsi que le certificat public pour la signature des *requests*.
  - DOIT utiliser au moins, l'un des Bindings HTTP-POST et/ou HTTP-redirect.
  - Le SP DOIT définir et utiliser le service AssertionConsumerService.
  - L'IdP DOIT définir et utiliser le service SingleSignOnService.
  - L'IdP DOIT définir et utiliser le service SLO.
- Un bloc Organization (optionnel)
- Un bloc Contact (optionnel)

A noter que le bloc « Extensions » peut être défini, soit globalement dans EntityDescriptor (voir exemple du SP), soit dans les éléments IDPSSODescriptor/SPSSODescriptor. Le fichier de métadonnées peut être signé et une validité peut être instaurée.

### Liste de signatures et digests admissibles pour Edulog :

- `<alg:DigestMethod Algo-rithm="http://www.w3.org/2001/04/xmlenc#sha512"/>`
- `<alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha384"/>`
- `<alg:DigestMethod Algo-rithm="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"/>`

Par exemple : <http://www.w3.org/2009/xmldsig11#rsa-sha1> ou <http://www.w3.org/2009/xmldsig11#rsa-md5> ne sont pas autorisés.

La Fédération supporte les méthodes standards de canonicalisation (*Canonicalization-Method*) et de transformation (*Transform*).

Pour plus d'information sur la sécurité XML, voir ici <https://www.w3.org/TR/xmlsec-algorithms/>



## Exemple de fichier de métadonnées pour un SP :

```

<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" entityID="https://unexemple.ser-
viceprovider.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/samlssso" index="0"/>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bind-
ings:HTTP-Redirect" Location="https://unexemple.servicepro-
vider.com/plugins/servlet/samlssso" 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 contacttype="technical">
    <md:SurName>Muster</md:SurName>
    <md:EmailAddress>Erika.Muster@serviceprovider.com</md:EmailAddress>
  </md:ContactPerson>
</EntityDescriptor>

```

## Exemple de fichier de métadonnées pour un IdP :

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" entityID="https://unexemple.identitypro-
vider.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/xmldsig-more#sha256" />
        <DigestValue>6MEvGDddl...ZBA0+aA=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>ijbL...Xl0Rg==</SignatureValue>
  </Signature>
  <KeyInfo>
    <X509Data>
      <X509Certificate>MIID...i8</X509Certificate>
    </X509Data>
  </KeyInfo>
</EntityDescriptor>

<IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  <md:Extensions xmlns:alg="urn:oasis:names:tc:SAML:2.0:metadata:alg-support">
    <alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha512" />
    <alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha384" />
    <alg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#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 contacttype="technical">
  <md:SurName>Muster</md:SurName>
  <md:EmailAddress>Peter.Muster@identityprovider.com</md:EmailAddress>
</md:ContactPerson>
</EntityDescriptor>
```

Pour plus d'explications ou d'autres exemples, voir : <https://www.oasis-open.org/committees/download.php/51890/SAML%20MD%20simplified%20overview.pdf>

## 4. Sécurité du service web

La Fédération RECOMMANDE qu'un certain nombre de *headers* de sécurité soient présents dans la réponse que fournit un site web lorsqu'un client se connecte. Cela implique tant la sécurité des clients comme celle du serveur.

Les propositions données ici pour chacun des *headers*, ainsi que la configuration NGINX **sont données à titre d'exemples**. Les *headers* doivent être adaptés pour chaque implémentation avec les paramètres et modifications nécessaires (par ex : définir les sources autorisées dans le header Content-Security-Policy, etc).

Web header	Protection contre	Implémentation nginx	Commentaires
X-XSS-Protection	XSS (cross-site-scripting) reflection attacks	add_header X-XSS-Protection "1; mode=block";	Active le filtre XSS présent dans la majorité des navigateurs.
Content-Security-Policy	XSS and Code Injection attacks	add_header Content-Security-Policy "default-src 'self';"	Définit les sources de contenu qui sont approuvées et permet au navigateur de les lire. Plus d'infos ici : <a href="https://content-security-policy.com/">https://content-security-policy.com/</a>
X-Frame-Options	Clickjacking protection	add_header X-Frame-Options "SAMEORIGIN" always;	Empêche le chargement d' <i>iframes</i> sur le site web.
X-Content-Type-options	MIME type sniffing vulnerabilities	add_header X-Content-Type-Options "nosniff";	Empêche le navigateur d'interpréter des fichiers avec un type MIME différent de ce qui est spécifié dans le header HTTP Content-Type (par ex. : en traitant text/plain comme text/css).
HSTS	MiTM type attacks	add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;" always; <i>max age</i> : temps en s pendant lequel le client ne se reconnecte au serveur qu'en utilisant HTTPS. Autres options : includeSubDomains; preload	Mécanisme qui restreint les navigateurs web à n'accéder à un serveur qu'en utilisant HTTPS. Plus d'info ici : <a href="#">OWASP HSTS Cheat Sheet</a>

Feature-Policy	Permet aux développeurs d'activer et de désactiver de manière sélective l'utilisation de diverses fonctions de navigation et d'API.	<pre>add_header Feature-Policy "geolocation 'none';midi 'none'; sync-xhr 'self';microphone 'none';camera 'none'; magnetometer 'none';gyro- scope 'none';speaker 'self'; vibrate 'none';fullscreen 'self';payment 'none';" always;</pre>	<p>Peut être utile pour vérifier l'impact du site sur la confidentialité. Verrouille l'accès des tiers aux capacités du navigateur de l'utilisateur.</p> <p>Les violations de la politique de fonctionnalité peuvent être signalées via une <a href="#">API de signalement</a>. Va être remplacé par « Permissions-Policy » et « Document-Policy ».</p>
Permissions-Policy	Permet aux développeurs d'activer et de désactiver de manière sélective l'utilisation de diverses fonctions de navigation et d'API.	<pre>add_header Permissions-Pol- icy "geolocation=();midi=(); sync-xhr=(self);micro- phone=();camera=();magne- tometer=(); gyroscope=();speaker=(self); vibrate=();full- screen=(self);payment=()";</pre>	Il est possible de mettre « Feature-Policy » et ce <i>header</i> en même temps dans le fichier de configuration.
Expect-CT	MiTM type Attacks	<pre>add_header Expect-CT "en- force, max-age=604800, re- port-uri='https://www.your-re- port-website.com/'; max-age: indique au naviga- teur combien de temps il doit garder en cache la policy - en s. report-uri : (option) le naviga- teur envoi un rapport à cette URL quand un certificat ne vérifiant pas la policy est identifié.</pre>	<p>Vérifier les certificats SSL qui répondent aux règles de transparence des certificats de Google. L'Expect-CT deviendra probablement obsolète en juin 2021.</p> <p>Indique au navigateur de vérifier le certificat par rapport à un log de la CA. S'il n'est pas conforme, il est contrefait et le site n'est pas fiable.</p>

#### 4.1 Exemple : ajout de caractéristiques web security à la configuration de nginx

1. Créer des clefs Diffie-Hellman améliorées - longueur minimum de 2048 bits.

```
#openssl dhparam -out dhparam.pem 2096
```

Les copier sous : /etc/nginx

2. Ajouter les lignes suivantes dans le fichier nginx.conf :

```
##
# 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:!a-
NULL:!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. Enlever l'information non-nécessaire

Par exemple, enlever le *header* X-Powered-By

## Notes de version

Date	Version	Changements
20.5.2021	1.1	inclut une modification concernant l'obligation d'avoir un SLO pour les IdP (cf. points 3.1 et 3.4). Création « Notes de version »