eCH-0251 – Sécurité Web dans le domaine de l’IAM

Nom	Sécurité Web dans le domaine de l’IAM
eCH-nombre	eCH-0251
Catégorie	Document auxiliaire
Stade	Défini
Version	1.0.0
Statut	Approuvé
Date de décision	2024-01-08
Date de publication	2023-12-12
Remplace la version	–
Conditions préalables	-
Annexes	-
Langues	Allemand (original), français (traduction)
Auteurs	Groupe spécialisé IAM Florian Forster (ZITADEL AG) Daniel Muster (it-rm IT-Riskmanagement GmbH)
Editeur / distribution	Association eCH, Räffelstrasse 20, 8045 Zurich T 044 388 74 64, F 044 388 71 80 www.ech.ch / info@ech.ch

Condensé

Le GS IAM a élaboré par le présent un document auxiliaire traitant de la sécurité web dans le domaine IAM, qui se veut un complément nécessaire aux autres normes eCH en la matière.

Ce document auxiliaire formule des recommandations quant à la sécurité sur le web pour les protocoles IAM «SAML» et «OpenID Connect» visant notamment à réduire les vulnérabilités inhérentes aux protocoles et à ainsi réduire les risques au minimum. Ces vulnérabilités s'expliquent notamment par le fait que les normes courantes à ce sujet ne font guère mention des vulnérabilités/risques thématisés ou n'en traitent que certaines parties, avec pour conséquence des déficiences au moment de la mise en œuvre. Les recommandations du présent document auxiliaire s'appuient sur des normes internationalement reconnues d'une part et sur des publications scientifiques d'autre part.

Certaines des recommandations présentées ici peuvent aussi trouver une application dans d'autres domaines de la sécurité des réseaux. Aucune de ces recommandations toutefois n'a trait à la sécurité des systèmes.

Table des matières

1 Introduction

1.1 Statut

Approuvé: le document a été approuvé par le Comité des experts. Il a pouvoir normatif pour le domaine d'utilisation défini dans le domaine de validité donné.

1.2 But du document

Ce document est un document auxiliaire destiné aux autorités et aux particuliers dans le domaine de la sécurité Internet & Access Management ou sécurité IAM en abrégé, dans le contexte des administrations. Le document auxiliaire devrait servir notamment à un appel d'offres public ou à une mise en service d'un composant (IAM). Des mesures (non contraignantes) relatives à la sécurité web et des mesures organisationnelles y afférentes sont préconisées à cet effet.

Le modèle simple sous-jacent repose quant à lui sur les 3 composants IAM suivants.

- User Agent and User
- Relying Party (définition du terme, voir glossaire)
- Identity Provider (définition du terme, voir glossaire)

Hypothèse: La gestion des éléments susmentionnés est confiée à différentes collectivités autonomes. L'User peut être par exemple un client d'une banque (RP) qui s'y connecte via un IdP dont la gestion incombe à une autorité.

Les recommandations fournies dans ces pages s'appliquent également lorsque deux ou tous les composants sont gérés par une même collectivité autonome. Lorsque le RP et l'IdP sont administrés par la même autorité par exemple.

Les mesures de sécurité informatique visent à réduire les menaces/risques (threat en anglais), resp. les vulnérabilités (définition du terme, voir glossaire). Les menaces/risques répertoriés ici reposent sur le RFC 6819. Comme cela a déjà été évoqué, seuls les aspects de la sécurité web dans l'environnement IAM et les mesures organisationnelles, qui y sont directement liées, sont traités dans ces pages. Certaines des recommandations formulées à cet égard peuvent également s'appliquer à d'autres domaines de la sécurité web, tels ceux présentés aux chapitres 3.1, 3.2 ou 4.2.

Ces recommandations incluent:

- des contre-mesures aux attaques et aux menaces en découlant
- accompagnées de références bibliographiques.
- des activités ou omissions visant à réduire les vulnérabilités
- ainsi accompagnées de références bibliographiques.

p. ex. usurpation d'identité (attaque), menace sur l'authenticité, mesures: Signature RSA 2048 Bit SHA-512 Bit (comme contre-mesure).

Ce document s'adresse aux professionnels de la sécurité informatique. Pour autant qu'il y en ait, les explications sur les recommandations, vulnérabilités et attaques restent rudimentaires. Des références sont néanmoins fournies sur le thème en question.

Les recommandations dans le présent document et les explications à ce sujet se cantonnent aux technologies IAM suivantes:

- SAML
- OpenID Connect

Aucune de ces recommandations toutefois n'a trait à la sécurité des systèmes.

1.3 Délimitation

Il existe d'autres modèles/concepts autour de l'identification électronique (authentification) et des prestations de services associées, telle la Self Sovereign Identity (SSI) ou une approche purement «PKI». Se reporter au glossaire concernant les termes SSI et «approche PKI». Nombre des recommandations formulées ici peuvent également s'appliquer à la SSI ou à l'approche «PKI». Le chapitre 6 «Comparaison approche PKI – OIDC et SAML» propose une liste des éléments précis de cette approche PKI.

1.4 Terminologie des recommandations

Les directives dans le présent document sont indiquées selon la terminologie de [RFC2119]. Dans ce contexte, les expressions suivantes apparaissant en **LETTRES MAJUSCULES** en tant que mots, ont les significations suivantes (citation du [RFC2119]):

- **MUST:** This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification. (Il faut que ou c'est à faire.)
- **MUST NOT:** This phrase, or the phrase "SHALL NOT" mean that that definition is an absolute prohibition of the specification. (il n'est pas permis que, ce n'est pas permisil n'est pas autorisé que ou il faut ne pas le faire.)
- **SHOULD:** This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course. (Devoir, en forme conditionale «devrait».)
- **SHOULD NOT:** This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label. (Ne pas devoir, en forme conditionale «devrait».)
- **MAY:** This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)(pouvoir)

1.5 Sélection des normes

Les recommandations formulées ici sont à multiples facettes, ce qui réduit à néant toute velléité d'exhaustivité dans la présentation des normes sur le sujet. Des références à des publications scientifiques sont toutefois proposées.

2 Vue d'ensemble/structure du document

2.1 Public cible

Ce document s'adresse aux professionnels de la sécurité informatique dans le domaine de la sécurité IAM et web.

2.2 Priorité dans les recommandations

Dans les présentes recommandations, la priorité est donnée à la sécurité informatique, principalement la prévention de l'usurpation d'identité électronique, au détriment de la fonctionnalité ou de la convivialité. Pour un même prix, la sécurité d'une part et la fonctionnalité et la convivialité d'autre part se révèlent bien souvent des objectifs irréconciliables.

2.3 Contenu des différents chapitres

Le présent document cible en priorité l'usurpation d'identité ou *impersonation* en anglais. Les sous-thèmes traités ici sont organisés comme suit:

- Chapitre 3 «Recommandations générales». Comme l'indique le titre, des recommandations générales sont formulées concernant la prévention de l'usurpation d'identité électronique. Ces recommandations peuvent aussi trouver une application dans d'autres domaines de la sécurité des réseaux.
- Chapitre 4 «Enregistrer et authentifier». L'accent est mis sur les processus d'authentification (identification électronique) et les attributs et paramètres nécessaires à cet effet, ainsi que sur ce qu'il convient de vérifier lors de l'authentification.
- Chapitre 5 «Techniques d'attaque»
- Chapitre 6 «Comparaison approche PKI – OIDC et SAML». Il s'agit ici d'une comparaison de sécurité, mieux encore d'une référence à une comparaison de sécurité, entre ces 3 méthodes d'authentification d'un User. Il est également précisé quelles sont les recommandations pertinentes dans le cadre d'une approche purement PKI.

3 Recommandations générales

Ce chapitre contient des recommandations d'ordre général visant à réduire les risques découlant des vulnérabilités.

Remarque: faute de différenciation entre l'IdP et le RP, la recommandation s'applique aussi bien à l'un qu'à l'autre.

3.1 Cryptographie

3.1.1 Cryptographie asymétrique

Les procédures (algorithmes) recommandées et la façon de les paramétrer sont répertoriées dans l'ETSI TS 119 312. Le chapitre 6 de l'ENISA ECRYPT II propose une mise en perspective de la complexité estimée pour casser chaque méthode avec les longueurs de clé. A la différence d'autres normes: si aucune recommandation n'est faite quant à la durée d'utilisation d'une clé privée, la complexité pour casser le processus avec une longueur de clé appropriée y est proposée.

L'utilisatrice ou l'utilisateur peut ainsi comparer la puissance de calcul/son coût à la complexité et déterminer d'elle/lui-même la période d'utilisation de la clé en fonction de ses propres objectifs de gestion des risques.

Remarque concernant la signature avec logarithme discret ou courbes elliptiques: contrairement à RSA, la qualité du générateur de nombres aléatoires est éminemment importante au moment de la création d'une signature avec des courbes elliptiques ou des méthodes par logarithme discret (Diffie-Hellmann). Un générateur aléatoire de qualité insuffisante permet de tirer des conclusions quant à la clé privée à partir de la signature fournie.

SHOULD: cet aspect devrait être pris en compte au moment de choisir un support de stockage externe pour créer la signature (HSM, Crypto Card).

L'exigence de qualité du générateur de nombres aléatoires pourrait, dans certaines circonstances, interdire l'utilisation de cryptocartes à courbes elliptiques.

Génération: le chapitre 6.1.2 ETSI TS 119 312, V1.1.1 fournit les critères de génération des paramètres pour chaque méthode (RSA, logarithme discret, courbes elliptiques).

3.1.2 Cryptographie symétrique

SHOULD NOT: différentes clés devraient être mises à contribution afin de protéger l'authenticité (p. ex., HMAC) et la confidentialité (p. ex. chiffrement).

L'utilisation d'une même clé pour l'authenticité et le chiffrement représente une vulnérabilité du point de vue cryptographique. Par conséquent:

SHOULD NOT: le Galois Counter Mode ne devrait pas être utilisé.

3.1.3 Générateur de nombres aléatoires

MUST NOT: l'utilisation de la méthode Dual EC du NIST Special Publication 800-90, edition 2007, n'est pas autorisée.

Le chapitre 8.2.3 ETSI TS 119 312, V1.1.1 expose les critères relatifs à la génération de nombres aléatoires.

3.1.4 Protocoles de réseau (SSL/TLS)

SHOULD NOT: TLS v1.3 ne devrait pas être utilisé (tant que les risques ou vulnérabilités évoquées n'ont pas été clarifiés).

SHOULD: TLS v1.2 devrait être utilisé, Cipher: AES 256 MAC avec SHA-1 ou SHA-256

Les vulnérabilités cryptographiques de TLS v1.3 par rapport à TLS v1.2 sont les suivantes:

- Les méthodes de chiffrement se limitent à 2 algorithmes (AES et une méthode méconnue du grand public).
- Il n'existe que 5 Cipher Suites, dont 4 avec AES.
- Pour chacune de ces méthodes, la même clé est utilisée pour le chiffrement et pour la somme de contrôle cryptographique (protection de l'intégrité et authenticité).
- La longueur de la valeur de la somme de contrôle ne dépasse pas les 128 bits (SHA-1 en compte 160).
- Les paramètres de la cryptographie asymétrique sont prescrits et ne peuvent être ni modifiés ni configurés.
- Un algorithme de compression ne peut plus être mis en œuvre au niveau TLS par rapport aux versions antérieures.

Remarque: le recours à SHA-1 a beau ne plus être préconisé en cas d'utilisation de signatures valides à long terme, la méthode n'en reste pas moins utilisable dans ce contexte en raison de la courte durée de validité des valeurs SHA-1 respectives (durée de la session TLS).

3.2 Certificats X.509

SHOULD: des certificats réglementés selon la SCSE devraient être utilisés pour authentifier les serveurs.

SHOULD: des clés privées différentes devraient être utilisées pour l'authentification du serveur, pour la création des signatures de Tokens et pour le déchiffrement des Tokens.

SHOULD: dans le cas où un client est authentifié sur la base d'un certificat, ce dernier devrait être réglementé selon la SCSE.

Motif: concernant les certificats réglementés selon la SCSE, la responsabilité et le devoir de vigilance sont prescrits par la loi. La sécurité juridique s'en trouve renforcée. Concernant le cas des applications des autorités, ce type de responsabilité se révèle généralement plus avantageux pour l'autorité dans l'exercice de ses fonctions.

3.3 Session (hijacking)

Un Session ID est l'incarnation de l'authentification et de l'autorisation d'une entité pour SSL/TLS. La prise de contrôle ou le vol par un cyberpirate de la session d'une entité a des répercussions correspondantes. Le cyberpirate se voit ainsi octroyer les mêmes droits que l'entité dans le cadre de la communication.

Nombre d'attaques visent la prise de contrôle d'une session (connexion) d'autrui et ainsi l'envoi de commandes en son nom, autrement dit en faisant valoir ses droits. Les recommandations qui suivent ont pour seul but de diminuer la vulnérabilité d'une connexion (Session) aux connexions HTTP.

3.3.1 Lien entre les applications et la technologie de sécurité

Différents Session ID sont utilisés au niveau TLS (TLS Session ID) et au niveau de l'application (HTTP avec Session Cookie). Une telle séparation entre la technologie de sécurité et l'application qui doit ainsi être protégée offre des vulnérabilités qui peuvent être exploitées lors d'une attaque. L'extraction d'un cookie représente une possibilité d'attaque.

Exemple: si le TLS Session ID et le cookie n'ont aucun lien et si l'exactitude du lien ne fait l'objet d'aucune vérification, le cyberpirate peut alors se faire passer pour quelqu'un autre, rien qu'avec le cookie.

SHOULD: un lien virtuel (Channel Binding) devrait être établi entre le TLS Session ID et le HTTP Session Cookie placé par le service web.

Dans le cas où un tel lien est établi:

MUST: si la relation établie par le lien entre le HTTP Session Cookie et le TLS Session ID ne concorde pas, il faut que la connexion soit réinitialisée/interrompue.

Le serveur a recours au HTTP Session Cookie afin de distinguer les différentes connexions HTTP. Le TLS Session ID pour distinguer les différentes connexions TLS. L'absence de lien virtuel établi entre l'application (HTML, HTTP) et la technologie de sécurité (TLS) signifie que la sécurité (d'une connexion TLS) peut être contournée:

A titre d'illustration: La connexion HTTP A est établie via la TLS Session a. Le cookie HTTP â est utilisé à cet effet. S'il n'existe aucun lien virtuel correspondant, la connexion TLS B peut servir à injecter le Session Cookie â (Cookie Injection) pour ensuite prendre contrôle de la connexion HTTP A.

Et ce, quelque que soit le critère utilisé pour distinguer une HTTP Session, un JSON Web Token par exemple. Le lien entre la technologie de sécurité et l'application doit toutefois être établi puis contrôlé.

Pour un autre exemple du lien nécessaire entre l'application et la sécurité dans le cas de la signature XML, voir [12].

3.3.2 HTTP Session Cookies

Les cookies de session représentent l'identifiant de session de la connexion HTTP. Certaines attaques visent à en prendre connaissance, autrement dit à s'enquérir du Session ID. Du point de vue du cyberpirate, le jeu en vaut vraiment la chandelle dès lors qu'il n'existe aucun lien entre le SessionID de l'application et le TLS SessionID.

Le serveur dispose d'un vaste choix de paramètres dans un même cookie. Les recommandations qui suivent sont faites aux fins de protéger les Session Cookies et donc la session:

MUST: la valeur du cookie (Value) doit être générée de manière aléatoire et avoir une longueur minimale de 128 bits afin qu'elle ne soit pas prévisible.

MUST: la période de validité (Expire) d'un cookie doit être définie.

MUST: le cookie ne doit être renvoyé que pour le domaine sélectionné (adresse www).

SHOULD: les cookies ne devraient être valables que pour le domaine (Path) sélectionné.

MUST: l'échange de cookies n'est autorisé que via des connexions chiffrées (Secure).

MUST: l'échange de cookies n'est autorisé que via des connexions HTTP (HttpOnly).

MUST: les Session Cookies doivent être protégés contre toute consultation par des tiers non autorisés.

Remarque: il peut être judicieux d'enregistrer les cookies dans une optique de traçabilité des opérations.

3.3.3 Recommandations générales concernant la session

3.3.3.1 Durée d'une connexion

MUST: une connexion doit toujours être renouvelée au terme d'une période définie, et ce quelles que soient les activités effectuées avec la connexion.

MUST: la connexion doit être interrompue au bout d'un certain temps d'inactivité.

SHOULD: une seule session avec l'IdP devrait être autorisée pour chaque User Agent et chaque User.

SHOULD: le nombre de sessions par User avec l'IdP devrait être restreint.

3.3.3.2 Suppression des SessionID (cookies) qui ne sont plus utilisés

MUST: les Session ID doivent être générés de manière aléatoire.

Le choix des valeurs possibles qu'un Session ID peut avoir dépend de la probabilité qu'un identifiant soit réutilisé sur une période donnée. Cela dépend, entre autres, du nombre moyen de cookies/SessionID déposés chaque seconde.

3.3.3.3 Interruption d'une connexion (logout)

MUST: l'User doit avoir la possibilité de mettre fin à la connexion à l'IdP.

SHOULD: l'User devrait avoir la possibilité de se déconnecter du RP.

3.4 Technologies web

3.4.1 SOP/CORS

Same Origin Policy (SOP) est un caractère de sécurité dans le navigateur. Selon ce caractère de sécurité, un script sur une page web affichée pour l'User n'a pas le droit de/ne devrait pas avoir accès au contenu d'un site web d'une autre origine.

CORS est l'abréviation de l'expression anglaise Cross Origin Resource Sharing. Cela permet de partager des informations et des programmes dans divers domaines et ce, pour la construction, la conception et l'utilisation dynamiques d'un site web. Pour faire simple, CORS va à l'encontre de la Same Origin Policy, ce qui, en soi, constitue une vulnérabilité majeure.

Le partage entre domaines constitue par définition une vulnérabilité importante, à plus forte raison lorsqu'il a lieu entre les domaines de collectivités autonomes distinctes.

MUST NOT: l'IdP n'est pas autorisé à se livrer au Cross Origin Resource Sharing.

SHOULD NOT: le RP ne devrait pas se livrer au Cross Origin Resource Sharing.

Cette recommandation n'est pas sans conséquence sur la conception des sites web.

3.4.2 JavaScript

MUST: les programmes JavaScript d'un site web ne doivent être téléchargés qu'à partir du même domaine que le site web. Une règle qui vaut également pour d'autres programmes comme ActiveX.

JSON (JavaScript Object Notation) est un format de données qui est adapté au langage de programmation JavaScript. Des programmes peuvent aussi s'y trouver.

MUST NOT: le parsing du JSON ne doit pas être effectué au moyen de la fonction «eval». Avec «eval», les programmes peuvent être exécutés, sans vérification préalable, directement sur le site web chargé chez l'User.

SHOULD: JSON devrait procéder au traitement avec la fonction «JSON.parse».

SHOULD NOT: le format de fichier/d'objet JSONP permettant de charger des données dans plusieurs domaines, son utilisation devrait être évitée.

MUST NOT: les demandes «interdomaines» avec XMLhttpRequest ne doivent pas pouvoir être déposées sur la page web d'un IdP.

SHOULD NOT: les demandes «interdomaines» avec XMLhttpRequest ne devraient pas pouvoir être déposées sur la page web d'un RP.

3.4.3 iframes

MUST NOT: l'échange de données/programmes entre 2 Frames de différente origine ne doit pas être effectué auprès de l'IdP.

SHOULD NOT: l'échange de données/programmes entre 2 Frames de différente origine ne devrait pas être effectué auprès du RP,

sous peine de provoquer des vulnérabilités susceptibles d'être exploitées par une attaque, p. ex. cf. chapitre 5.4 «UI Redressing (Click Jacking)» et chapitre 5.5 «Callback Function».

3.5 HTTP Header

Tous les caractères de sécurité suivants, si tant est qu'ils le soient, ne sont pris (correctement) en charge par les navigateurs qu'à partir d'une certaine version. Il est donc conseillé de commencer par vérifier quel navigateur, à partir de quelle version, gère correctement les caractères de sécurité requis. Avec pour conséquence que certaines versions de navigateur se verront (ou devraient se voir) refuser l'accès au site web ou bloquées. La Security Policy doit servir à déterminer si les caractères de sécurité sont nécessaires et à en déduire les exigences imposées au navigateur.

Source: https://caniuse.com/

3.5.1 Transmission HTTPS

MUST: pour éviter une connexion non sécurisée de l'User Agent à l'IdP et au RP, l'IdP et le RP doivent procéder à une redirection http vers https. Seule la redirection des appels au sein du même domaine est alors autorisée.

Exemple: http://ech.ch -> https://ech.ch mais pas http://www.ech.ch -> https://ech.ch

Motif: réduction du risque face aux attaques suivantes: MitM, http Bookmark, Web Application configuration error

MUST: si une redirection du domaine est en outre nécessaire, elle doit être effectuée selon le protocole d'origine.

Exemple: http://www.ech.ch -> http://ech.ch -> https://ech.ch

Contre-exemple: http://www.ech.ch -> https://ech.ch

Sources: OWASP Cheat Sheet [3]

3.5.2 Strict Transport Security

MUST: afin d'utiliser, autant que faire se peut, une connexion protégée par TLS entre l'User Agent et l'IdP après l'établissement de la première connexion, l'IdP doit définir un HSTS Header ayant une longue durée de vie.

SHOULD: afin d'utiliser autant que faire se peut une connexion protégée par TLS entre l'User Agent et le RP après l'établissement de la première connexion, le RP devrait définir un HSTS Header ayant une longue durée de vie.

Remarque: les avantages et inconvénients de cette recommandation sont explicités dans l'OWASP Cheat Sheet [3]. L'adresse de domaine du serveur peut être déterminée de manière granulaire afin de permettre la mise en œuvre de la recommandation et d'éviter les effets secondaires indésirables.

Exemple: utiliser le HTTS Header en conséquence sur pattern.example.com et non de manière exhaustive sur example.com

Configuration: Strict-Transport-Security: max-age=63072000; includeSubDomains; preload

Sources: RFC 6797 OWASP Cheat Sheet [3]

Réduction du risque face aux attaques suivantes: MitM, http Bookmark, Web Application configuration error

3.5.3 Strict Transport Security Preload List

MUST: Pour qu'une connexion TLS soit déjà à l'œuvre avant même la première visite d'un site web, un domaine doit être chargé dans la HSTS Preload List Submission chez l'IdP. L'IdP se doit de l'enregistrer.

Dans certaines circonstances, cette recommandation peut ne pas être applicable à l'Intranet. Toutefois, le RP et l'IdP sont alors administrés par la même collectivité autonome. Cependant, les recommandations formulées ici visent principalement à ce que le RP et l'IdP soient administrés par des collectivités distinctes, voir l'encadré au chapitre 1.2.

SHOULD: il en va de même du RP.

Motif: Réduction du risque face aux attaques suivantes: MitM, http Bookmark, Web Application configuration error

Sources: RFC 6797 OWASP Cheat Sheet [3], HSTS Preload List Submission

3.5.4 X-Frame-Options

Les options X-Frame donnent la capacité de contrôler le chargement des iframes dans un Frame.

3 arguments différents plaident en faveur des options X-Frame. DENY, SAMEORIGIN, ALLOW-FROM ORIGIN

MUST: les IdP doivent utiliser l'argument «DENY», qui interdit de charger le du site web de l'IdP sur tout autre site web.

SHOULD: les RP devraient utiliser l'argument «DENY» ou «SAMEORIGIN». Ce dernier permet de ne charger que les pages web de même origine.

La RP s'expose à un risque considérable lorsqu'il se refuse à suivre la recommandation.

L'User fait attention à bien saisir le mot de passe pour l'authentification auprès de l'IdP dans un Frame séparé du RP.

Remarque: la directive CSP (Content Security Policy) «frame-ancestors» écrase le X-Frame-Options Header. Si une ressource établit les deux directives, la directive CSP frame-ancestors est appliquée et la directive X-Frame-Options est ignorée.

3.5.5 Content Security Policy

MUST: afin de protéger l'application web contre toute modification non souhaitée, l'IdP est tenu de fournir une CSP (Content Security Policy).

SHOULD: le RP devrait également mettre cela à disposition.

Motif: réduction du risque face aux attaques suivantes: XSS, Framing, Clickjacking (UI Redressing)

Sources: Content Security Policy Cheat Sheet, OWASP, Cross Site Scripting Prevention, Content Security Policy (CSP) – HTTP, https://report-uri.com/home/generate

Remarque: seules quelques versions de navigateur plus récentes prennent en charge les caractères de sécurité, voir https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP.

MUST: Il faut commencer par vérifier quels navigateurs, et à partir de quelle version, prenaient en charge les caractères de sécurité suivants.

3.5.5.1 Recommandations des CSP pour l'IdP

Le tableau suivant, qui présente les caractères des CSP, est fourni par l'OWASP.

base-uri	Define the base uri for relative uri. **MUST:** l'IdP doit le définir et l'appliquer.
default-src	Define loading policy for all resources type in case of a resource type dedicated directive is not defined (fallback). **MUST:** l'IdP doit le définir et l'appliquer. La Fallback Resource doit être l'IdP.
script-src	Define which scripts the protected resource can execute. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
object-src	Define from where the protected resource can load plugins. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
style-src	Define which styles (CSS) the user applies to the protected resource. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
img-src	Define from where the protected resource can load images. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.

media-src	Define from where the protected resource can load video and audio. **SHOULD NOT:** le site web de l'IdP ne devrait en principe contenir aucune référence à des sources audio et vidéo. Les exceptions sont par exemple l'accessibilité, tel l'accès pour les personnes malvoyantes. Que les références aux sources vidéo et audio soient incluses ou non. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
frame-src	Deprecated and replaced by child-src. Define from where the protected resource can embed frames.
child-src	Define from where the protected resource can embed frames. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
frame-ancestors	Define from where the protected resource can be embedded in frames. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
font-src	Define from where the protected resource can load fonts. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
connect-src	Define which URIs the protected resource can load using script interfaces. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
manifest-src	Define from where the protected resource can load manifest. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
form-action	Define which URIs can be used as the action of HTML form elements. **MUST:** ce caractère doit être utilisé et la ressource ne doit être que l'IdP.
sandbox	Specifies an HTML sandbox policy that the user agent applies to the protected resource. **MUST:** même si les iframes ne sont pas autorisés, pour tous les cas, ce caractère doit être défini de sorte à être aussi restrictif que possible.
script-nonce	Define script execution by requiring the presence of the specified nonce on script elements. **MUST:** ce caractère doit être utilisé.
plugin-types	Define the set of plugins that can be invoked by the protected resource by limiting the types of resources that can be embedded. **MUST:** le nombre de plugins doit être restreint.

reflected-xss	Instructs a user agent to activate or deactivate any heuristics used to filter or block reflected cross-site scripting attacks, equivalent to the effects of the non-standard X-XSS-Protection header. **MUST:** ce caractère «Activer» doit être activé.
block-all-mixed-content	Prevent user agent from loading mixed content. **MUST:** ce caractère doit être utilisé. Le site web ne doit pas contenir un mix de contenus (protégé par TLS, non protégé).
upgrade-in-secure-requests	Instructs user agent to download insecure resources using HTTPS. **MUST NOT:** ce caractère ne doit pas être utilisé.
referrer	Define information user agent must send in Referer header. **MUST NOT:** ce caractère ne doit pas être utilisé, car il n'est pas pris en charge par le navigateur, voir https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP L'alternative est la Referrer Policy, voir chapitre 3.5.7.
report-uri	Specifies a URI to which the user agent sends reports about policy violation. Voir chapitre 3.5.5.3 «Content Security Policy Reporting»
report-to	Specifies a group (defined in Report-To header) to which the user agent sends reports about policy violation. Voir chapitre 3.5.5.3 «Content Security Policy Reporting»

Tableau 1: Recommandations pour l'IdP concernant les paramètres pour la Content Security Policy

3.5.5.2 Recommandations de la Content Security Policy pour le RP

base-uri	Define the base uri for relative uri. **MUST:** Ceci doit être défini et appliqué.
default-src	Define loading policy for all resources type in case of a resource type dedicated directive is not defined (fallback). **MUST:** le RP doit le définir et l'appliquer. La Fallback Resource doit être le RP.
script-src	Define which scripts the protected resource can execute. **SHOULD:** ce caractère devrait être utilisé et la ressource devrait être uniquement le RP.
object-src	Define from where the protected resource can load plugins. **SHOULD:** ce caractère devrait être utilisé et la ressource devrait être uniquement le RP.

style-src	Define which styles (CSS) the user applies to the protected resource. **SHOULD:** ce caractère devrait être utilisé et la ressource devrait être uniquement celle du RP.
img-src	Define from where the protected resource can load images. **SHOULD:** ce caractère devrait être utilisé et la ressource devrait être uniquement celle du RP.
media-src	Define from where the protected resource can load video and audio. **SHOULD:** ce caractère devrait être utilisé et la ressource devrait être uniquement celle du RP.
frame-src	Deprecated and replaced by child-src. Define from where the protected resource can embed frames.
child-src	Define from where the protected resource can embed frames. **MUST:** ce caractère doit être utilisé et la ressource ne peut être que le RP.
frame-ancestors	Define from where the protected resource can be embedded in frames. **MUST:** ce caractère doit être utilisé et la ressource ne peut être que le RP.
font-src	Define from where the protected resource can load fonts. **SHOULD:** ce caractère devrait être utilisé et la ressource devrait être uniquement le RP.
connect-src	Define which URIs the protected resource can load using script interfaces. **SHOULD:** ce caractère devrait être utilisé et la ressource devrait être uniquement le RP.
manifest-src	Define from where the protected resource can load manifest. **SHOULD:** ce caractère devrait être utilisé et la ressource devrait être uniquement le RP.
form-action	Define which URIs can be used as the action of HTML form elements. **SHOULD:** ce caractère devrait être utilisé et la ressource devrait être uniquement celle du RP.

sandbox	Specifies an HTML sandbox policy that the user agent applies to the protected resource. **MUST:** ce caractère doit être utilisé pour les iframes provenant d'une source externe. **SHOULD:** le caractère devrait être défini de sorte à être aussi restrictif que possible. **SHOULD:** ce caractère devrait être utilisé pour les iframes provenant d'une source externe. **MUST:** fans le cas où l'iframe provient d'une source interne, le caractère doit être utilisé de sorte à être cohérent avec les autres caractères la Content Security Policy.
script-nonce	Define script execution by requiring the presence of the specified nonce on script elements. **SHOULD:** ce caractère devrait être exploité.
plugin-types	Define the set of plugins that can be invoked by the protected resource by limiting the types of resources that can be embedded. **MUST:** le nombre de plugins doit être restreint.
reflected-xss	Instructs a user agent to activate or deactivate any heuristics used to filter or block reflected cross-site scripting attacks, equivalent to the effects of the non-standard X-XSS-Protection header. **MUST:** ce caractère «Activer» doit être activé.
block-all-mixed-content	Prevent user agent from loading mixed content. **MUST:** ce caractère est à utiliser. **MUST NOT:** le site web ne doit pas contenir un mix de contenus (protégé par TLS, non protégé).
upgrade-in-secure-requests	Instructs user agent to download insecure resources using HTTPS. **MUST NOT:** Ce caractère ne doit pas être utilisé.
referrer	Define information User Agent must send in Referrer header. **MUST NOT:** ce caractère ne doit pas être utilisé, car il n'est pas pris en charge par les navigateurs, voir https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP L'alternative est la Referrer Policy, voir chapitre 3.5.7.
report-uri	Specifies a URI to which the user agent sends reports about policy violation. Voir chapitre 3.5.5.3 «Content Security Policy Reporting»

report-to	Specifies a group (defined in Report-To header) to which the user agent sends reports about policy violation. Voir chapitre 3.5.5.3 «Content Security Policy Reporting»

Tableau 2: Recommandations pour l'IdP concernant les paramètres pour la Content Security Policy

3.5.5.3 Content Security Policy Reporting

MUST: le reporting doit être cantonné aux événements survenant au moment des connexions entre le navigateur et l'IdP et entre le navigateur et le RP.

SHOULD: à défaut de base juridique imposant un reporting, le principe de proportionnalité devrait s'appliquer:

> Dans le cas où le besoin de protection du RP ou de l'IdP surpasse le besoin de protection de la vie privée de l'User, le besoin de protection de la vie privée de l'User doit être relégué au second plan pour la sécurité de l'IdP ou du RP. Un reporting devrait donc être effectué. Encore faut-il pour cela que l'User y ait donné son accord au préalable.

MAY: dans le cas où l'User n'est pas d'accord, la connexion peut être interrompue.

Sources pour la mise en œuvre du reporting: Content Security Policy Cheat Sheet, OWASP

3.5.6 Enregistrement de Tokens tels que JWT, SAML

MUST: dans le cas où les Tokens reçus doivent être conservés chez l'IdP/le RP à des fins de traçabilité des processus, alors le RP/l'IdP doit durablement protéger l'intégrité et la confidentialité des Tokens reçus (contre l'accès non souhaité par des tiers).

Remarque: le stockage des Tokens, des annonces SAML par exemple, peut être judicieux/nécessaire, notamment pour la traçabilité des opérations. La traçabilité permet, en cas d'erreur, de déterminer de manière fiable la cause de l'erreur et ainsi, le cas échéant, la responsabilité. Rappelons que «L'IdP et le RP peuvent être gérés par des collectivités différentes».

3.5.7 Referrer Policy

MUST: Afin de garantir une protection adéquate des informations entre l'IdP et le RP, tous deux doivent s'entendre sur une Referrer Policy qui proscrive toute «fuite» de données sensibles vers une autre partie.

Les mesures suivantes doivent être respectées afin de réduire le risque de fuite de données (Data Leakage): strict-origin-when-cross-origin, no-referrer, strict-origin.

Sources: OWASP Secure Headers Project, Referrer Policy, W3C Candidate Recommendation, 26 January 2017

3.5.8 X-Content-Type-Options

Le http-Header x-content-type-options peut empêcher l'User Agent d'interpréter les types MIME de façon autonome. Cela permet d'éviter le chargement et l'exécution non souhaités de code en tant qu'autre type de données.

La mesure réduit le risque par rapport aux attaques suivantes: MIME sniffing, Remote Code Execution, OWASP Secure Headers Project

MUST: l'IdP et le RP doivent utiliser http header x-content-type-options avec l'option «nosniff».

Sources: X-Content-Type-Options – HTTP, MIME types (IANA media types)

3.5.9 X-XSS-Protection

Le X-XSS-Protection-Header n'a plus cours dans les navigateurs modernes et son utilisation peut être à l'origine de problèmes de sécurité côté client. Il est donc conseillé de définir le Header sur Protection X-XSS sur «0», le but étant de désactiver le XSS-Auditor et de ne pas le laisser influer sur le comportement par défaut du navigateur.

MUST: ce Header doit être réglé sur X-XSS-Protection = «0».

3.5.10 Cache-Control

Les navigateurs enregistrent souvent des données susceptibles de laisser derrière elles des informations sensibles pour les autres. Le HTTP Header «Cache-Control» est censé servir à gérer le stockage des informations dans le cache du navigateur.

SHOULD: le Cache Control Header pour les sites privés/confidentiels devrait être défini sur no-store. Les ressources statiques font exception à cette règle.

Dans le cas où il n'est pas possible de se conformer à la recommandation, no-cache, max-age=1000 doit être défini.

Sources: RFC 7234, RFC 5861, RFC 8246, https://developer.mozilla.org/de/docs/Web/HTTP/Headers/Cache-Control#spezifikationen

3.5.11 Feature Policy

Le caractère «Feature Policy», précédemment connu sous le nom de «Permissions Policy», sert à déterminer ce qui peut être utilisé dans le Frame du navigateur.

> «The HTTP Feature-Policy header provides a mechanism to allow and deny the use of browser features in its own frame, and in content within any <iframe> elements in the document. »

Le risque de fuites de données (vers des tiers de manière involontaire) s'en trouve ainsi réduit.

SHOULD: il faudrait tout d'abord convenir de vérifier quels navigateurs prennent en charge ce caractère, base sur laquelle il devrait être décidé ou non de l'utiliser. Concernant la prise en charge de cette fonctionnalité dans les navigateurs, voir https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Feature-Policy.

Des explications sur les fonctionnalités suivantes sont disponibles à l'adresse ci-dessus.

> accelerometer=(),autoplay=(),camera=(),display-capture=(),document-domain=(),encrypted-media=(),fullscreen=(),geolocation=(),gyroscope=(),magnetometer=(),microphone=(),midi=(),payment=(),picture-in-picture=(),publickey-credentials-get=(),screen-wake-lock=(),sync-xhr=(self),usb=(),web-share=(),xr-spatial-tracking=()

Dans le cas où le navigateur prend en charge cette fonctionnalité:

> **MUST:** concernant l'IdP, il faut: Toutes les fonctionnalités doivent être définies et non autorisées, hormis microphone (), camera=(),display-capture=(), publickey-credentials-get=() en raison de l'accessibilité et de l'utilisation de codes QR et WebAuthN.
>
> **MUST:** le RP doit proscrire les fonctions inutiles au moyen de la sélection de paramètres appropriés.

MAY: <allowlist> peut actuellement être explicitement vide.

Sources: OWASP Secure Headers Project – OWASP, https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Feature-Policy

3.5.12 Autres

Les Headers suivants pourront devenir pertinents à l'avenir, car recommandés par l'OWASP. Cross Origin Resource Sharing (CORS)

- Cross Origin Opener Policy (COOP)
- Cross Origin Embedder Policy (COEP)
- Cross Origin Resource Policy (CORP)
- Cross Origin Read Blocking (CORB)

C'est la raison pour laquelle il convient de garder un œil sur les activités entourant les recommandations des Headers. Reste à déterminer ensuite si elles devraient être prises en compte.

MAY: Ces caractères de sécurité peuvent être intégrés en gardant à l'esprit que tous les navigateurs (versions) ne sont pas forcément en mesure de les prendre en charge.

MUST: Si les caractères de sécurité sont utilisés, les points suivants doivent s'appliquer:

> **MUST NOT:** l'IdP ne devrait pas se livrer au CrossResource Sharing.
>
> **MUST:** en conséquence de quoi, les caractères correspondants doivent être définis de façon à remplir cette exigence.
>
> **SHOULD:** le RP ne devrait pas se livrer au Cross Resource Sharing. Si tel est le cas, alors
>
> **MUST:** les paramètres du RP doivent alors être définis de manière à ce que seule l'utilisation de cette ressource soit autorisée.

Sources: OWASP Secure Headers Project – OWASP

3.6 Domain Name System Isolation

MUST: l'IdP doit réserver un nom de domaine FQDN uniquement pour l'interaction (Login, Reset, Register) avec l'utilisatrice/utilisateur.

MUST NOT: aucun autre service ne peut utiliser le même nom de domaine.

Ceci réduit le risque par rapport aux attaques suivantes: Cookie extraction, XSS, CSRF

Sources: NIST Special Publication 800-81-2

3.7 Domain Name System Security Extensions

On utilise DNSSec afin de préserver l'authenticité des réponses aux requêtes DNS.

MUST: les réponses DNS aux domaines de l'IdP doivent être protégées en utilisant DNSSec.

SHOULD: les réponses DNS aux domaines du RP devraient être protégées en utilisant DNSSec.

DANE utilise un point de départ différent pour la fiabilité. Le point de départ recommandé ici est celui des certificats d'un Certificate Service Provider reconnu selon la SCSE.

Sources: NIST Special Publication 800-81-2

4 Enregistrer et authentifier

Ce chapitre formule des recommandations de vigilance en lien avec l'enregistrement et l'authentification en vue de réduire au minimum les vulnérabilités en matière d'authentification.

4.1 De quoi est-il question?

Trop souvent, les technologies de sécurité et les applications qu'elles protègent ne sont pas adaptées les unes aux autres et tendent à utiliser différents identifiants pour accéder aux ressources. Il en résulte un risque d'attaques.

Exemple: avec OpenID Connect, les composants sous l'administration/la responsabilité de l'IdP et du RP ont des identifiants différents pour leurs objets.

Exemple: les connexions TLS et HTTP ne sont, en soi, pas coordonnées. Avec la connexion HTTP, l'accès aux ressources est accordé au moyen du cookie. Il n'existe en règle générale aucun lien virtuel entre le cookie et le TLS Session ID.

Si un tel lien existait, il ferait l'objet d'une vérification à chaque fois que le cookie concorde avec le TLS Session ID ayant servi à l'authentification de l'User. La «mise en relation» du TLS Session ID avec le cookie offre une protection contre les abus en cas de vol/copie d'un cookie par un tiers non autorisé.

Remarque: le lien établi entre TLS et l'application (Web) est également connu sous le nom de Channel Binding.

A l'inverse: Si aucun lien correspondant n'est établi entre le cookie et le TLS Session ID et que l'on ne vérifie pas à chaque fois que cela concorde, il en résulte une vulnérabilité.

Exemple: l'attaque XML Signature Wrapping exploite l'inadéquation entre l'application et la technologie de sécurité (signature XML), voir à ce sujet [12]: Si une signature est bien créée, les informations sensibles envoyées à l'application ne sont toutefois plus protégées par la signature après l'attaque, elles peuvent donc être remplacées. Il est possible de réduire le risque d'une telle attaque en procédant comme suit:

> Convenir d'un schéma XML. Celui-ci stipule l'endroit où trouver les informations relatives à l'utilisatrice/l'utilisateur et ce que la signature doit protéger. Outre la validité de la signature, il faut aussi vérifier si l'objet XML obtenu est conforme au schéma et si la signature protège comme prévu les sous-objets requis.

4.2 Principes fondamentaux concernant l'enregistrement/l'authentification

L'enregistrement consiste à recueillir et à vérifier les attributs se rapportant à l'identité d'une personne physique ou morale. Ces attributs devraient permettre, lors de l'authentification, de rattacher (autant que possible) sans équivoque l'information reçue à la personne concernée.

La fiabilité de cette attribution et la désignation de la responsabilité sont fonction de la fiabilité et du type d'attributs, ainsi que de la qualité de l'authentification, se reporter à ce sujet à eCH-0170 et à eCH-0171. Il faut à ce sujet définir les exigences minimales en matière de qualité de l'authentification.

MUST: tous les identifiants nécessaires à l'authentification et à l'autorisation doivent être inclus. Dans le cas des serveurs, ceux-ci doivent être mis en relation avec les renseignements du certificat X.509, autrement dit être reliés entre eux, sous forme de tableau ou par une signature par exemple, dans le but de protéger l'authenticité et l'intégrité de la relation.

Les valeurs dynamiques individuelles (paramètres) pour chaque processus d'authentification doivent également être mises en rapport avec les identifier enregistrés et vérifiés pour la durée de la session.

Exemple: du point de vue du RP pour l'User Access à OpenID Connect, la relation est constituée comme suit:

> Confirmation d'authentification de l'IdP, TLS Session avec l'User et cookie de la connexion http ainsi protégée. Si la valeur du cookie ne correspond plus au TLS Session ID, il faut interrompre la connexion ou (re)valider l'authentification.

MUST: il faut vérifier que cette relation (voir exemple ci-dessus) ait bien été respectée lors de l'authentification et pour la durée de la session. S'il est constaté que la relation ne correspond pas à ce qui a été défini/convenu, l'établissement de la connexion ou la session doit être interrompu et une erreur correspondante être signalée/enregistrée.

Exemple: concernant l'authentification de l'IdP du point de vue du RP dans OpenID Connect, voir le chapitre 5.4.3. du présent document.

Comme cela a été évoqué, le non-respect des règles mentionnées provoque des vulnérabilités/risques (supplémentaires) au niveau de l'authentification et de l'autorisation.

Les sous-chapitres suivants énoncent des recommandations relatives à l'enregistrement, à l'authentification et aux accords à passer entre les parties concernées.

4.3 Entre l'IdP et l'User/User Agent

4.3.1 Enregistrement

4.3.1.1 L'User auprès de l'IdP

SHOULD: l'IdP devrait identifier les utilisatrices et les utilisateurs, en saisir les attributs et les vérifier. Les attributs nécessaires à l'authentification doivent être assortis d'un moyen d'authentification.

Du (niveau de) sécurité exigé(e) dépend la possibilité d'identifier une personne et avec quelle fiabilité. Le règlement d'exécution (UE) 2015/1502 de la Commission du 8 septembre 2015 relatif à l'eIDAS spécifie les exigences en matière d'authentification et d'identification par niveau de sécurité.

Remarque: si l'authentification nécessite un niveau de sécurité élevé, un enregistrement (saisie des attributs) en personne s'impose. Justification: si les moyens d'authentification peuvent être transférés, il n'en va pas de même des caractéristiques biométriques d'une personne physique.

SHOULD: l'IdP devrait collecter suffisamment d'attributs auprès de l'User, les vérifier, puis les joindre à la confirmation d'authentification ou à l'ID Token (User Info préférable avec OpenID Connect), afin que le RP puisse procéder à l'autorisation sur la base des attributs sur l'User fournis par l'IdP. En d'autres termes, l'opération de connexion et l'autorisation peuvent être clairement attribuées à un User. Si ce n'est pas le cas, un enregistrement séparé de l'User doit être effectué auprès du RP ou les informations doivent être obtenues auprès d'une autre base de données, responsable de l'exactitude des attributs correspondants.

Par exemple si le RP demande une certification professionnelle et que celle-ci n'est pas enregistrée auprès de l'IdP ou que cet enregistrement n'est pas autorisé. Les permis d'exercer la profession de médecin, de notaire ou d'avocat sont autant d'exemples de certifications professionnelles. Ces informations sont administrées dans les registres correspondants, autrement dit mises à jour, entre autres. Si le RP veut maintenant savoir si la certification professionnelle est toujours valide, il doit adresser au registre une demande à cet effet, car les informations ne sont pas disponibles auprès de l'IdP, ne doivent pas l'être ou ne sont pas mises à jour assez rapidement.

4.3.1.2 L'IdP auprès de l'User

MUST: le serveur doit être authentifié sur la base d'un certificat X.509 avec TLS.

MUST: l'User doit avoir implémenté l'URL de l'IdP et le certificat racine auprès de l'User Agent, qui est utilisé pour la vérification du certificat TLS de l'IdP.

4.3.2 A vérifier lors de l'authentification

4.3.2.1 Par l'User

MUST: en cas d'annonce d'erreur portant sur l'établissement d'une connexion TLS par l'User Agent, l'User doit fermer la fenêtre du navigateur et ainsi interrompre la connexion servant à la communication de données.

MUST: une fois la connexion établie avec l'IdP, la personne physique doit vérifier que l'URL au niveau de l'User Agent Frame correspond bien à l'URL de l'IdP qu'il connaît et que la connexion est protégée par TLS.

4.3.2.2 Par l'IdP

MUST: l'IdP doit authentifier l'User. Il faut s'assurer que le moyen utilisé à cette occasion pour l'authentification concorde bien avec celui convenu au moment de l'enregistrement.

SHOULD: l'IdP devrait vérifier au moyen de méthodes heuristiques si la connexion entre l'User Agent et l'IdP est intacte et valide.

4.3.3 Règles de responsabilité (droits et obligations)

MUST: la personne physique doit notamment être informée de la façon dont elle doit traiter le moyen d'authentification, autrement dit des obligations de vigilance qu'elle doit remplir à cet égard.

SHOULD: les obligations de vigilance à respecter devraient être formulées avec la plus grande précision possible et être fonction du niveau de sécurité visé. Les formulations telles que «approprié» ou «selon les circonstances» sont à proscrire ou à spécifier, faute de quoi l'User risque d'avoir du mal à s'en faire une idée précise.

MUST: les devoirs de l'IdP en matière de traitement des données enregistrées et dans la vérification de l'authentification de l'User doivent être définis et rendus accessibles à l'User. Cela signifie qu'il doit être mis au courant de l'endroit où il peut obtenir ces informations.

4.4 Entre le RP et l'User Agent/User

4.4.1 Enregistrement

L'User et le RP ne doivent en principe pas s'enregistrer mutuellement. Les cas où le RP ne reçoit pas suffisamment d'informations de l'IdP pour pouvoir mener à bien la procédure d'autorisation de l'User font exception à cette règle.

4.4.2 A vérifier lors de l'authentification

4.4.2.1 Par l'User

MUST: le serveur sera authentifié sur la base d'un certificat X.509 avec TLS.

MUST: une fois la connexion établie avec le RP, l'User doit vérifier que l'URL au niveau de l'User Agent Frame correspond bien à l'URL du RP qu'il connaît et que la connexion est protégée par TLS.

MUST: l'User doit donner son accord au préalable à ce que le RP ait accès à la ressource de l'User ou reçoive ses données personnelles. Au moment de donner son consentement, l'User doit pouvoir identifier de manière claire et sans équivoque la personne à laquelle il accorde son consentement. C'est à l'IdP qu'il incombe d'y veiller.

4.4.2.2 L'User auprès du RP

L'authentification est réalisée via la confirmation d'authentification de l'IdP. Concernant OpenID Connect, cela se fait avec ID Token ou un Refresh Token Avec SAML, via une assertion.

SHOULD: la confirmation de l'authentification réussie de l'User devrait être signée par l'IdP. Dans le cas d'OpenID Connect, p. ex., avec une signature du JSON Token.

Concernant la traçabilité des processus du point de vue du RP en particulier, l'envoi au RP d'attributs de l'User qui n'ont pas été signés par l'IdP ou l'instance User Info pose problème avec OpenID Connect.

MUST: dans le cas où le RP et l'IdP ne sont pas gérés par la même collectivité autonome, les attributs de personne nécessaires à l'authentification et à l'autorisation doivent être signés et envoyés par l'IdP au RP. Le RP doit contrôler la signature et le contenu de la confirmation.

Important: la façon dont l'User s'est authentifié auprès de l'IdP importe peu au RP. Ce qui compte avant pour lui, c'est la fiabilité de la «confirmation d'authentification» de l'IdP avec la procédure d'authentification qui y est stipulée et les conséquences juridiques en découlant dès lors que la confirmation n'est pas fiable, autrement dit si elle comporte des erreurs.

4.4.3 Règles de responsabilité (droits et obligations)

SHOULD: les responsabilités et les droits devraient découler du type de service offert par le RP.

4.5 Entre le RP et l'IdP

4.5.1 Enregistrement

4.5.1.1 Le RP auprès de l'IdP

SHOULD NOT: un enregistrement dynamique ne devrait pas être effectué. Cette forme d'enregistrement peut contenir des vulnérabilités susceptibles d'être exploitées par un cyberpirate. De telles vulnérabilités sont décrites par exemple dans [2] pour l'enregistrement dynamique avec OpenID Connect.

MUST NOT: en cas d'exigences de sécurité élevées, pour les données personnelles sensibles par exemple, l'enregistrement ne doit pas être effectué de façon dynamique.

MUST: l'IdP doit inclure tous les identifiants du RP nécessaires à l'authentification et à l'autorisation du RP. Ceux-ci doivent être mis en relation avec les renseignements du certificat X.509 pour l'authentification TLS, autrement dit être reliés entre eux, sous forme de tableau ou par une signature par exemple.

Concernant OpenID Connect, il s'agit de Client-ID, Redirect URI, certificat X.509 TLS.

MUST: c'est la raison pour laquelle les identifiants correspondants dans le certificat et le certificat de la CA pour la vérification des certificats X.509 doivent être connus.

Remarque: «OpenID Connect Dynamic Client Registration» répertorie des attributs pouvant être enregistrés auprès de l'IdP. Toutefois, tout enregistrement dynamique devrait/doit être évité. Le document n'est qu'un simple exemple des attributs qui peuvent être enregistrés auprès de l'IdP via le RP.

4.5.1.2 L'IdP auprès du RP

MUST: le RP doit intégrer tous les identifiants de l'IdP nécessaires à l'authentification de l'IdP. Ces identifiants doivent être mis en relation avec les renseignements du certificat X.509, autrement dit être reliés entre eux, sous forme de tableau ou par une signature par exemple. Ou être contenu dans le certificat lui-même.

Avec OpenID Connect, il s'agit des points finaux publiés dans le discover endpoint. Entre autres: Issuer Identifier, URL Token (User) Endpoint, URL dans le certificat TLS X.509, certificat X.509 pour la vérification des JSON Web Token.

Les certificats X.509 ne sont pas statiques. C'est la raison pour laquelle les identifiants correspondants dans le certificat et le certificat de la CA pour la vérification des certificats X.509 doivent être connus.

4.5.2 Règle de responsabilité (droits et obligations)

SHOULD: le RP devrait savoir dans quelle mesure l'IdP doit répondre de la confirmation d'authentification.

4.5.3 Exigences concernant l'authentification

Exigence en matière de paramètres d'authentification dynamique avec OpenID Connect:

MUST: les paramètres suivants doivent être générés et utilisés de manière aléatoire dans l'environnement de la cyberadministration: State, Nonce et Code Challenge pour la procédure PKCE (RFC 7636).

MUST: les attributs nécessaires à l'authentification et à l'autorisation de l'User doivent être signés par l'IdP ou par l'instance User Info. S'ils ne sont pas signés, le RP n'a aucun document/aucune preuve à faire valoir pour savoir si les valeurs reçues de l'IdP sont correctes.

SHOULD: un cookie est créé lors de la première connexion entre l'User Agent et le RP. Le Redirect URI devrait tomber dans le champ de validité de ce cookie, de sorte qu'aucune autre connexion HTTP ne soit établie entre l'User Agent et le Redirect URI.

SHOULD: l'User devrait être informé à chaque fois par l'IdP du RP avec lequel il a l'intention d'établir une connexion. L'User devrait voir, outre l'adresse web du RP, l'emplacement et un nom distinctif. Ce nom doit lui aussi être saisi au moment de l'enregistrement du RP auprès de l'IdP.

4.5.4 A vérifier lors de l'authentification

4.5.4.1 Authentification du RP auprès de l'IdP

L'Authorization Code Flow d'OpenID Connect est utilisé afin d'exposer ce que l'IdP doit vérifier lors de l'authentification du RP.

MUST: les recommandations applicables à la norme OpenID Connect doivent être observées.

SHOULD: avec Token Request, il devrait être vérifié si l'Authorization Code attribué de façon dynamique, Client ID, concorde avec l'URL dans le certificat X.509 de RP pour l'authentification au moyen de TLS. Cela passe par une procédure mTLS au moment de l'établissement de la connexion.

Remarque: il existe des alternatives à mTLS pouvant être utilisées pour authentifier le RP. Toutefois, celles-ci n'offrent pas un niveau de sécurité aussi élevé. De plus, le RP doit détenir un Server Certificate pour pouvoir établir la connexion avec l'User.

SHOULD: le RP devrait s'authentifier auprès des serveurs gérés par l'IdP comme auprès de l'Authorization Server au moyen de mTLS.

4.5.4.2 Authentification de l'IdP auprès du RP

L'Authorization Code Flow d'OpenID Connect est utilisé afin d'exposer ce que le RP doit vérifier lors de l'authentification de l'IdP et quels identifiants du RP doivent être enregistrés auprès de l'IdP.

Token Response: à la réception du Token, le RP doit vérifier, outre la signature, si les identifiants concordent. Il s'agit de: Issuer Identifier, URL Token (User) Endpoint, URL dans le certificat TLS X.509, certificat X.509 pour la vérification des JSON Web Token.

Afin de préserver l'authenticité, les paramètres Nonce et State, dont les valeurs doivent être générées de manière aléatoire à chaque établissement de connexion, sont employés pour chaque connexion établie/échange de données.

Il faut en outre vérifier si les valeurs State et Nonce reçues concordent bien avec les valeurs de l'Authentication Request.

De plus, l'examen doit être valide selon la procédure PKCE conformément au RFC 7636 (PKCE)

4.5.5 Accord

SHOULD: L'IdP et le RP devraient s'entendre sur une Referrer Policy, voir chapitre 3.5.7 «Referrer Policy».

4.5.6 Informations concernant l'autorisation du RP par l'User

MUST: pour que le RP puisse obtenir des données personnelles concernant l'User auprès de l'IdP, l'User doit se voir indiquer, outre l'adresse web du RP, l'emplacement et un nom distinctif. Ce nom doit lui aussi être saisi au moment de l'enregistrement.

4.6 Remarque concernant la sécurité lors du déroulement du protocole

Etablir une connexion avec SAML présente l'avantage par rapport à OpenID Connect que toutes les demandes (Requests) et les réponses peuvent être signées. Une telle précaution offre davantage de protection. Toutefois, une authentification de l'User directement auprès du RP, sur la base de certificats X.509, offre en principe une bien plus grande sécurité, voir à ce sujet [11].

5 Techniques d'attaque

Les techniques d'attaque exposées ont pour principal objectif d'envoyer des commandes et de déclencher des actions au nom de l'User. Le destinataire de la commande ne peut pas voir que la commande n'a pas été envoyée par l'User.

Dans le RFC 6819, les autres menaces et contre-mesures sont répertoriées dans le cadre du protocole OpenID Connect/OAUTH.

Le chapitre 6 propose une synthèse des techniques d'attaque qui représente une menace, même dans le cadre de l'approche PKI.

5.1 Cross Side Scripting (CSS, XSS)

5.1.1 Description de l'attaque

Un cyberpirate fait en sorte qu'un site web (serveur du site web) envoie des commandes à l'User Agent et lui fait exécuter des actions. Dans la plupart des cas, le but est de permettre à un cyberpirate d'agir au nom de l'User. Par exemple, en repérant le cookie de la connexion puis en envoyant au serveur des commandes au nom de l'User.

Pour plus de détails, se reporter notamment à [5], [4], [6]. [14] OWASP: https://owasp.org/www-community/attacks/xss/.

5.1.2 Qui peut lancer l'attaque

Tout serveur (Web), avec lequel l'User Agent établit une connexion, est en position de lancer une attaque. Les victimes prédestinées d'une telle attaque sont les RP et les IdP, ne serait-ce qu'en raison de l'établissement de la connexion.

L'attaque peut également être lancée en enregistrant un fichier sur un serveur vulnérable. Ce fichier déclenche alors chez l'User Agent des commandes non souhaitées par l'User, même lorsqu'il le télécharge.

Cette attaque se distingue par le fait que l'User Agent fait trop confiance aux réponses du serveur.

5.1.3 Menace

Des (trans)actions non souhaitées par l'User sont déclenchées en son nom.

5.1.4 Contre-mesures

Le serveur prend

- un contrôle de réception des données (formats) qu'il accepte
- Contrôle initial des données (formats), quelles sont les données envoyées à l'User.

Pour plus de détails, se reporter notamment à [5], [4], [6]. Voir aussi les caractères de sécurité correspondants au chapitre 3.5 «HTTP Header».

5.2 Cross Site Request Forgery (CSRF, XSRF)

5.2.1 Description de l'attaque

Lors du chargement d'un site web (du cyberpirate), des commandes sont envoyées à une autre site web et des actions y sont déclenchées. [5], [4], [6]. Cette attaque est caractérisée par un excès de confiance de la part du serveur (site web attaqué) quant à à la Request adressée par l'User.

5.2.2 Qui peut lancer l'attaque

Tout serveur (web) avec lequel l'User Agent établit une connexion. Les RP et l'IdP sont prédestinés à être victimes d'une telle attaque, par le simple fait qu'une connexion soit établie.

5.2.3 Menace

La menace consiste en des (trans)actions non souhaitées par l'User qui sont déclenchées en son nom.

5.2.4 Contre-mesures

Utilisation de «Tokens cachés» (CSRF Token) dans le http Header, de sorte qu'il soit possible de savoir si la commande ou la requête provient de l'User Agent ou bien d'un tiers. Autres mesures: Double Submit Cookie, SameSite Cookie

5.3 Server Site Request Forgery

5.3.1 Description de l'attaque

La technique SSRF (Server-Site-Request-Forgery) consiste à amener un serveur à déclencher des requêtes du cyberpirate. Ces requêtes peuvent être transmises aux serveurs ou à d'autres serveurs. Lorsque les requêtes sont envoyées à un système tiers, un cyberpirate peut s'en servir pour faire croire que le serveur en est l'auteur. Pour plus de détails, voir [10].

5.3.2 Qui peut lancer l'attaque

Toute personne en mesure d'établir une connexion avec le serveur à attaquer.

5.3.3 Menace

P. ex. l'exécution non souhaitée de commandes et la divulgation non souhaitée de données sensibles.

5.3.4 Contre-mesures

Une White Liste énumérant les commandes qu'une application peut envoyer. Pour plus de détails, voir [10].

5.4 UI Redressing (Click Jacking)

5.4.1 Description de l'attaque

Une tierce partie (cyberpirate) pousse l'User à envoyer des commandes sur un autre site web ou à y déclencher des actions. Par exemple, un User pense adresser des commandes ou déclencher les actions sur le site web du tiers, alors qu'en réalité il le fait sur le site web d'un autre domaine. Par exemple, il pense cliquer sur un bouton sur le site web du tiers, alors qu'en réalité il confirme une action sur un autre site web.

Le site web présume que l'action a été effectuée par l'User en toute connaissance de cause. Pour plus de détails, se reporter à [5], [4].

5.4.2 Qui peut lancer l'attaque

Tout serveur (web) avec lequel l'User Agent établit une connexion. Le RP et l'IdP sont prédestinés à être victimes d'une telle attaque, par le simple fait qu'une connexion soit établie.

5.4.3 Menace

Saisie de commandes non souhaitées sur un site web par l'User

5.4.4 Contre-mesures

Insérer «X-Frame-Options» dans le http Response Header ou «frame-ancestors» dans la politique de sécurité du contenu du http Header. Les navigateurs ne prennent en charge/suivent cette instruction dans le http Response Header qu'à partir d'une certaine version.

MUST: seules les versions de navigateur qui comprennent et suivent les commandes dans les «X-Frame-Options» ou dans les «frame-ancestors» dans la Content Policy doivent être prises en charge.

Autres sources: X-Frame-Options – HTTP, OWASP Secure Headers Project – OWASP

5.5 Callback Function

La notion de «Callback Function» n'est pas sans ambigüité. On entend par là ce qui suit: une ou plusieurs fonctions dans un logiciel, en abrégé fonction, sont transmises en tant que paramètres pour une autre fonction.

Cible de l'attaque: Contournement de la Same Origin Policy (SOP). L'environnement dans lequel une fonction est/peut être exécutée est transféré dans l'environnement cible souhaité. L'environnement d'origine est appelé Callback URL.

5.5.1 Qui peut lancer l'attaque

Tout serveur (web) avec lequel l'User Agent établit une connexion.

5.5.2 Menace

Exécution non souhaitée de code dans l'User Agent

5.5.3 Contre-mesures

Aucun Cross Resource Sharing (CORS) sur le site web, Callback statique, White List du Callback possible. Pour plus de détails, voir [9].

5.6 Code Injection dans le token

5.6.1 Description de l'attaque

Dans SAML, ou mieux XML, ou dans un JSON Web Token, il peut y avoir du code exécutable comme ActiveX, JavaScript ou des Java Applets, ou un lien vers ceux-ci. Cela peut avoir pour effet que des programmes soient exécutés contre le gré du destinataire du Token (RP).

Cette opération est à distinguer de l'Authorization Code Injection. Dans ce cas de figure, un Authorization Code est injecté, voir à ce sujet [13]. Voir chapitre 4.5.1 concernant la description des attaques.

5.6.2 Qui peut lancer l'attaque

Tout serveur (web) qui peut émettre des SAML ou JSON Web Tokens que reçoit le serveur cible ou l'User Agent.

5.6.3 Menace

Emission d'ordres non souhaités dans l'User Agent ou sur le serveur cible.

5.6.4 Contre-mesures

Contre-mesures à la Code Injection: Vérification correspondante du contenu dans le Token. Contre-mesures à l'Authorization Code Injection, voir [13] chapitre 4.5.3.

5.7 Déplacement de la signature XML dans un SAML Token

5.7.1 Description de l'attaque

Une signature et le contenu qu'elle englobe (sous-objets) sont déplacés dans un objet XML. Le contenu (les sous-objets) est remplacé par un autre (d'autres objets). Si maintenant la vérification de la signature est dissociée de la vérification du contenu, les sous-objets remplacés peuvent être acceptés, car la signature a été vérifiée avec succès, même si elle a été déplacée dans l'objet principal. Le déplacement de la signature et des sous-objets qu'elle englobe n'a pas rendu la signature caduque. Or elle ne protège plus ce que l'application reçoit. Pour plus de détails, se reporter à eCH-0091, page 17 colonne 2, [7], [8]. L'attaque est appelée Signature Wrapping.

5.7.2 Menace

Acceptation non souhaitée du contenu d'une Request ou d'une Response par le RP ou l'IdP.

5.7.3 Contre-mesures

eCH-0091, 2.0.0. page 17 «Interaction entre l'application XML et la vérification de la signature XML», [7], [8].

5.8 Certificate Impersonation

5.8.1 Description de l'attaque

L'User est amené à accepter une connexion TLS fondée sur un certificat de serveur qui n'a pas été émis par la CA requise.

p. ex.: Un certificat avec l'entrée www.admin.ch n'est pas délivré par l'Admin PKI, mais par la CA Trust Me. Le certificat racine de cette CA est inclus dans le navigateur.

5.8.2 Menace

Une connexion TLS est établie involontairement et des données confidentielles sont échangées avec un serveur sans que cela soit voulu.

5.8.3 Qui peut lancer l'attaque

Toute CA dont le certificat racine dans le navigateur est jugé digne de confiance.

5.8.4 Contre-mesures

MUST: afin que seuls les certificats TLS souhaités soient utilisés pour authentifier le serveur, l'IdP doit présenter une entrée CAA dans le DNS. Qui plus est, il doit contrôler la délivrance des certificats de son domaine.

SHOULD: le RP devrait également procéder de la sorte.

Source: RFC 6844

5.9 Denial of Service

5.9.1 Description de l'attaque

Un flot de demandes adressées à un serveur en réduit, voire paralyse la capacité de communication.

5.9.2 Menace

La disponibilité de la communication avec le serveur s'en trouve affectée. L'IdP faisant figure de charnière pour l'authentification, réduire sa capacité de communication se traduirait par une dégradation, voire une rupture de la communication entre un User et le RP.

5.9.3 Contre-mesures

MUST: l'IdP doit mettre en œuvre des mesures efficaces visant à se protéger contre les différentes attaques de type Denial of Service, p. ex. au moyen de Rate-Limit, Captcha, IP Tracking.

5.10 Brute Force / Exploration

5.10.1 Description de l'attaque

Les attaques de type Brute Force cherchent à déterminer le nom d'utilisateur et le mot de passe ou un Token afin d'obtenir un accès non autorisé. Les attaques de type Brute Force reposent sur des tentatives et des erreurs, qui consistent à deviner puis à vérifier les informations souhaitées.

Les attaques de type exploration visent notamment à sonder les comportements des participants à la communication et à en tirer des enseignements.

5.10.2 Menace

Autorisation d'accès ou consultation involontaire des données par un tiers.

5.10.3 Qui peut lancer l'attaque

Toute personne en mesure d'établir une connexion avec le serveur.

5.10.4 Contre-mesures

MUST: l'IdP doit mettre en œuvre les mesures suivantes pour se protéger contre les attaques de type Brute Force et les attaques par exploration: Rate-Limit, Captcha, IP Tracking.

6 Comparaison approche PKI – OIDC et SAML

[11] propose une comparaison entre une approche PKI et le modèle IAM sous-jacent composé d'un User Agent, d'un RP et d'un IdP. L'approche PKI a plusieurs arguments majeurs a faire valoir en matière de sécurité:

- Disponibilité élevée
- Etablissement d'une connexion moins complexe (l'approche PKI implique un composant de moins). Les recommandations concernant le RP au chapitre 4 «Enregistrer et authentifier» sont donc sans importance.
- La traçabilité devient alors plus facile.
- L'User a moins de vérifications à effectuer.
- Mauvaise prise en charge des navigateurs (User Experience)
- Meilleures performances de connexion

Pour des explications à ce sujet ainsi que d'autres avantages propres à l'approche PKI en matière de sécurité réseau, se reporter à la source mentionnée. Les recommandations dans les chapitres suivants sont elles aussi pertinentes pour une approche purement PKI:

- Chapitre 3 «Recommandations générales»
- Chapitre 5 «Techniques d'attaque»

7 Condensé

7.1 Atteinte à l'authenticité d'une e-ID et de l'autorisation

Les attaques suivantes peuvent contourner l'authentification avec des identités électroniques (E-ID) du fait d'un manque de précautions, sans que l'on puisse reprocher au détenteur d'une E-ID un défaut de vigilance dans l'utilisation de l'E-ID:

- Manque de prudence au moment d'enregistrer la création de l'identité électronique
- CSS/XSS
- CSRF/XSRF
- UI-Redressing (Click Jacking)
- Session Hijacking
- Code Injection et Authorization Code Injection
- Déplacement de la signature XML dans un SAML Token (XML Signature Wrapping)

Les précautions/mesures contre les attaques évoquées doivent être prises par les serveurs, sans que l'User n'ait la moindre influence dessus. L'User non avisé en matière de sécurité informatique ne peut prétendre savoir si les mesures de sécurité ont bien été mises en œuvre.

7.2 A vérifier par l'User

L'User doit vérifier les points suivants:

- URL du RP
- URL de l'IdP (qu'il ne soit pas redirigé vers un autre site web et qu'il n'y saisisse son mot de passe).
- Si un Frame séparé du RP est ouvert pour la connexion à l'IdP.
- Si, lorsque l'IdP demande son accord pour envoyer les données au RP, l'URL qui y figure correspond à l'URL du RP sélectionné.

Remarque: L'User a davantage de points à vérifier au moment où une connexion est établie qu'avec une approche purement PKI ! Il devient alors plus difficile de procéder aux vérifications requises et de prouver qu'elles ont été effectuées ou non.

7.3 Menace/vulnérabilité

Quasiment tout tournant autour du thème ou de la menace «Impersonation», un tableau comparatif des vulnérabilités/menaces n'aboutit pas à grand-chose. à l'exception du chapitre:

- 5.9 «Denial of Service»
- 5.10 «Brute Force / Exploration»

Le chapitre 5 «Techniques d'attaque» répertorie les menaces/attaques possibles et préconise les contre-mesures à cet effet.

8 Exclusion de responsabilité – droits de tiers

Les normes élaborées par l'Association **eCH** et mises gratuitement à la disposition des utilisateurs ainsi que les normes de tiers adoptées, ont seulement valeur de recommandations. L'Association **eCH** ne peut en aucun cas être tenue pour responsable des décisions ou mesures prises par un utilisateur sur la base des documents qu'elle met à disposition. L'utilisateur est tenu d'étudier attentivement les documents avant de les mettre en application et au besoin de procéder aux consultations appropriées. Les normes **eCH** ne remplacent en aucun cas les consultations techniques, organisationnelles ou juridiques appropriées dans un cas concret.

Les documents, méthodes, normes, procédés ou produits référencés dans les normes **eCH** peuvent le cas échéant être protégés par des dispositions légales sur les marques, les droits d'auteur ou les brevets. L'obtention des autorisations nécessaires auprès des personnes ou organisations détentrices des droits relève de la seule responsabilité de l'utilisateur.

Bien que l'Association **eCH** mette tout en œuvre pour assurer la qualité des normes qu'elle publie, elle ne peut fournir aucune assurance ou garantie quant à l'absence d'erreur, l'actualité, l'exhaustivité et l'exactitude des documents et informations mis à disposition. La teneur des normes **eCH** peut être modifiée à tout moment sans préavis.

Toute responsabilité relative à des dommages que l'utilisateur pourrait subir par suite de l'utilisation des normes **eCH** est exclue dans les limites des règlementations applicables.

9 Droits d'auteur

Tout auteur de normes **eCH** en conserve la propriété intellectuelle. Il s'engage toutefois à mettre gratuitement, et pour autant que ce soit possible, la propriété intellectuelle en question ou ses droits à une propriété intellectuelle de tiers à la disposition des groupes de spécialistes respectifs ainsi qu'à l'Association **eCH** pour une utilisation et un développement sans restriction dans le cadre des buts de l'association.

Les normes élaborées par les groupes de spécialistes peuvent, moyennant mention des auteurs **eCH** respectifs, être utilisées, développées et déployées gratuitement et sans restriction.

Les normes **eCH** sont complètement documentées et libres de toute restriction relevant du droit des brevets ou de droits de licence. La documentation correspondante peut être obtenue gratuitement.

Les présentes dispositions s'appliquent exclusivement aux normes élaborées par **eCH**, non aux normes ou produits de tiers auxquels il est fait référence dans les normes **eCH**. Les normes incluront les références appropriées aux droits de tiers.

Annexe A – Références & bibliographie

Bibliographie

[1]	Annexe 8 Ordonnance de l'EDI du 22 mars sur le dossier électronique du patient.
[2]	Vladislav Mladenow, Christian Mainka, OpenID Connect Security Consideration, Ruhr Universität Bochum, 2017
[3]	Cross-Site Request Forgery Prevention Cheat Sheet, https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html#synchronizer-token-pattern
[4]	Prof. Dr. Jörg Schwenk, Netzwerksicherheit 3, Ruhr Universität Bochum, 5. Auflage, 2017
[5]	Darfydd Stuttard, Marcus Pinto, Hacker's Handbook, 2nd edition, Wiley, 2011
[6]	Bryan Sullivan, Vincent Liu, Web Application Security, McGraw Hill, 2012
[7]	Michael MacIntosh, Paula Austel, XML Signature Wrapping Attacks and Countermeasures
[8]	Juray Somorovsky, Andreas Mayer at al, On Breaking SAML: Be Whoever You Want to be
[9]	Ben Hayak, Same Origin Method Execution (SOME), Exploiting A callback for Same Oirigin Policy Bypass, Nov. 2014
[10]	Andrea Hauser, Server-Site-Request-Forgery, Was es ist und wie man sich schützen kann, https://www.scip.ch/?labs.20200618
[11]	Florian Forster, Daniel Muster, Vergleich von online Authentisierungen im eGov Bereich, 13 octobre 2020 http://www.it-rm.ch/files/Technologie_Vergleich_1_2.pdf
[12]	On Breaking SAML: Be Whoever You Want to Be, Juraj Somorovsky, Andreas Mayer, Jörg Schwenk, Marco Kampmann, and Meiko Jensen
[13]	OAuth Security Best Practices, https://tools.ietf.org/id/draft-ietf-oauth-security-topics-13.html
[14]	Cross Site Scripting, OWASP: https://owasp.org/www-community/attacks/xss/.

Normes

eCH-0091	Norme de signature et chiffrement XML / 2.0.0
ECRYPT II	European Network of Excellence in Cryptology, ECRYPT II Yearly Report on Algorithms and Keysizes, (2011-2012), Revision 1.0, 30. Sept 2012
ETSI TS 119	Electronic Signatures and Infrastructures (ESI); Cryptographic Suites

312 V1.1.1	
FIPS PUB 180-4	Secure Hash Standard (SHS)
ITU X.509	Telecommunication Standard de l'ITU pour les certificats
NIST 800-63-3	NIST Special Publication 800-63-3, Digital Identity Guidelines
NIST 800-81-2	Secure Domain Name System (DNS) Deployment Guide
NIST 800-90C	Recommendation for Random Bit Generator (RBG) Constructions
OpenID Client Registration	OpenID Connect Dynamic Client Registration 1.0 incorporating errata set 1, OpenID Foundation
OPENID Connect	OpenID Connect Core 10 incorporating errata set 1, OpenID Foundation
Referrer-Policy	Referrer Policy, W3C Candidate Recommendation, 26 January 2017
RFC 5246	TLS 1.2
RFC 6749	The OAUTH 2.0 Authorization Framework, IETF
RFC 6750	OAUTH 2.0 Authorization Framework: Bearer Token Usage
RFC 6819	OAUTH 2.0 Threat Model and Security Consideration,
RFC 6819	OAUTH Threat Model and Security Consideration
RFC 7159	JavaScript Object Notation (JSON) Data Interchange Format
RFC 7515	JSON Web Signature,
RFC 7636	Proof Key for Code Exchange by OAUTH Public Client
RFC 8446	TLS 1.3
SAML	Security Assertion Markup Language (SAML) V2.0 Technical Overview. OASIS
XML	Extended Markup Language, Published, W3C recommendation

Annexe B – Collaboration & vérification

<Toutes les collaboratrices et tous les collaborateurs ayant contribué à cette version du document sont répertorié(e)s ici.>

Florian Forster	ZITADEL AG
Daniel Muster	it-rm IT-Riskmanagement GmbH

Annexe C – Abréviations et glossaire

AES	Advanced Encryption Standard
Menace	Un risque qui devient réalité
COEP	Cross Origin Embedder Policy (COEP)
COOP	Cross Origin Opener Policy
CORB	Cross Origin Read Blocking (CORB)
CORP	Cross Origin Resource Policy (CORP)
CORS	Cross Origin Resource Sharing
CSP	Certificate Service Provider
CSRF	Cross Site Request Forgery
CSS	1. Cascading Style Sheet 2. Cross Site Scripting
DDoS	Distribute Denial of Service Attack
DoS	Denial of Service Attack
EC	Elliptic Curve
ETSI	European Telecommunications Standards Institut
EU	European Union
Risque	Possibilité qu'un événement indésirable se produise.
HMAC	hash-based message authentication code
HSTS	HTTP Strict Transport Security
HTTP	Hypertext Transfer Protocol
IAM	Identity and Access Management
IdP	Prestataire de services d'identité électronique
ITU	International Telecommunication Union
JSON	JavaScript Object Notation (JSON) Data Interchange Format
JWT	JSON Web Token
MAC	Message Authentication Code
MIME	Multipurpose Internet Mail Extensions
mTLS	Mutual Transport Layer Security. Client et serveur s'authentifient au moyen du protocole TLS basé sur des certificats X.509.

OASIS	Organization for the Advancement of Structured Information Standards
OAUTH	Open Authorization
OIDC	OpenID Connect
OWASP	Open Web Application Security Project
PKCE	Proof Key for Code Exchange
PKI	Public Key Infrastructure
RP	Une application informatique qui utilise un IdP afin de procéder à l'authentification (identification électronique) de l'User.
RSA	Public Key Verfahren von Rivest, Shamir et Adleman
SAML	Security Assertion Markup Language
Vulnérabilité	Une vulnérabilité est, comme son nom l'indique, une situation qui présente une ou plusieurs faiblesses et qui est donc susceptible de provoquer un préjudice. Une vulnérabilité résulte notamment d'un défaut ou d'une insuffisance de vigilance (action ou omission). Quant à savoir si quelque chose est considéré comme une vulnérabilité et si des mesures doivent/devraient être prises afin d'y remédier, il convient de s'en remettre à la réglementation ou à la gestion des risques.
SOP	Same Origin Policy
SSI	Self Sovereign Identity
SSL	Secure Socket Layer
TLS	Transport Layer Security
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extended Markup Language
XSRF	Cross Site Request Forgery
XSS	Cross Site Scripting

Annexe D – Modifications par rapport à la version précédente

Il s'agit de la première version.

Annexe E – Liste des illustrations

Aucun

Annexe F – Liste des tableaux