eCH-0251 – Web-Sicherheit im IAM Bereich

Name	Web-Sicherheit im IAM Bereich
eCH-Nummer	eCH-0251
Kategorie	Hilfsmittel
Reifegrad	Definiert
Version	1.0.0
Status	Genehmigt
Beschluss am	2024-01-08
Ausgabedatum	2023-12-12
Ersetzt Version	–
Voraussetzungen	-
Beilagen	-
Sprachen	Deutsch (Original), Französisch (Übersetzung)
Autoren	Fachgruppe IAM Florian Forster (ZITADEL AG) Daniel Muster (it-rm IT-Riskmanagement GmbH)
Herausgeber / Vertrieb	Verein eCH, Räffelstrasse 20, 8045 Zürich T 044 388 74 64, F 044 388 71 80 www.ech.ch / info@ech.ch

Zusammenfassung

Die FG IAM hat hiermit ein Hilfsmittel zur Web-Sicherheit im IAM-Bereich erstellt, welches als notwendige Ergänzung zu den anderen eCH-Standards in diesem Bereich zu verstehen ist.

Das Hilfsmittel enthält Empfehlungen zur Websicherheit für die IAM-Protokolle «SAML» und «OpenID Connect», um u.a. die im Protokoll enthaltenen Schwachstellen zu verkleinern und somit die Risiken zu minimieren. Die Schwachstellen resultieren u.a. daraus, dass die gängigen Standards dazu die hier thematisierten Schwachstellen/Risiken nicht erwähnen oder nur Teile davon behandeln, was zu Schwachstellen bei der Implementierung führen kann. Die Empfehlungen in diesem Hilfsmittel basieren einerseits auf international anerkannten Standards, anderseits auf wissenschaftlichen Veröffentlichungen.

Einige der hier aufgeführten Empfehlungen lassen sich auch auf andere Bereiche der Netzwerksicherheit anwenden. Hier sind Empfehlungen zur Systemsicherheit jedoch nicht enthalten.

Inhaltsverzeichnis

1 Einleitung

1.1 Status

Genehmigt: Das Dokument wurde vom Expertenausschuss genehmigt. Es hat für das definierte Einsatzgebiet im festgelegten Gültigkeitsbereich normative Kraft.

1.2 Ziel des Dokuments

Dieses Dokument ist ein Hilfsmittel für Behörden und Private im Bereich Internet und Access Management Sicherheit, kurz IAM-Sicherheit, im Government Umfeld. Das Hilfsmittel soll unter anderem einer öffentlichen Ausschreibung oder einer Inbetriebnahme einer (IAM)-Komponente dienen. Es werden hierzu (unverbindliche) Massnahmen zur Web-Sicherheit und dazugehörige organisatorische Massnahmen empfohlen.

Das hier zu Grunde liegende einfache Modell basiert auf folgenden 3 IAM-Komponenten.

- User Agent und User
- Relying Party (Definition des Begriffs, s. Glossar)
- Identity Provider (Definition des Begriffs, s. Glossar)

Annahme: Die oben genannten Komponenten werden von jeweils unterschiedlichen autonomen Körperschaften verwaltet. Z.B. Der User ist Kunde der Bank (RP) und meldet sich dort über einen IdP an, welcher von einer Behörde verwaltet wird.

Die hier vorgestellten Empfehlungen sind auch anwendbar, wenn zwei oder alle Komponenten von der gleichen autonomen Körperschaft verwaltet werden. Z.B., wenn die RP und der IdP von der gleichen Behörde administriert werden.

Die IT-Sicherheitsmassnahmen dienen zur Minderung der Bedrohung/Gefahren (engl. threat), resp. der Schwachstellen (Definition des Begriffs, siehe Glossar). Die hier aufgeführten Bedrohungen/Gefahren basieren auf RFC 6819. Wie erwähnt, werden lediglich die Aspekte der Web-Sicherheit im IAM Umfeld und der damit unmittelbar verbundenen organisatorischen Massnahmen behandelt. Einige der dabei abgegebenen Empfehlungen lassen sich auch auf andere Bereiche der Web-Sicherheit anwenden, wie die in den Kapiteln 3.1, 3.2 oder 4.2.

Die Empfehlungen enthalten:

- Gegenmassnahmen zu Angriffen und den damit verursachten Bedrohungen
- sowie Literaturhinweise.
- Tätigkeiten oder Unterlassungen zur Minderung von Schwachstellen
- sowie Literaturhinweise.

Z.B. Impersonation (Angriff), Bedrohung für die Authentizität, Massnahmen: Signatur RSA 2048 Bit SHA-512 Bit (als Gegenmassnahme).

Dieses Dokument adressiert IT-Sicherheitsfachleute. Falls überhaupt, werden Empfehlungen, Schwachstellen und Angriffe nur rudimentär erklärt. Jedoch werden Quellenangaben zum Thema beigefügt.

Die Empfehlungen in diesem Dokument und die Erläuterungen dazu beschränken sich auf folgende IAM Technologien:

- SAML
- OpenID Connect

Nicht enthalten sind Empfehlungen zur Systemsicherheit.

1.3 Abgrenzung

Es gibt andere Modelle/Konzepte rund um die elektronische Identifizierung (Authentisierung) und der damit verbundenen Dienstleistungen, wie Self Sovereign Identity (SSI) oder einen reinen „PKI-Ansatz“. Zu den Begriffen SSI und „PKI-Ansatz“, s. Glossar. Viele der hier gemachten Empfehlungen lassen sich auch auf SSI oder auf den „PKI-Ansatz“ anwenden. Welche genau für den PKI-Ansatz werden im Kapitel 6 «Vergleich PKI-Ansatz – OIDC und SAML» aufgelistet.

1.4 Terminologie der Empfehlungen

Richtlinien in diesem Dokument werden gemäss der Terminologie aus [RFC 2119] angegeben, dabei kommen die folgenden Ausdrücke zur Anwendung, die durch **GROSSSCHREIBUNG** als Wörter mit den folgenden Bedeutungen kenntlich gemacht werden (Zitat aus [RFC 2119]):

- **MUST:** This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- **MUST NOT:** This phrase, or the phrase "SHALL NOT" mean that that definition is an absolute prohibition of the specification.
- **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.
- **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.
- **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.)

1.5 Auswahl der Standards

Die hier vorgestellten Empfehlungen sind facettenreich, sodass nicht eine dem Thema entsprechende und umfassende Auswahl an Standards vorgenommen werden kann. Jedoch wird hier auch auf wissenschaftliche Publikationen verwiesen.

2 Überblick/Aufbau des Dokuments

2.1 Zielpublikum

Das Dokument richtet sich an IT-Sicherheitsfachleute im Bereich IAM- und Web-Sicherheit.

2.2 Fokus bei den Empfehlungen

Der Fokus der hier vorliegenden Empfehlungen liegt auf IT-Sicherheit, im Wesentlichen auf die Verhinderung des elektronischen Identitätsdiebstahls, und nicht auf Funktionalität oder Anwenderfreundlichkeit. Sicherheit oder Funktionalität und Anwenderfreundlichkeit stellen oft entgegengesetzte Zielsetzungen bei gleichem Preis dar.

2.3 Inhalt der einzelnen Kapitel

Der Fokus in diesem Dokument liegt im Wesentlichen auf dem Thema Impersonation (Identitätsdiebstahl). Die hier behandelten Unterthemen sind wie folgt gegliedert:

- Kapitel 3 «Allgemeine Empfehlungen». Wie der Titel verrät, werden allgemeine Empfehlungen zur Verhinderung des elektronischen Identitätsdiebstahls abgegeben. Diese Empfehlungen lassen sich auch auf andere Netzwerkkonstellationen anwenden.
- Kapitel 4 «Registrieren und Authentisieren». Das Augenmerk richtet sich auf die Abläufe bei der Authentisierung (elektronischen Identifizierung) und der dafür benötigten Attribute und Parameter und was bei der Authentisierung geprüft werden sollte.
- Kapitel 5 «Angriffstechniken» Verschiedene Angriffstechniken werden vorgestellt, wie ein elektronischer Identitätsdiebstahl erreicht werden kann. Zudem werden die jeweiligen Gegenmassnahmen präsentiert.
- Kapitel 6 «Vergleich PKI-Ansatz – OIDC und SAML». Hier wird ein Sicherheitsvergleich, besser ein Verweis auf einen Sicherheitsvergleich, zwischen diesen 3 Authentisierungsverfahren eines Users vorgenommen. Dabei wird auch aufgeführt, welche der hier abgegebenen Empfehlungen für einen reinen PKI-Ansatz relevant sind.

3 Allgemeine Empfehlungen

In diesem Kapitel werden Empfehlungen allgemeiner Natur abgegeben, um die Gefahr aus Schwachstellen zu verkleinern.

Anmerkung: Falls keine Differenzierung zwischen IdP und RP vorgenommen wird, dann gilt die Empfehlung sowohl für die RP als auch für den IdP.

3.1 Kryptographie

3.1.1 Asymmetrische Kryptographie

Welche Verfahren (Algorithmen) empfohlen werden und wie sie zu parametrisieren sind, ist in ETSI TS 119 312 aufgeführt. In ENISA ECRYPT II, Kapitel 6, ist die geschätzte Komplexität zum Brechen des jeweiligen Verfahrens den Schlüssellängen gegenübergestellt. Im Unterschied zu anderen Standards werden dort nicht Empfehlungen zur Verwendungsdauer eines privaten Schlüssels angegeben, sondern die geschätzte Komplexität, um das Verfahren bei entsprechender Schlüssellänge zu brechen.

Somit kann der Anwender die Rechenleistung/deren Kosten der Komplexität gegenüberstellen und die Verwendungsdauer des Schlüssels gemäss seinen Vorgaben des Risikomanagements selber bestimmen.

Anmerkung zur Signatur mit diskretem Logarithmus oder Elliptischen Kurven: Im Unterschied zu RSA ist die Qualität des Zufallszahlengenerators bei der Bildung einer Signatur mit Elliptischen Kurven oder mit diskretem Logarithmus Verfahren (Diffie-Hellmann) eminent wichtig. Ist der Zufallsgenerator nicht von ausreichender Qualität, so können Rückschlüsse von der geleisteten Signatur auf den privaten Schlüssel gezogen werden.

SHOULD: Dieser Aspekt soll bei der Wahl eines externen Aufbewahrungsmediums zur Herstellung der Signatur (HSM, Crypto Card) berücksichtigt werden.

Die Anforderung an die Qualität des Zufallszahlengenerators könnte es unter Umständen verbieten, Crypto Cards mit Elliptischen Kurven einzusetzen.

Generierung: In Kapitel 6.1.2 ETSI TS 119 312, V1.1.1 sind Kriterien für die Generierung der Parameter für die jeweiligen Verfahren (RSA, Diskreter Logarithmus, Elliptische Kurven) aufgeführt.

3.1.2 Symmetrische Kryptographie

SHOULD NOT: Es sollen unterschiedliche Schlüssel zum Schutz der Authentizität (z.B. HMAC) und zum Schutz der Vertraulichkeit (z.B. Verschlüsselung) verwendet werden.

Der Einsatz des gleichen Schlüssels für die Authentizität und die Verschlüsselung stellt eine kryptographische Schwäche dar. Deswegen:

SHOULD NOT: Galois Counter Mode soll nicht verwendet werden.

3.1.3 Zufallszahlengenerator

MUST NOT: Dual EC Verfahren von NIST Special Publication 800-90, edition 2007, darf nicht verwendet werden.

In Kapitel 8.2.3 ETSI TS 119 312, V1.1.1 sind Kriterien für die Generierung der Zufallszahlen aufgeführt.

3.1.4 Netzwerkprotokolle (SSL/TLS)

SHOULD NOT: TLS v1.3 soll (bis zur Klärung der Risiken oder der genannten Schwachstellen) nicht verwendet werden.

SHOULD: TLS v1.2 soll verwendet werden, Cipher: AES 256, MAC mit SHA-1 oder SHA-256

TLS v1.3 hat gegenüber TLS v1.2 folgende kryptographische Schwächen:

- Die Verschlüsselungsverfahren beschränken sich auf 2 Algorithmen (auf AES und auf ein nicht allgemein bekanntes Verfahren).
- Es bestehen lediglich 5 Cipher Suites, wovon 4 mit AES.
- Bei jedem der Verfahren wird für die Verschlüsselung und für die kryptographische Prüfsumme (Schutz der Integrität und Authentizität) der gleiche Schlüssel verwendet.
- Die Länge des Prüfsummenwerts beträgt lediglich 128 Bit (SHA-1 hat 160 Bit).
- Die Parameter der asymmetrischen Kryptographie sind vorgegeben und können nicht verändert/konfiguriert werden.
- Ein Komprimierungsalgorithmus kann gegenüber den vorangehenden Versionen nicht mehr auf Ebene TLS implementiert werden.

Anmerkung: Obwohl SHA-1 bei der Verwendung von langfristig gültigen Signaturen nicht mehr verwendet werden soll, kann das Verfahren in diesem Zusammenhang eingesetzt werden, weil die jeweiligen SHA-1 Werte nur eine kurze Gültigkeit (Dauer der TLS Session) aufweisen.

3.2 X.509 Zertifikate

SHOULD: Zur Authentisierung der Server sollen nach ZertES geregelte Zertifikate verwendet werden.

SHOULD: Für das Authentisieren des Servers, für die Bildung der Token-Signaturen und für die Entschlüsselung der Token sollen je unterschiedliche private Schlüssel verwendet werden.

SHOULD: Falls auf Basis eines Zertifikats ein Client authentisiert wird, dann soll dies mit einem nach ZertES geregelten Zertifikat erfolgen.

Grund: Bei nach ZertES geregelten Zertifikaten sind Haftung und Sorgfaltspflicht per Gesetz vorgeschrieben. Dies bietet mehr Rechtssicherheit. Bei Behördenanwendungen ist diese Art der Haftung der Haftung in Ausübung des Amtes im Allgemeinen vorteilhafter für die Behörde.

3.3 Session (Hijacking)

Eine Session ID verkörpert Authentisierung und Autorisierung einer Entität bei SSL/TLS. Die Übernahme oder das Stehlen der Session einer Entität durch einen Angreifer hat entsprechende Auswirkungen. Der Angreifer erhält dadurch die gleichen Rechte wie die Entität bei der Kommunikation.

Ziel vieler Angriffe ist es, die Session (Verbindung) eines anderen zu übernehmen und dadurch in dessen Namen, d.h. mit dessen Rechten, Befehle abzusetzen. In Folgendem beschränken sich die Empfehlungen zur Verringerungen der Verletzlichkeit einer Verbindung (Session) auf HTTP-Verbindungen.

3.3.1 Link zwischen Anwendungen und Sicherheitstechnologie

Auf TLS-Ebene (TLS Session ID) und Applikationsebene (HTTP mit Session Cookie) werden verschiedene Session IDs verwendet. Eine solche Trennung zwischen Sicherheitstechnologie und der damit zu schützenden Anwendung bietet Schwachstellen, welche durch einen Angriff ausgenutzt werden kann. Ein möglicher Angriff stellt das Extrahieren eines Cookie dar.

Beispiel: Sind TLS Session ID und Cookie nicht miteinander verlinkt und wird nicht geprüft, ob die Verlinkung stimmt, dann kann sich der Angreifer alleine mit dem Cookie als jemanden anderen ausgeben.

SHOULD: Es soll ein virtueller Link (Channel Binding) zwischen der TLS Session ID und dem vom Web-Dienst gesetzten HTTP Session Cookie hergestellt werden.

Falls ein solcher Link aufgebaut wird, dann:

MUST: Stimmt die über den Link hergestellte Relation zwischen HTTP Session Cookie und TLS Session ID nicht überein, dann muss die Verbindung zurückgesetzt/abgebrochen werden.

Das HTTP Session Cookie dient dem Server zur Unterscheidung der verschiedenen HTTP Verbindungen. Die TLS-Session ID für die Unterscheidung der verschiedenen TLS-Verbindungen. Wird kein virtueller Link zwischen Applikation (HTML, HTTP) und Sicherheitstechnologie (TLS) hergestellt, kann die Sicherheit (einer TLS-Verbindung) umgangen werden:

Zur Illustration: Die HTTP Verbindung A wird über TLS Session a aufgebaut. Dabei wird das HTTP Cookie â verwendet. Besteht nun kein entsprechender virtueller Link, kann über die TLS Verbindung B das Session Cookie â eingeschleust (Cookie Injection) und damit die HTTP Verbindung A übernommen werden.

Es ist egal, auf Basis von welchem Kriterium eine HTTP Session unterschieden wird, z.B. mit einem JSON Web Token. Der Link zwischen der Sicherheitstechnologie und der Anwendung soll jedoch hergestellt und muss dann kontrolliert werden.

Weiteres Beispiel für die notwendige Verlinkung zwischen Anwendung und Sicherheit bei der XML-Signatur, siehe [12].

3.3.2 HTTP Session Cookies

Session Cookies verkörpern die Session ID der HTTP-Verbindung. Ziel einiger Angriffe ist es, Kenntnis darüber zu erlangen, d.h. die Session ID in Erfahrung zu bringen. Aus Sicht des Angreifers ist dies besonders lohnenswert, wenn kein Link zwischen der SessionID der Applikation und der TLS SessionID besteht.

Der Server hat eine grosse Auswahl an Parametern in einem Cookie. Zum Schutz der Session Cookies und somit der Session werden folgende Empfehlungen abgegeben:

MUST: Der Wert des Cookie (Value) muss zufällig generiert werden und mindestens 128 Bit lang sein, damit der Wert nicht vorhersehbar ist.

MUST: Die Gültigkeitsdauer (Expire) eines Cookies muss gesetzt werden.

MUST: Cookie soll nur für die angewählte Domain (www-Adresse) zurückgesandt werden dürfen.

SHOULD: Cookies sollen einzig für die angewählte Domain (Path) gültig sein.

MUST: Cookies dürfen nur über verschlüsselte Verbindungen (Secure) ausgetauscht werden.

MUST: Cookies dürfen nur über HTTP-Verbindungen (HttpOnly) ausgetauscht werden.

MUST: Session Cookies müssen gegenüber Einsicht unberechtigter Dritter geschützt sein.

Anmerkung: Ein Speichern der Cookies kann für die Nachvollziehbarkeit der Abläufe sinnvoll sein.

3.3.3 Allgemeine Empfehlungen zu Session

3.3.3.1 Dauer einer Verbindung

MUST: Eine Verbindung ist immer nach einer definierten Zeit zu erneuern, dies unabhängig von den Aktivitäten auf der Verbindung.

MUST: Die Verbindung ist nach einer bestimmten Zeit der Inaktivität auf der Verbindung auszulösen.

SHOULD: Pro User Agent und User sollen nur eine Session mit dem IdP erlaubt sein.

SHOULD: Die Anzahl Session pro User mit dem IdP soll begrenzt sein.

3.3.3.2 Löschen von nicht mehr genutzten SessionID (Cookies)

MUST: Session IDs müssen zufällig generiert werden.

Die Auswahl der möglichen Werte, die eine Session ID haben kann, hängt davon ab, wie gross die Wahrscheinlichkeit sein darf, dass eine ID in einem bestimmten Zeitraum wiederverwendet wird. Dies hängt u.a. von der durchschnittlichen Anzahl Cookies/SessionIDs ab, welche pro Sekunde gesetzt werden.

3.3.3.3 Abbrechen einer Verbindung (Logout)

MUST: Es muss die Möglichkeit bestehen, dass der User die Verbindung zum IdP terminieren kann.

SHOULD: Es soll die Möglichkeit bestehen, dass der User die Verbindung zur RP abbrechen kann.

3.4 Web Technologien

3.4.1 SOP/CORS

Same Origin Policy (SOP) ist ein Sicherheitsmerkmal im Browser. Gemäss diesem Sicherheitsmerkmal darf/soll ein Script auf einer dem User angezeigten Webseite keinen Zugriff auf den Inhalt einer Webseite anderen Ursprungs haben.

CORS steht für den englischen Ausdruck Cross Origin Resource Sharing. Dies ermöglicht, das Teilen von Informationen und Programmen in unterschiedlichen Domänen; dies für den dynamischen Aufbau, Gestaltung und Anwendung einer Webseite. Einfach ausgedrückt, widerspricht CORS der Same Origin Policy, was wiederum eine grosse Schwachstelle in sich birgt.

Das Domänen übergreifende Teilen stellt per se eine grosse Schwachstelle dar, dies erst recht, wenn dies über Domänen unterschiedlicher autonomer Körperschaften stattfindet.

MUST NOT: Der IdP darf kein Cross Origin Resource Sharing betreiben.

SHOULD NOT: Die RP soll kein Cross Origin Resource Sharing betreiben.

Diese Empfehlung hat Folgen für die Gestaltung der Webseiten.

3.4.2 JavaScript

MUST: JavaScript Programme einer Webseite dürfen nur von der gleichen Domäne wie die Webseite heruntergeladen werden. Dies gilt auch für andere Programme wie z.B. ActiveX.

JSON (JavaScript Object Notation) ist ein Datenformat, welches auf JavaScript Programmiersprache abgestimmt ist. Darin können auch Programme enthalten sein.

MUST NOT: Das Parsen von JSON darf nicht mit der Funktion «eval» vorgenommen werden. Mit «eval» können Programme direkt ohne vorgängige Prüfung auf der beim User geladenen Webseite ausgeführt werden.

SHOULD: JSON soll mit der Funktion «JSON.parse» verarbeitet werden.

SHOULD NOT: Das Datei-/Objektformat JSONP soll nicht verwendet werden, weil damit Daten Domänen übergreifend geladen werden können.

MUST NOT: Domänenübergreifende Anfragen mit XMLhttpRequest dürfen auf der Webseite eines IdP nicht abgesetzt werden können.

SHOULD NOT: Domänenübergreifende Anfragen mit XMLhttpRequest sollen auf der Webseite einer RPs nicht abgesetzt werden können.

3.4.3 iframes

MUST NOT: Der Austausch von Daten/Programmen zwischen 2 Frames unterschiedlichen Ursprungs darf beim IdP nicht vorgenommen werden.

SHOULD NOT: Der Austausch von Daten/Programmen zwischen 2 Frames unterschiedlichen Ursprungs soll bei der RP nicht vorgenommen werden.

Ansonsten werden Schwachstellen geschaffen, welche mit einem Angriff ausgenutzt werden können, z.B. s. Kapitel 5.4 «UI Redressing (Click Jacking)» und Kapitel 5.5 «Callback Function».

3.5 HTTP Header

Wenn überhaupt, unterstützen Browser erst ab einer bestimmten Version alle folgenden Sicherheitsmerkmale (korrekt). Deswegen empfiehlt es sich, zuerst zu prüfen, welcher Browser ab welcher Version die entsprechenden benötigten Sicherheitsmerkmale korrekt handhabt. Mit der entsprechenden Konsequenz, dass gewisse Browserversionen beim Zugriff auf die Webseite abgewiesen, resp. gesperrt werden (müssen). Anhand der Security Policy ist zu entscheiden, ob die jeweiligen Sicherheitsmerkmale benötigt werden, und dann daraus die Anforderungen an den Browser abzuleiten.

Quelle: https://caniuse.com/

3.5.1 HTTPS Weiterleitung

MUST: Um einer unsicheren Verbindung vom User-Agent zum IdP und zur RP vorzubeugen, muss der IdP und die RP ein http zu https redirect implementierten. Dieser darf nur Aufrufe innerhalb derselben Domain weiterleiten.

Beispiel: http://ech.ch -> https://ech.ch nicht aber http://www.ech.ch -> https://ech.ch

Grund: Reduktion der Gefahr gegenüber folgenden Angriffen: MitM, http Bookmark, Web Application configuration error

MUST: Ist eine Weiterleitung der Domain zusätzlich nötig, so muss diese Weiterleitung mit dem Ursprungsprotokoll erfolgen.

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

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

Quellen: OWASP Cheat Sheet [3]

3.5.2 Strict Transport Security

MUST: Damit möglichst eine TLS geschützte Verbindung zwischen dem User-Agent und dem IdP nach dem ersten Verbindungsaufbau genutzt wird, muss vom IdP ein HSTS Header mit langer Lebenszeit gesetzt werden.

SHOULD: Damit möglichst eine TLS geschützte Verbindung zwischen dem User-Agent und der RP nach dem ersten Verbindungsaufbau genutzt wird, soll vom RP ein HSTS Header mit langer Lebenszeit gesetzt werden.

Anmerkung: Vor- und Nachteile dieser Empfehlung sind in OWASP Cheat Sheet [3] erläutert. Damit der Empfehlung auch Folge geleistet werden kann und keine unerwünschten Nebenfolgen auftreten, kann die Domain Adresse des Server entsprechend granular bestimmt werden.

Beispiel: Den HTTS Header entsprechend auf pattern.example.com anwenden und nicht umfassend auf example.com

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

Quellen: RFC 6797, OWASP Cheat Sheet [3]

Reduktion der Gefahr gegenüber folgenden Angriffen: MitM, http Bookmark, Web Application configuration error

3.5.3 Strict Transport Security Preload List

MUST: Damit bereits vor dem ersten Besuch einer Website eine TLS Verbindung zum Einsatz kommt, muss eine Domain in die HSTS Preload List Submission beim IdP geladen werden. Dies muss vom IdP registriert werden.

Diese Empfehlung ist unter Umständen fürs Intranet nicht umsetzbar. Dann werden die RP und der IdP jedoch von der gleichen autonomen Körperschaft verwaltet. Die Empfehlungen hier sind jedoch primär darauf ausgerichtet, dass die RP und der IdP von unterschiedlichen Körperschaften verwaltet werden, siehe Kasten im Kapitel 1.2.

SHOULD: Dasselbe soll für die RP gelten.

Grund: Reduktion der Gefahr gegenüber folgenden Angriffen: MitM, http Bookmark, Web Application configuration error

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

3.5.4 X-Frame-Options

Mit X-Frame Options kann das Laden von iframes in einem Frame gesteuert werden.

Für X-Frame-Options gibt es folgende 3 verschiedene Argumente. DENY, SAMEORIGIN, ALLOW-FROM ORIGIN

MUST: IdP müssen das Argument «DENY» verwenden, welche verbietet, die Webseite des IdP in irgendeiner anderer Webseite zu laden.

SHOULD: RP sollen das Argument «DENY» oder «SAMEORIGIN» verwenden. Letzteres erlaubt es, nur Webseiten vom gleichen Ursprung zu laden.

Die RP handelt sich ein beträchtliches Risiko damit ein, wenn sie der Empfehlung nicht Folge leistet.

Der User achtet darauf, dass er das Passwort für die Authentisierung beim IdP in einem zur RP separaten Frame eingibt.

Anmerkung: Die CSP (Content Security Policy) Richtlinie “frame-ancestors” überschreibt den X-Frame-Options Header. Wenn eine Ressource beide Richtlinien aufstellt, wird die CSP frame-ancestors-Richtlinie durchgesetzt und die X-Frame-Options-Richtlinie ignoriert.

3.5.5 Content Security Policy

MUST: Zum Schutz der Webanwendung vor ungewollter Veränderung muss der IdP eine CSP (Content Security Policy) bereitstellen.

SHOULD: Die RP soll dies auch bereitstellen.

Grund: Reduktion der Gefahr gegenüber folgenden Angriffen: XSS, Framing, Clickjacking (UI Redressing)

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

Anmerkung: Erst einige Browser neuerer Versionen unterstützen die Sicherheitsmerkmale, siehe https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP.

MUST: Es muss zuerst geprüft werden, welche Browser ab welcher Version die folgenden Sicherheitsmerkmale unterstützten.

3.5.5.1 Empfehlungen der CSP für den IdP

Die folgende Tabelle mit den Merkmalen zu der CSP ist von der OWASP.

base-uri	Define the base uri for relative uri. **MUST:** Der IdP muss dies definieren und anwenden.
default-src	Define loading policy for all resources type in case of a resource type dedicated directive is not defined (fallback). **MUST:** Der IdP muss dies definieren und anwenden. Die Fallback Ressource muss der IdP sein.
script-src	Define which scripts the protected resource can execute. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
object-src	Define from where the protected resource can load plugins. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
style-src	Define which styles (CSS) the user applies to the protected resource. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
img-src	Define from where the protected resource can load images. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.

media-src	Define from where the protected resource can load video and audio. **SHOULD NOT:** Die Webseite des IdP soll grundsätzliche keine Quellenverweise auf Audio und Video haben. Ausnahmen bilden z.B. Accessability wie Zugang für Sehbehinderte. Ob nun Quellenverweise auf Video und Audio enthalten sind oder nicht. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
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:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
frame-ancestors	Define from where the protected resource can be embedded in frames. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
font-src	Define from where the protected resource can load fonts. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
connect-src	Define which URIs the protected resource can load using script interfaces. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
manifest-src	Define from where the protected resource can load manifest. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
form-action	Define which URIs can be used as the action of HTML form elements. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur der IdP sein.
sandbox	Specifies an HTML sandbox policy that the user agent applies to the protected resource. **MUST:** Auch wenn iframes nicht erlaubt sind, muss für alle Fälle dieses Merkmal so gesetzt werden, dass es so restriktiv wie möglich wirkt.
script-nonce	Define script execution by requiring the presence of the specified nonce on script elements. **MUST:** Dieses Merkmal muss verwendet werden.

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:** Die Anzahl Plugins muss beschränkt werden.
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:** Dieser Merkmal «Aktivieren» muss gesetzt werden
block-all-mixed-content	Prevent user agent from loading mixed content. **MUST:** Dieses Merkmal ist zu verwenden. Die Webseite darf keine gemischten Inhalte (TLS geschützt, ungeschützt) enthalten.
upgrade-insecure-requests	Instructs user agent to download insecure resources using HTTPS. **MUST NOT:** Dieses Merkmal darf nicht genutzt werden.
referrer	Define information user agent must send in Referer header. **MUST NOT:** Dieses Merkmal darf nicht verwendet werden, da es von den Browser nicht unterstützt wird, siehe https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP Alternative ist die Referrer Policy, siehe Kapitel 3.5.7
report-uri	Specifies a URI to which the user agent sends reports about policy violation. Siehe Kapitel 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. Siehe Kapitel 3.5.5.3 «Content Security Policy Reporting»

Tabelle 1: Empfehlungen für den IdP zu den Parametern bei der Content Security Policy

3.5.5.2 Empfehlungen der Content Security Policy für die RP

base-uri	Define the base uri for relative uri. **MUST:** Dies muss definiert und angewandt werden.
default-src	Define loading policy for all resources type in case of a resource type dedicated directive is not defined (fallback). **MUST:** Die RP muss dies definieren und anwenden. Die Fallback Ressource muss die RP sein.
script-src	Define which scripts the protected resource can execute. **SHOULD:** Dieses Merkmal soll verwendet werden zu verwenden, und die Ressource soll nur die RP sein.

object-src	Define from where the protected resource can load plugins. **SHOULD:** Dieses Merkmal soll verwendet werden zu verwenden, und die Ressource soll nur die RP sein.
style-src	Define which styles (CSS) the user applies to the protected resource. **SHOULD:** Dieses Merkmal soll verwendet werden zu verwenden, und die Ressource soll nur die der RP sein.
img-src	Define from where the protected resource can load images. **SHOULD:** Dieses Merkmal soll verwendet werden zu verwenden, und die Ressource soll nur die der RP sein.
media-src	Define from where the protected resource can load video and audio. **SHOULD:** Dieses Merkmal soll verwendet werden zu verwenden, und die Ressource soll nur die der RP sein.
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:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur die RP sein.
frame-ancestors	Define from where the protected resource can be embedded in frames. **MUST:** Dieses Merkmal ist zu verwenden, und die Ressource darf nur die RP sein.
font-src	Define from where the protected resource can load fonts. **SHOULD:** Dieses Merkmal soll verwendet werden zu verwenden, und die Ressource soll nur die RP sein.
connect-src	Define which URIs the protected resource can load using script interfaces. **SHOULD:** Dieses Merkmal soll verwendet werden zu verwenden, und die Ressource soll nur die RP sein.
manifest-src	Define from where the protected resource can load manifest. **SHOULD:** Dieses Merkmal soll verwendet werden, und die Ressource soll nur die RP sein.
form-action	Define which URIs can be used as the action of HTML form elements. **SHOULD:** Dieses Merkmal soll verwendet werden zu verwenden, und die Ressource soll nur der RP sein.

sandbox	Specifies an HTML sandbox policy that the user agent applies to the protected resource. **MUST:** Dieses Merkmal muss bei iframes aus einer externen Quelle verwendet werden. **SHOULD:** Dabei soll das Merkmal so gesetzt werden, dass es so restriktiv wie möglich wirkt. **SHOULD:** Dieses Merkmal soll bei iframes aus einer internen Quelle verwendet werden. **MUST:** Falls das iframe aus einer internen Quelle stammt, muss das Merkmal so verwendet werden, dass es mit den anderen Merkmalen der Content Security Policy konsistent ist.
script-nonce	Define script execution by requiring the presence of the specified nonce on script elements. **SHOULD:** Dieses Merkmal soll genutzt werden.
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:** Die Anzahl Plugins muss beschränkt werden.
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:** Dieser Merkmal «Aktivieren» muss gesetzt werden
block-all-mixed-content	Prevent user agent from loading mixed content. **MUST:** Dieses Merkmal ist zu verwenden. **MUST NOT:** Die Webseite darf keine gemischte Inhalte (TLS geschützt, ungeschützt) enthalten.
upgrade-insecure-requests	Instructs user agent to download insecure resources using HTTPS. **MUST NOT:** Dieses Merkmal darf nicht genutzt werden.
referrer	Define information User Agent must send in Referrer header. **MUST NOT:** Dieses Merkmal darf nicht verwendet werden, da es von den Browsern nicht unterstützt wird, siehe https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP Alternative ist die Referrer Policy, siehe Kapitel 3.5.7
report-uri	Specifies a URI to which the user agent sends reports about policy violation. Siehe Kapitel 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. Siehe Kapitel 3.5.5.3 «Content Security Policy Reporting»

Tabelle 2: Empfehlungen für die RP zu den Parametern bei der Content Security Policy

3.5.5.3 Content Security Policy Reporting

MUST: Das Reporting muss sich auf Ereignisse bei den Verbindungen zwischen Browser und IdP und zwischen Browser und RP beschränken.

SHOULD: Falls keine rechtliche Grundlage für ein Reporting besteht, dann soll im Sinne der Verhältnismässigkeit gelten:

> Falls der Schutzbedarf bei der RP oder beim IdP grösser ist als an Privacy beim User, dann hat das Bedürfnis an Privacy beim User für die Sicherheit beim IdP oder der RP zurückzustehen. Folglich soll ein Reporting vorgenommen werden. Vorausgesetzt der User gibt zuvor seine Zustimmung.

MAY: Falls der User nicht zustimmt, kann der Verbindungsaufbau abgebrochen werden.

Quellen zum Umsetzen des Reporting: Content Security Policy Cheat Sheet, OWASP

3.5.6 Speicherung von Tokens wie JWT, SAML

MUST: Falls die erhaltenen Tokens bei IdP/RP zwecks (Nachvollziehbarkeit der Prozesse) gespeichert werden müssen, dann muss die RP/IdP die erhaltenen Tokens in Bezug auf Integrität und Vertraulichkeit (vor ungewollter Einsichtnahme durch Dritte) nachhaltig schützen.

Anmerkung: Das Speichern der Token, z.B. der SAML Meldungen, kann u.a. für die Nachvollziehbarkeit der Abläufe sinnvoll/notwendig sein. Die Nachvollziehbarkeit bietet die Möglichkeit, im Falle eines Fehlers verlässlich die Ursache des Fehlers und somit gegebenenfalls die Verantwortlichkeit zu bestimmen. In Erinnerung rufend: «IdP und RP können von unterschiedlichen Körperschaften verwaltet werden.»

3.5.7 Referrer Policy

MUST: Für einen wirksamen Informationsschutz zwischen IdP und RP müssen beide eine Referrer Policy setzen, welche verbietet, dass sensitive Daten an eine andere Partei «durchsickern».

Dabei sind folgende Massnahmen zur Reduktion der Gefahr eines Durchsickerns (Data Leakage) von Daten einzuhalten: strict-origin-when-cross-origin, no-referrer, strict-origin.

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

3.5.8 X-Content-Type-Options

Der HTTP Header x-content-type-options kann verhindern, dass der User-Agent MIME Types selbständig interpretiert. Dies verhindert, dass ungewollt Code als anderer Datentyp geladen und ausgeführt wird.

Die Massnahme reduziert die Gefahr gegenüber folgenden Angriffen: MIME sniffing, Remote Code Execution, OWASP Secure Headers Project

MUST: IdP und RP müssen http header x-content-type-options mit der Option «nosniff» verwenden.

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

3.5.9 X-XSS-Protection

Der X-XSS-Protection-Header wird von modernen Browsern nicht mehr verwendet, und seine Verwendung kann zu weiteren Sicherheitsproblemen auf der Client-Seite führen. Es wird daher empfohlen, den Header auf X-XSS-Protection auf „0“ zu setzen. Dies, um den XSS-Auditor zu deaktivieren und nicht zuzulassen, dass er das Standardverhalten des Browsers beeinflusst.

MUST: Dieser Header muss auf X-XSS-Protection = «0» gesetzt werden.

3.5.10 Cache-Control

Browser speichern oft Daten, welche sensitive Informationen für andere hinterlassen können. Mit dem HTTP Header «Cache-Control» soll das Speichern von Informationen im Cache des Browser gesteuert werden.

SHOULD: Der Cache-Control Header für private/vertrauliche Seiten soll auf no-store gesetzt werden. Davon ausgenommen sind statische Ressourcen.

Falls es nicht möglich ist, der Empfehlung zu entsprechen, dann no-cache, max-age=1000 setzen.

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

3.5.11 Feature Policy

Das Merkmal «Feature-Policy», früher «Permissions-Policy» genannt, dient dazu, festzulegen, was im Frame des Browser genutzt werden kann.

> «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.»

Damit wird die Gefahr reduziert, dass Daten (unbeabsichtigt an Dritte) durchsickern.

SHOULD: Zuerst soll geprüft werden, welche Browser dieses Merkmal unterstützen. Aufgrund dessen soll dann entschieden werden, ob es verwendet werden soll. Zur Unterstützung dieses Feature in den Browsern siehe https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Feature-Policy.

Die folgenden Feature sind bei der oben genannten Adresse erklärt.

> 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=()

Falls der Browser diese Feature unterstützt, dann:

> **MUST:** Für den IdP muss gelten: Alle Features sollen gesetzt und nicht erlaubt sein, mit Ausnahme von microphone (), camera=(),display-capture=(), publickey-credentials-get=() wegen der Barrierefreiheit und der Nutzung von QR Codes, sowie WebAuthN.
>
> **MUST:** Die RP muss nicht benötigte Funktionen mittels der entsprechenden Parameterwahl unterbinden.

MAY: <allowlist> kann zurzeit explizit leer stehen.

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

3.5.12 Weitere

Folgende Header können in Zukunft relevant sein, weil sie von der OWASP empfohlen werden. 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)

Deswegen sollten die Aktivitäten rund um die Empfehlungen der Headers beobachtet werden. Danach ist festzulegen, ob sie berücksichtigt werden sollen.

MAY: Diese Sicherheitsmerkmale können eingebaut werden, im Bewusstsein, dass nicht alle Browser (Versionen) diese zu unterstützen vermögen.

MUST: Falls die Sicherheitsmerkmale verwendet werden, dann muss gelten:

> **MUST NOT:** Der IdP soll kein Cross Resource Sharing betreiben.
>
> **MUST:** Folglich müssen die entsprechenden Merkmale so gesetzt werden, dass diese Vorgabe erfüllt wird.
>
> **SHOULD:** Die RP soll kein Cross Resource Sharing betreiben. Falls doch, dann
>
> **MUST:** Dann müssen die Parameter bei der RP so gesetzt werden, dass einzig diese Ressource verwendet werden darf.

Quellen: OWASP Secure Headers Project – OWASP

3.6 Domain Name System Isolation

MUST: Der IdP muss einen Domain Namen FQDN einzig für die Interaktion (Login, Reset, Register) mit dem Benutzer reservieren

MUST NOT: Weitere Dienste dürfen nicht denselben Domain Namen nutzen.

Dies reduziert die Gefahr gegen folgende Angriffe: Cookie extraction, XSS, CSRF

Quellen: NIST Special Publication 800-81-2

3.7 Domain Name System Security Extensions

Um die Authentizität der Antworten auf DNS-Anfragen zu schützen, wird DNSSec verwendet.

MUST: Die DNS Antworten zu Domains des IdP müssen mittels DNSSec geschützt werden.

SHOULD: Die DNS Antworten zu Domains des RP sollen mittels DNSSec geschützt werden.

DANE verwendet einen anderen Ausgangspunkt für die Verlässlichkeit. Der hier empfohlene Ausgangspunkt sind die Zertifikate eines nach ZertES anerkannten Certificate Service Provider.

Quellen: NIST Special Publication 800-81-2

4 Registrieren und Authentisieren

In diesem Kapitel werden Empfehlungen zur Sorgfalt bei der Registrierung und der Authentisierung abgegeben, damit Schwachstellen im Bereich der Authentisierung minimiert werden.

4.1 Worum geht es?

Sicherheitstechnologie und die damit geschützte Anwendung sind oft nicht aufeinander abgestimmt und verwenden auch des Öfteren unterschiedliche IDs, welche den Zugang zu Ressourcen ermöglichen. Dies birgt Risiken eines Angriffs.

Beispiel: Bei OpenID Connect haben die unter der Verwaltung/Verantwortlichkeit des IdP stehenden Komponenten sowie der RP unterschiedliche IDs für deren Objekte.

Beispiel: TLS- und HTTP-Verbindungen sind per se nicht miteinander abgestimmt. Bei der HTTP-Verbindung wird der Zugang mittels des Cookie zu den Ressourcen gewährt. Zwischen Cookie und TLS Session ID besteht im Allgemeinen kein virtueller Link.

Bestünde ein solcher Link, dann würde jedes Mal geprüft, ob das Cookie mit der TLS Session ID übereinstimmt, worüber der User authentisiert worden ist. Das „In Verbindung setzen“ der TLS Session ID mit dem Cookie schützt vor Missbrauch beim Stehlen/Kopieren eines Cookie durch einen unberechtigten Dritten.

Anmerkung: Die Verlinkung zwischen TLS und (Web-)Anwendung wird auch als Channel Binding bezeichnet.

Umgekehrt: Wird nicht ein entsprechender Link zwischen Cookie und TLS-Session ID hergestellt und jeweils geprüft, ob dies übereinstimmt, dann entsteht dadurch eine Schwachstelle.

Beispiel: Der XML-Signature Wrapping Angriff nützt aus, dass Anwendung und Sicherheitstechnologie (XML-Signatur) nicht aufeinander abgestimmt sind, siehe dazu [12]: Es wird zwar eine Signatur angefertigt, aber die sensitive Info an die Anwendung wird nach dem Angriff nicht mehr von der Signatur geschützt und kann somit ersetzt werden. Folgendes könnte das Risiko eines solchen Angriffs reduzieren:

> Vereinbaren eines XML-Schemas. Darin wird festgelegt, wo sich die Anwenderinformation zu befinden und was die Signatur zu schützen hat. Nebst der Gültigkeit der Signatur ist zu prüfen, ob das erhaltene XML-Objekt schemakonform ist und die Signatur die erforderlichen Unterobjekte wie beabsichtigt schützt.

4.2 Grundsätzliches zur Registrierung/Authentisierung

Beim Registrieren geht es darum, identitätsbezogene Attribute einer natürlichen oder juristischen Person zu sammeln und zu verifizieren. Anhand dieser Attribute soll bei der Authentisierung die empfangene Information der entsprechenden Person (möglichst) eindeutig zugeordnet werden können.

Wie verlässlich diese Zuordnung vorgenommen und die Verantwortlichkeit bestimmt werden kann, hängt von der Verlässlichkeit und der Art der Attribute, sowie von der Qualität der Authentisierung ab, siehe hierzu eCH-0170 und eCH-0171. Dabei sind die Mindestanforderungen an die Qualität der Authentisierung zu definieren.

MUST: Alle IDs sind aufzunehmen, welche für die Authentisierung und Autorisierung notwendig sind. Bei Servern müssen diese in Relation mit den Angaben im X.509 Zertifikat gesetzt werden, d.h. z.B. tabellarisch oder über eine Signatur miteinander verbunden sein, um die Authentizität und Integrität der Relation zu schützen.

Die pro Authentisierungsprozess individuellen dynamischen Werte (Parameter) müssen auch in Relation zu den registrierten IDs gesetzt und während der Dauer der Session geprüft werden.

Beispiel: Aus Sicht der RP für den User Access bei OpenID Connect besteht die Relation aus:

> Authentisierungsbestätigung des IdP, TLS Session mit dem User und Cookie der damit geschützten http-Verbindung. Stimmt der Cookie Wert mit der TLS Session ID nicht mehr überein, dann ist die Verbindung abzubrechen oder die Authentifizierung zu (re)validieren.

MUST: Das Einhalten dieser Relation (siehe Beispiel oben) muss bei der Authentisierung und während der Session-Dauer geprüft werden. Wird festgestellt, dass die Relation nicht wie definiert/vereinbart übereinstimmt, dann muss der Verbindungsaufbau oder die Session abgebrochen und ein entsprechender Fehler gemeldet/geloggt werden.

Beispiel: Für die Authentisierung des IdP aus Sicht der RP bei OpenID Connect, s. Kapitel 5.4.3. in diesem Dokument.

Wie erwähnt, werden die genannten Regeln nicht beachtet, werden (zusätzliche) Schwachstellen/Gefahren bei der Authentisierung und der Autorisierung geschaffen.

In den folgenden Unterkapiteln werden Empfehlungen zur Registrierung, Authentisierung und den zu treffenden Vereinbarungen zwischen den jeweiligen Parteien abgegeben.

4.3 Zwischen IdP und User/User Agent

4.3.1 Registrierung

4.3.1.1 User beim IdP

SHOULD: Der IdP soll die Benutzer identifizieren, deren Attribute erfassen und verifizieren. Attribute, welche für die Authentisierung benötigt werden, sind mit einem Authentisierungsmittel zu versehen.

Ob und wie gut eine Person zu identifizieren ist, hängt u.a. von der geforderten Sicherheit(stufe) ab. In der Durchführungsverordnung (EU) 2015/1502 der Kommission vom 8. September 2015 zu eIDAS sind Anforderungen an die Authentisierung und an die Identifizierung pro Sicherheitsstufe definiert.

Anmerkung: Ist eine hohe Sicherheit bei der Authentisierung erforderlich, so ist ein persönliches Erscheinen zwecks Registrierung (Erfassung der Attribute) notwendig. Begründung: Die Mittel fürs Authentisieren sind übertragbar, die biometrische Beschaffenheit einer natürlichen Person jedoch nicht.

SHOULD: Der IdP soll ausreichend Attribute vom User sammeln, verifizieren und später der Authentisierungsbestätigung oder dem ID-Token beifügen (Userinfo wäre zu bevorzugen mit OpenID Connect), so dass die RP die Autorisierung aufgrund der vom IdP gelieferten Attributen über den User vornehmen kann. D.h. den Anmeldevorgang und die Autorisierung eindeutig einem User zuordnen kann. Falls dem nicht so ist, ist eine separate Registrierung des Users bei der RP vorzunehmen oder die Informationen von einer anderen Datenbank zu beziehen, welches für die Richtigkeit der entsprechenden Attribute verantwortlich ist.

Z.B., wenn die RP eine Berufsbescheinigung einfordert und diese beim IdP nicht registriert wird oder nicht werden darf. Z.B. stellt die Approbation eines Arztes oder die Zulassung als Notar oder Rechtsanwalt eine Berufsbescheinigung dar. Diese Informationen werden in den entsprechenden Registern verwaltet, d.h. u.a. aktualisiert. Will die RP nun wissen, ob die Berufsbescheinigung noch gültig ist, dann muss sie eine entsprechende Anfrage ans Register stellen, weil die Infos nicht beim IdP vorliegen, nicht vorliegen dürfen oder nicht ausreichend zeitnah aktualisiert werden.

4.3.1.2 IdP beim User

MUST: Der Server muss auf Basis eines X.509 Zertifikats mit TLS authentisiert werden.

MUST: Der User muss die URL des IdP und das Root-Zertifikat beim User Agent implementiert haben, welches für die Verifikation des TLS-Zertifikats des IdP verwendet wird.

4.3.2 Zu prüfen beim Authentisieren

4.3.2.1 Durch den User

MUST: Bei einer Fehlermeldung betreffend TLS-Verbindungsaufbau durch den User-Agent muss der User das Browser Fenster schliessen und damit die Datenkommunikationsverbindung abbrechen.

MUST: Die natürliche Person muss nach dem Verbindungsaufbau mit dem IdP prüfen, ob die URL beim User-Agent Frame mit der ihm bekannten URL des IdP übereinstimmt und die Verbindung mit TLS geschützt ist.

4.3.2.2 Durch den IdP

MUST: Der IdP muss den User authentisieren. Es ist zu prüfen, ob das dabei verwendete Authentisierungsmittel mit dem bei der Registrierung vereinbarten übereinstimmt.

SHOULD: Der IdP soll mit heuristischen Methoden prüfen, ob die Verbindung zwischen dem User-Agent und dem IdP intakt und valid ist

4.3.3 Regeln der Verantwortlichkeit (Rechte und Pflichten)

MUST: Die natürliche Person ist u.a. in Kenntnis zu setzen, wie sie mit dem Authentisierungsmittel umzugehen hat, d.h. welche Sorgfaltspflichten sie dabei zu erfüllen hat.

SHOULD: Die zu erfüllenden Sorgfaltspflichten sollen möglichst präzise formuliert und abhängig von der anzustrebenden Sicherheit sein. Formulierung wie «angemessen» oder «den Umständen entsprechend sind» sind zu vermeiden oder zu konkretisieren, weil der User sich im Allgemeinen darunter nichts Konkretes vorstellen kann.

MUST: Die Pflichten des IdP im Umgang mit den registrierten Daten und bei der Prüfung der Authentisierung des Users müssen definiert und dem User zugänglich gemacht werden. D.h. er muss in Kenntnis gesetzt werden, wo er diese Informationen beziehen kann.

4.4 Zwischen RP und User Agent/User

4.4.1 Registrierung

User und RP haben sich grundsätzlich gegenseitig nicht zu registrieren. Ausnahme bildet u.a., wenn die RP nicht ausreichend Informationen vom IdP erhält, damit sie die Autorisierung des Users vornehmen kann.

4.4.2 Zu prüfen beim Authentisieren

4.4.2.1 Durch den User

MUST: Der Server wird auf Basis eines X.509 Zertifikats mit TLS authentisiert.

MUST: Der User muss nach dem Verbindungsaufbau mit der RP prüfen, ob die URL beim User-Agent Frame mit der ihm bekannten URL der RP übereinstimmt und die Verbindung mit TLS geschützt ist.

MUST: Damit die RP auf die Ressource des Users zugreifen kann oder dessen Personendaten erhält, muss der User diesem zuvor zustimmen. Beim Einholen der Zustimmung muss der User klar und eindeutig erkennen können, wem er die Zustimmung erteilt. Dafür muss der IdP besorgt sein.

4.4.2.2 User bei der RP

Die Authentisierung erfolgt über die Authentisierungsbestätigung des IdP. Bei OpenID Connect erfolgt dies über ein ID-Token oder Refresh-Token. Bei SAML über eine Assertion.

SHOULD: Die Bestätigung, dass der User erfolgreich authentisiert wurde, soll vom IdP signiert werden. Bei OpenID Connect z.B. mit einer Signatur des JSON Token.

U.a. für die Nachvollziehbarkeit der Prozesse aus Sicht der RP ist bei OpenID Connect problematisch, dass Attribute des Users an die RP gesendet werden können, welche vom IdP oder von der Userinfo-Stelle nicht signiert worden sind.

MUST: Falls die RP und der IdP nicht von der gleichen autonomen Körperschaft verwaltet werden, müssen die für die Authentisierung und Autorisierung erforderlichen Personenattribute signiert werden, welche der IdP an die RP sendet. Signatur und Inhalt der Bestätigung müssen von der RP geprüft werden.

Wichtig: Aus Sicht der RP ist irrelevant, wie sich der User beim IdP authentisiert hat. Wichtig für ihn ist, wie verlässlich die „Authentisierungsbestätigung" des IdP mit dem darin angegebenen Authentisierungsverfahren ist und welche rechtlichen Konsequenzen sich ergeben, wenn die Bestätigung nicht verlässlich ist, d.h. Fehler aufweist.

4.4.3 Regeln der Verantwortlichkeit (Rechte und Pflichten)

SHOULD: Die Verantwortlichkeiten und Rechte sollen sich aus der Art des Dienstes, welche die RP anbietet, ableiten.

4.5 Zwischen RP und IdP

4.5.1 Registrierung

4.5.1.1 RP beim IdP

SHOULD NOT: Eine dynamische Registrierung soll nicht durchgeführt werden. Diese Form der Registrierung kann Schwachstellen enthalten, welche von einem Angreifer ausgenutzt werden kann. Solche Schwachstellen sind z.B. in [2] für die dynamische Registrierung bei OpenID Connect beschrieben.

MUST NOT: Bei hohen Sicherheitsanforderungen, z.B. bei besonders schützenswerten Personendaten, darf die Registrierung nicht dynamisch vorgenommen werden.

MUST: Der IdP muss alle IDs der RP aufnehmen, welche für die Authentisierung und Autorisierung der RP notwendig sind. Diese müssen in Relation mit den Angaben im X.509 Zertifikat für die TLS-Authentisierung gesetzt werden, d.h. z.B. tabellarisch oder über eine Signatur miteinander verbunden sein.

Bei OpenID Connect sind es Client-ID, Redirect URI, X.509 TLS-Zertifikat.

MUST: Deswegen müssen die entsprechenden IDs im Zertifikat und das CA-Zertifikat zur Prüfung der X.509 Zertifikate bekannt sein.

Anmerkung: In «OpenID Connect Dynamic Client Registration» sind Attribute aufgeführt, welche beim IdP registriert werden können. Doch von einer dynamischen Registrierung soll/muss Abstand genommen werden. Das Dokument dient nur als Beispiel dafür, welche Attribute über die RP beim IdP registriert werden können.

4.5.1.2 IdP bei der RP

MUST: Die RP muss alle IDs des IdP aufnehmen, welche für die Authentisierung des IdP notwendig sind. Diese IDs müssen in Relation mit den Angaben im X.509 Zertifikat gesetzt werden, d.h. z.B. tabellarisch oder über eine Signatur/ miteinander verbunden sein. Oder im Zertifikat selber enthalten sein.

Bei OpenID Connect sind dies die im discover endpoint publizierten Endpunkte. Unter anderem: Issuer Identifier, URL Token (User) Endpoint, URL im TLS X.509 Zertifikat, X.509 Zertifikat für die Verifikation der signierten JSON Web Token.

Die X.509 Zertifikate sind nicht statisch. Deswegen müssen die entsprechenden IDs im Zertifikat und das CA-Zertifikat zur Prüfung der X.509 Zertifikate bekannt sein.

4.5.2 Regelung der Verantwortlichkeit (Rechte und Pflichten)

SHOULD: Der RP soll bekannt sein, in welchem Umfang der IdP für die Authentisierungsbestätigung einzustehen hat.

4.5.3 Anforderungen beim Authentisieren

Anforderung an die dynamischen Authentisierungsparameter bei OpenID Connect:

MUST: Folgende Parameter müssen im eGovernment Umfeld zufällig generiert und verwendet werden: State, Nonce und Code Challenge für das PKCE-Verfahren (RFC 7636).

MUST: Die für die Authentisierung und Autorisierung des Users benötigten Attribute müssen vom IdP oder vom Userinfo-Point signiert sein. Sind sie nicht signiert, dann hat die RP keinen Beleg/Nachweis dafür, ob die vom IdP erhaltenen Werte stimmen.

SHOULD: Bei der ersten Verbindung zwischen User-Agent und RP wird ein Cookie gesetzt. Die Redirect-URI sollte in den Gültigkeitsbereich dieses Cookies fallen, sodass zwischen User-Agent und Redirect-URI keine weitere HTTP-Verbindung aufgebaut wird.

SHOULD: Der User soll vom IdP jedes Mal in Kenntnis darüber gesetzt werden, mit welcher RP er gedenkt, eine Verbindung aufzubauen. Dabei sollen dem User die Web-Adresse der RP, plus Ort und ein kennzeichnender Name angezeigt werden. Dieser Name muss bei der Registrierung der RP beim IdP ebenfalls erfasst werden.

4.5.4 Zu Prüfen beim Authentisieren

4.5.4.1 Authentisieren der RP beim IdP

Anhand des Authorization Code Flow von OpenID Connect wird dargelegt, was der IdP bei der Authentisierung der RP zu prüfen hat.

MUST: Die OpenID Connect Standard geltenden Empfehlungen müssen eingehalten werden.

SHOULD: Beim Token Request soll geprüft werden, ob der dynamisch vergebene Authorization Code, Client ID mit der URL im X.509 Zertifikat der RP für die Authentisierung mittels TLS übereinstimmt. Dies bedingt ein mTLS Verfahren beim Verbindungsaufbau.

Anmerkung: Es gibt Alternativen zu mTLS, welche zur Authentisierung von RP genutzt werden können. Doch diese sind weniger sicher. Zudem muss die RP für den Verbindungsaufbau mit dem User ein Server Zertifikat haben.

SHOULD: Die RP soll sich mittels mTLS bei den vom IdP verwalteten Servern und beim Authorization Server authentisieren.

4.5.4.2 Authentisieren des IdP bei der RP

Anhand des Authorization Code Flow von OpenID Connect wird dargelegt, was die RP bei der Authentisierung des IdP zu prüfen hat und welche IDs der RP beim IdP registriert werden müssen.

Token Response: Die RP hat beim Erhalt des Tokens nebst der Signatur zu prüfen, ob die IDs übereinstimmen. Dies sind: Issuer Identifier, URL Token (User) Endpoint, URL im TLS X.509 Zertifikat, X.509 Zertifikat für die Verifikation der signierten JSON Web Token.

Zum Schutz der Authentizität werden pro Verbindungsaufbau/Datenaustausch die Parameter Nonce und State verwendet, deren Werte pro Verbindungsaufbau zufällig erzeugt werden müssen.

Zudem muss geprüft werden, ob der erhaltene State Wert, Nonce Wert mit den Werten im Authentication Request übereinstimmen.

Weiter muss die Prüfung nach dem PKCE-Verfahren gemäss RFC 7636 (PKCE) gültig sein.

4.5.5 Vereinbarung

SHOULD: IdP und RP sollen sich auf eine Referrer Policy einigen, siehe Kapitel 3.5.7 «Referrer Policy».

4.5.6 Informationen zur Autorisierung der RP durch den User

MUST: Für die Einverständniserklärung, dass die RP Personendaten über den User beim IdP erhalten darf, müssen dem User die Web-Adresse der RP, plus Ort und ein kennzeichnender Name angezeigt werden. Dieser Name muss bei der Registrierung ebenfalls erfasst werden.

4.6 Anmerkung zur Sicherheit beim Protokollablauf

Ein Verbindungsaufbau mit SAML hat gegenüber OpenID Connect den Vorteil, dass alle Anfragen (Request) und Antworten signiert werden können. Dies bietet einen erhöhten Schutz. Doch weit sicherer ist es grundsätzlich, wenn die Authentisierung des Users direkt bei der RP stattfindet, dieses auf Basis von X.509 Zertifikaten, siehe hierzu [11].

5 Angriffstechniken

Hier werden vor allem Angriffstechniken präsentiert, welche das Ziel verfolgen, im Namen des Users Befehle abzusetzen und Aktionen auszulösen. Der Empfänger des Befehls vermag nicht zu erkennen, dass der Befehl nicht vom User gesandt wurde.

In RFC 6819 sind weitere Bedrohungen und Gegenmassnahmen im Rahmen des OpenID Connect/OAUTH Protokolls aufgeführt.

Welche Angriffstechniken auch beim PKI-Ansatz eine Bedrohung darstellen, sind im Kapitel 6 kurz zusammengefasst.

5.1 Cross Side Scripting (CSS, XSS)

5.1.1 Beschreibung des Angriffs

Ein Angreifer veranlasst eine Webseite (Server der Webseite) dazu, dass sie Befehle beim User Agent absetzt und dort Aktionen ausführen lässt. Das Ziel dabei ist meist, dass ein Angreifer die Möglichkeit erhält, im Namen des Users zu agieren. Z.B. indem er das Cookie der Verbindung ausfindig machen kann und dann im Namen des Users Befehle an den Server senden kann.

Für Details siehe u.a. [5], [4], [6]. [14] OWASP: https://owasp.org/www-community/attacks/xss/.

5.1.2 Wer kann den Angriff auslösen

Jeder (Web)Server, mit welchem der User-Agent eine Verbindung aufbaut, kann den Angriff initiieren. Prädestiniertes Opfer eines solchen Angriffs sind RP und IdP; bereits aufgrund des Verbindungsaufbaus.

Der Angriff kann auch gestartet werden, indem eine Datei auf einem verletzbaren Server gespeichert wird. Diese Datei löst dann beim User Agent vom User ungewollte Befehle auch, wenn er sie herunterlädt.

Kennzeichen dieses Angriffs ist, dass der User-Agent den Antworten des Servers zu sehr vertraut.

5.1.3 Bedrohung

Vom User ungewollte (Trans)Aktionen werden in seinem Namen ausgelöst.

5.1.4 Gegenmassnahmen

Der Server nimmt

- Eine Eingangsprüfung der Daten(-formate), welche er akzeptiert
- Ausgangsprüfung der Daten(-formate), welche Daten an den User gesandt werden.

Für Details siehe u.a. [5], [4], [6]. Siehe auch die entsprechenden Sicherheitsmerkmale im Kapitel 3.5 «HTTP Header».

5.2 Cross Site Request Forgery (CSRF, XSRF)

5.2.1 Beschreibung des Angriffs

Beim Laden einer Webseite (des Angreifers) werden Befehle an eine andere Webseite gesandt und Aktionen dort ausgelöst. [5], [4], [6]. Kennzeichen dieses Angriffs ist, dass der Server (angegriffene Webseite) dem Request des Users zu sehr vertraut.

5.2.2 Wer kann den Angriff auslösen

Jeder (Web)Server, mit welchem der User-Agent eine Verbindung aufbaut. Prädestiniert, Opfer eines solchen Angriffs zu werden, sind die RP und der IdP, schon alleine aufgrund des Verbindungsaufbaus.

5.2.3 Bedrohung

Bedrohung ist, dass vom User ungewollte (Trans)Aktionen in seinem Namen ausgelöst werden.

5.2.4 Gegenmassnahmen

Verwendung „versteckter Token“ (CSRF Token) im http-Header, sodass erkannt werden kann, ob der Befehl, resp. die Anfrage vom User-Agent stammt oder von einem Dritten. Weitere Massnahmen: Double Submit Cookie, SameSite Cookie

5.3 Server Site Request Forgery

5.3.1 Beschreibung des Angriffs

Bei Server-Site-Request-Forgery (SSRF) wird ein Server dazu gebracht, Anfragen vom Angreifer auszulösen. Diese Anfragen können auf den Server oder auf andere Server weitergeleitet werden. Wenn die Anfragen an ein Drittsystem versandt werden, kann ein Angreifer damit den Server als den Urheber erscheinen lassen. Details siehe [10].

5.3.2 Wer kann den Angriff auslösen

Jeder, der eine Verbindung mit dem anzugreifenden Server aufbauen kann.

5.3.3 Bedrohung

Z.B. das unerwünschte Ausführen von Befehlen und die unerwünschte Offenlegung sensitiver Daten.

5.3.4 Gegenmassnahmen

Eine White Liste, welche Befehle eine Applikation absetzen darf. Details siehe [10].

5.4 UI Redressing (Click Jacking)

5.4.1 Beschreibung des Angriffs

Eine dritte Partei (Angreifer) veranlasst den User, Befehle auf einer anderen Webseite abzusetzen oder Aktionen dort auszulösen. Z.B. Ein User meint, er gebe die Befehle oder löse die Aktionen auf der Webseite der Drittpartei aus, doch in Wirklichkeit löst er sie auf der Webseite einer anderen Domäne aus. Z.B. nimmt er an, er klicke einen Button auf der Webseite der dritten Partei, doch in Wirklichkeit bestätigt er eine Aktion auf einer anderen Webseite.

Die Webseite nimmt vermeintlich an, die Aktion sei durch den User bewusst ausgeführt worden. Für Details siehe [5], [4].

5.4.2 Wer kann den Angriff auslösen

Jeder (Web)Server, mit welchem der User-Agent eine Verbindung aufbaut. Prädestiniert, Opfer eines solchen Angriffs zu werden, sind die RP und der IdP; dies schon alleine aufgrund des Verbindungsaufbaus.

5.4.3 Bedrohung

Eingabe von ungewollten Befehlen auf einer Webseite durch den User.

5.4.4 Gegenmassnahmen

Einfügen «X-Frame-Options» im http-Response Header oder «frame-ancestors» in der Content Security Policy des http-Headers. Browser unterstützen/befolgen diese Anweisung im http-Response Header erst ab einer bestimmten Version.

MUST: Es dürfen nur Browser-Versionen unterstützt werden, welche die Befehle in den «X-Frame-Options» oder in «frame-ancestors» in der Content Policy verstehen und entsprechend befolgen.

Weitere Quellen: X-Frame-Options – HTTP, OWASP Secure Headers Project – OWASP

5.5 Callback Function

Der Begriff «Callback Function» ist mehrdeutig. Hier wird Folgendes verstanden: Eine oder mehrere Funktionen in einem SW-Programm, kurz Funktion, werden als Parameter für eine andere Funktion übergeben.

Ziel des Angriffs: Die Same Origin Policy (SOP) wird umgangen. Die Umgebung, in welcher eine Funktion ausgeführt wird/ werden darf, wird in die gewünschte Zielumgebung transferiert. Die Ursprungsumgebung wird als Callback URL bezeichnet.

5.5.1 Wer kann den Angriff auslösen

Jeder (Web)Server, mit welchem der User-Agent eine Verbindung aufbaut.

5.5.2 Bedrohung

Ungewolltes Ausführen von Code im User Agent.

5.5.3 Gegenmassnahmen

Kein Cross Resource Sharing (CORS) auf der Webseite, statische Callback, White List der möglichen Callback. Für Details siehe [9].

5.6 Code Injection in Token

5.6.1 Beschreibung des Angriffs

In SAML, besser XML, oder in einem JSON Web Token kann ausführbarer Code wie ActiveX, JavaScript oder Java Applets enthalten sein oder ein Link darauf. Dies kann dazu führen, dass die Programme beim Empfänger des Tokens (RP) ohne seinen Willen ausgeführt werden.

Dieser Vorgang ist zu unterscheiden zwischen Authorization Code Injection. In diesem Fall wird ein Authorization Code eingeschleust, siehe hierzu [13]. Zur Beschreibung der Attacke, siehe Kapitel 4.5.1.

5.6.2 Wer kann den Angriff auslösen

Jeder (Web)Server, welcher SAML oder JSON Web Token absetzen kann und beim Ziel-Server oder User-Agent empfangen wird.

5.6.3 Bedrohung

Auslösen von ungewollten Befehlen im User Agent oder beim Ziel-Server.

5.6.4 Gegenmassnahmen

Gegenmassnahmen zu Code Injection: Entsprechende Prüfung des Inhalts im Token. Gegenmassnahmen zu Authorization Code Injection, siehe [13] Kapitel 4.5.3.

5.7 Verschiebung der XML-Signatur in einem SAML Token

5.7.1 Beschreibung des Angriffs

Eine Signatur und der davon erfasste Inhalt (Unterobjekte) werden in einem XML-Objekt verschoben. Ersetzt wird der Inhalt (die Unterobjekte) durch einen anderen Inhalt (andere Objekte). Wenn nun die Signaturprüfung losgelöst von Inhaltsprüfung erfolgt, so können die ersetzen Unterobjekte akzeptiert werden, weil die Signaturprüfung erfolgreich war, obwohl sie im Hauptobjekt verschoben wurde. Durch das Verschieben der Signatur und der von ihr erfassten Unterobjekte hat die Signatur nicht an Gültigkeit verloren. Aber das, was die Applikation erhält, ist nicht mehr von der Signatur geschützt. Für Details, siehe eCH-0091, Seite 17, Spalte 2, [7], [8]. Der Angriff wird aus Signature Wrapping bezeichnet.

5.7.2 Bedrohung

Ungewolltes Akzeptieren des Inhalts einer Request oder Response durch die RP oder den IdP.

5.7.3 Gegenmassnahmen

eCH-0091, 2.0.0. Seite 17 «Zusammenspiel zwischen XML-Anwendung und XML-Signaturprüfung», [7], [8].

5.8 Certificate Impersonation

5.8.1 Beschreibung des Angriffs

Der User wird dazu verleitet, eine TLS-Verbindung zu akzeptieren, welche auf einem Server Zertifikat beruht, welches nicht von der geforderten CA ausgestellt worden ist.

Bsp.: Ein Zertifikat mit dem Eintrag www.admin.ch wird nicht von der Admin PKI ausgestellt, sondern von der CA Trust Me. Das Root-Zertifikat zu dieser CA ist im Browser enthalten.

5.8.2 Bedrohung

Unbeabsichtigter Aufbau einer TLS-Verbindung und ungewollter Austausch vertraulicher Daten mit einem Server.

5.8.3 Wer kann den Angriff auslösen

Jede CA, deren Root-Zertifikat im Browser als vertrauenswürdig erachtet wird.

5.8.4 Gegenmassnahmen

MUST: Damit nur die gewünschten TLS-Zertifikate für die Server Authentisierung verwendet werden, muss der IdP einen CAA Eintrag im DNS vorweisen. Zudem muss er die Ausstellung von Zertifikaten seiner Domain überwachen.

SHOULD: Die RP soll dies auch vornehmen.

Quelle: RFC 6844

5.9 Denial of Service

5.9.1 Beschreibung des Angriffs

Mittels einer Flut an Anfragen an einen Server wird dessen Kommunikationskapazität eingeschränkt oder gar die Kommunikation lahmgelegt.

5.9.2 Bedrohung

Die Verfügbarkeit der Kommunikation mit dem Server wird beeinträchtigt. Da der IdP Dreh- und Angelpunkt für die Authentisierung ist, würde eine Reduktion dessen Kommunikationskapazität zu einer Beeinträchtigung oder gar zu einem Unterbruch der Kommunikation zwischen einem User und der RP führen.

5.9.3 Gegenmassnahmen

MUST: Der IdP muss wirksame Massnahmen zum Schutz vor verschiedenen Denial of Service Angriffen implementieren. Z.B. mittels Rate-Limit, Captcha, IP Tracking.

5.10 Brute-Force / Exploration

5.10.1 Beschreibung des Angriffs

Brute Force Angriffe beabsichtigen, den Usernamen und das Passwort oder ein Token zu bestimmen, um nicht autorisierten Zugang zu erhalten. Brute Force Angriffe basieren auf Versuch und Irrtum, wobei die gewünschten Informationen erraten werden und dann verifiziert werden.

Bei Exploration Angriffen will man u.a. das Verhalten der Kommunikationsteilnehmer auskundschaften und daraus Erkenntnisse gewinnen.

5.10.2 Bedrohung

Unbeabsichtigte Zugangsberechtigung oder Einsichtnahme in Daten durch einen Dritten.

5.10.3 Wer kann den Angriff auslösen

Jeder, welcher eine Verbindung mit dem Server aufbauen kann.

5.10.4 Gegenmassnahmen

MUST: Der IdP muss folgende Massnahmen zum Schutz vor Brute-Force und Exploration Angriffen implementieren: Rate-Limit, Captcha, IP Tracking.

6 Vergleich PKI-Ansatz – OIDC und SAML

In [11] wird ein Sicherheitsvergleich zwischen einem PKI-Ansatz und dem hier zugrunde liegenden IAM-Modell aus User Agent, RP und IdP vorgenommen. Im Wesentlichen spricht puncto Sicherheit Folgendes für den PKI-Ansatz:

- Höhere Verfügbarkeit
- Weniger komplexer Verbindungsaufbau (beim PKI Ansatz ist eine Komponente weniger involviert. Deswegen sind die Empfehlungen betreffend die RP im Kapitel 4 «Registrieren und Authentisieren» nicht von Bedeutung.
- Die Nachvollziehbarkeit ist einfacher herzustellen
- Der User hat weniger zu prüfen.
- Schlechte Unterstützung der Browser (User Experience)
- Bessere Performance beim Verbindungsaufbau

Erläuterungen hierzu und weitere Vorteile für den PKI-Ansatz puncto Netzwerksicherheit können bei der erwähnten Quellenangabe gefunden werden. Die Empfehlungen in den folgenden Kapiteln sind ebenfalls für den reinen PKI-Ansatz relevant:

- Kapitel 3 «Allgemeine Empfehlungen»
- Kapitel 5 «Angriffstechniken»

7 Zusammenfassung

7.1 Beeinträchtigung der Authentizität einer E-ID und der Autorisierung

Folgende Attacken können die Authentisierung mit elektronischen Identitäten (E-ID) aufgrund mangelnder Vorkehrungen umgehen, ohne dass dem Inhaber einer E-ID eine mangelnde Sorgfalt im Umgang mit der E-ID vorgeworfen werden kann:

- Unsorgfältige Registrierung der Herstellung der elektronischen Identität
- CSS/XSS
- CSRF/XSRF
- UI-Redressing (Click Jacking)
- Session Hijacking
- Code Injection und Authorization Code Injection
- Verschiebung der XML-Signatur in einem SAML Token (XML Signature Wrapping)

Vorkehrungen/Massnahmen gegen die erwähnten Attacken sind bei den Servern zu treffen und können vom User nicht beeinflusst werden. Dabei kann der in IT-Sicherheit nicht versierte User auch nicht erkennen, ob die Sicherheitsmassnahmen umgesetzt wurden.

7.2 Zu prüfen durch den User

Der User hat Folgendes zu prüfen:

- URL der RP
- URL des IdP (nicht, dass er auf eine andere Webseite umgeleitet wird und dort sein Passwort eingibt.)
- Ob ein zur RP separater Frame für die Verbindung zum IdP geöffnet wird.
- Ob beim der Anfrage des IdP um sein Einverständnis, die Daten an die RP zu senden, die dort aufgeführte URL mit der URL der angewählten RP übereinstimmt.

Anmerkung: Der User hat bei einem Verbindungsaufbau mehr als mit reinem PKI Ansatz zu prüfen! Folglich ist es schwieriger, die Prüfung wie gefordert vorzunehmen und nachzuweisen, dass die Prüfung vorgenommen wurde oder nicht.

7.3 Bedrohung/Schwachstelle

Eine Tabelle mit einer Gegenüberstellung Schwachstelle/Bedrohung ergibt nicht viel, weil sich hier fast alles um das Thema, resp. um die Bedrohung «Impersonation» dreht. Mit Ausnahme in Kapitel:

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

In Kapitel 5 «Angriffstechniken» sind Bedrohungen/mögliche Angriffe aufgeführt und jeweils Gegenmassnahmen dazu empfohlen.

8 Haftungsausschluss/Hinweise auf Rechte Dritter

eCH-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellen oder welche **eCH** referenzieren, haben nur den Status von Empfehlungen. Der Verein **eCH** haftet in keinem Fall für Entscheidungen oder Massnahmen, welche der Benutzer auf Grund dieser Dokumente trifft und / oder ergreift. Der Benutzer ist verpflichtet, die Dokumente vor deren Nutzung selbst zu überprüfen und sich gegebenenfalls beraten zu lassen. **eCH**-Standards können und sollen die technische, organisatorische oder juristische Beratung im konkreten Einzelfall nicht ersetzen.

In **eCH**-Standards referenzierte Dokumente, Verfahren, Methoden, Produkte und Standards sind unter Umständen markenrechtlich, urheberrechtlich oder patentrechtlich geschützt. Es liegt in der ausschliesslichen Verantwortlichkeit des Benutzers, sich die allenfalls erforderlichen Rechte bei den jeweils berechtigten Personen und/oder Organisationen zu beschaffen.

Obwohl der Verein **eCH** all seine Sorgfalt darauf verwendet, die **eCH**-Standards sorgfältig auszuarbeiten, kann keine Zusicherung oder Garantie auf Aktualität, Vollständigkeit, Richtigkeit bzw. Fehlerfreiheit der zur Verfügung gestellten Informationen und Dokumente gegeben werden. Der Inhalt von **eCH**-Standards kann jederzeit und ohne Ankündigung geändert werden.

Jede Haftung für Schäden, welche dem Benutzer aus dem Gebrauch der **eCH**-Standards entstehen ist, soweit gesetzlich zulässig, wegbedungen.

9 Urheberrechte

Wer **eCH**-Standards erarbeitet, behält das geistige Eigentum an diesen. Allerdings verpflichten sich die Erarbeitenden, ihr betreffendes geistiges Eigentum oder ihre Rechte an geistigem Eigentum anderer, sofern möglich, den jeweiligen Fachgruppen und dem Verein **eCH** kostenlos zur uneingeschränkten Nutzung und Weiterentwicklung im Rahmen des Vereinszweckes zur Verfügung zu stellen.

Die von den Fachgruppen erarbeiteten Standards können unter Nennung der jeweiligen urhebenden Person von **eCH** unentgeltlich und uneingeschränkt genutzt, weiterverbreitet und weiterentwickelt werden.

eCH-Standards sind vollständig dokumentiert und frei von lizenz- und/oder patentrechtlichen Einschränkungen. Die dazugehörige Dokumentation kann unentgeltlich bezogen werden.

Diese Bestimmungen gelten ausschliesslich für die von **eCH** erarbeiteten Standards, nicht jedoch für Standards oder Produkte Dritter, auf welche in den **eCH**-Standards Bezug genommen wird. Die Standards enthalten die entsprechenden Hinweise auf die Rechte Dritter.

Anhang A – Referenzen & Bibliographie

Bibliographie

[1]	Anhang 8 Verordnung des EDI vom 22. März über das elektronische Patientendossier.
[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. Oktober 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/.

Standards

eCH-0091	Standard zu XML-Signatur und -Verschlüsselung / 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 der ITU für Zertifikate
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, incorperating errata set 1, OpenID Foundation
OPENID Connect	OpenID Connect Core 1.0 incorperating 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

Anhang B – Mitarbeit & Überprüfung

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

Anhang C – Abkürzungen und Glossar

AES	Advanced Encryption Standard
Bedrohung	Ein sich konkretisierende Gefahr
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
Gefahr	Möglichkeit, dass sich Unerwünschtes ereignen kann.
HMAC	hash-based message authentication code
HSTS	HTTP Strict Transport Security
HTTP	Hypertext Transfer Protocol
IAM	Identity and Access Management
IdP	Anbieter von elektronischen Identitätsdienstleistungen
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 und Server authentisieren sich mittels des TLS-Protokolls auf Basis von X.509 Zertifikaten

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	Eine Informatikanwendung, welche einen IdP dazu benutzt, die Authentisierung (elektronische Identifizierung) des Users vorzunehmen.
RSA	Public Key Verfahren von Rivest, Shamir und Adleman
SAML	Security Assertion Markup Language
Schwachstelle	Eine Schwachstelle ist, wie der Name bereits ausdrückt, eine Begebenheit, welche eine oder mehrere Schwächen aufweist und somit das Risiko in sich birgt, dass sich aufgrund derer ein Schaden ereignen kann. Eine Schwachstelle ergibt sich u.a. aufgrund mangelnder oder ungenügender Sorgfalt (Tun oder Unterlassen). Ob etwas als Schwachstelle taxiert wird und gegebenenfalls durch Massnahmen behoben werden muss/sollte, ergibt sich aus Vorschrift oder aus dem Risikomanagement.
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

Anhang D – Änderungen gegenüber Vorversion

Dies ist die erste Version.

Anhang E – Abbildungsverzeichnis

Keine

Anhang F – Tabellenverzeichnis