Vergleich von online Authentisierungen im eGov Bereich

In diesem Aufsatz wird das Single Sign-On Verfahren, insbesondere noch die OpenID Connect Authentisierung, mit einem hier vorgeschlagenen Authentisierungsverfahren betreffend technische Sicherheit verglichen.

Florian Forster (https://caos.ch/), Daniel Muster (www.it-rm.ch)

V.1.2, 13. Oktober 2020

Inhaltsverzeichnis

I Einleitung

I.1 Zielpublikum

Das Zielpublikum dieses Aufsatzes sind IT-Sicherheitsfachleute, welche sich mit online Authentisierungsverfahren auseinandersetzen. Deswegen wird hier auf viele technische Erläuterungen verzichtet und auf einschlägige Standards und technische Abhandlungen verwiesen.

I.2 Überblick

Zuerst wird ein Lösungsansatz eines Authentisierungsverfahrens auf Basis von mutual TLS (auch bekannt als client certificate authentication) präsentiert. Dieses wird puncto Sicherheit mit den prinzipiellen Eigenschaften eines Single Sign-On Verfahren, kurz SSO-Verfahren, verglichen. Basis dieses Vergleichs bildet ein Ablauf eines SSO-Verfahrens mit einem Browser, wie es in CLAIMS auf Seite 273 dargestellt ist.

Im Anschluss wird OpenID Connect mit dem hier vorgeschlagenen Lösungsansatz puncto technische Sicherheit verglichen werden. OpenID Connect ist ein von der OpenID Foundation standardisiertes SSO-Verfahren, siehe OPENID CONNECT.

Nur vereinzelt werden organisatorische/rechtliche Aspekte beim Vergleich hinzugezogen, sofern diese technische Implikationen haben.

Anmerkung: Quellenverweise werden in GROSSBUCHSTABEN angegeben.

II Begriffswahl

Folgende Begriffe werden verwendet:

Web-Client: Browser auf einem Kommunikationsgerät, welcher von einem User bedient wird, welcher sich bei einem Web-Dienst anmeldet. Synonyme sind: User-Agent (OPENID CONNECT)

User: Anwender, welcher den Web Client bedient und sich bei einem Web-Dienst anmelden will. Dies entspricht einem E-ID Inhaber (BGEID)

Web-Dienst: Web Server, bei welchem sich der User anmelden, Informationen oder Dienste beziehen will. Synonyme sind: Client (OAUTH) Relying Party (OPENID CONNECT), E-ID-Dienst (BGEID).

Identity Provider (IdP): Ein Server, welcher den User für den Web-Dienst authentisiert. Synonyme sind: OpenID Provider (OPENID CONNECT), Authorization Server (OAUTH), E-ID-System (BGEID)

Certification Authority (CA): Ein nach ZertES anerkannter Herausgeber von elektronischen Zertifikaten.

Crypto Chip: Mikroprozessor, auf welchem die kryptographischen Funktionen durchgeführt werden, wie z.B. die Operation mit einem asymmetrischen Schlüssel und einem Hashwert. Die dort

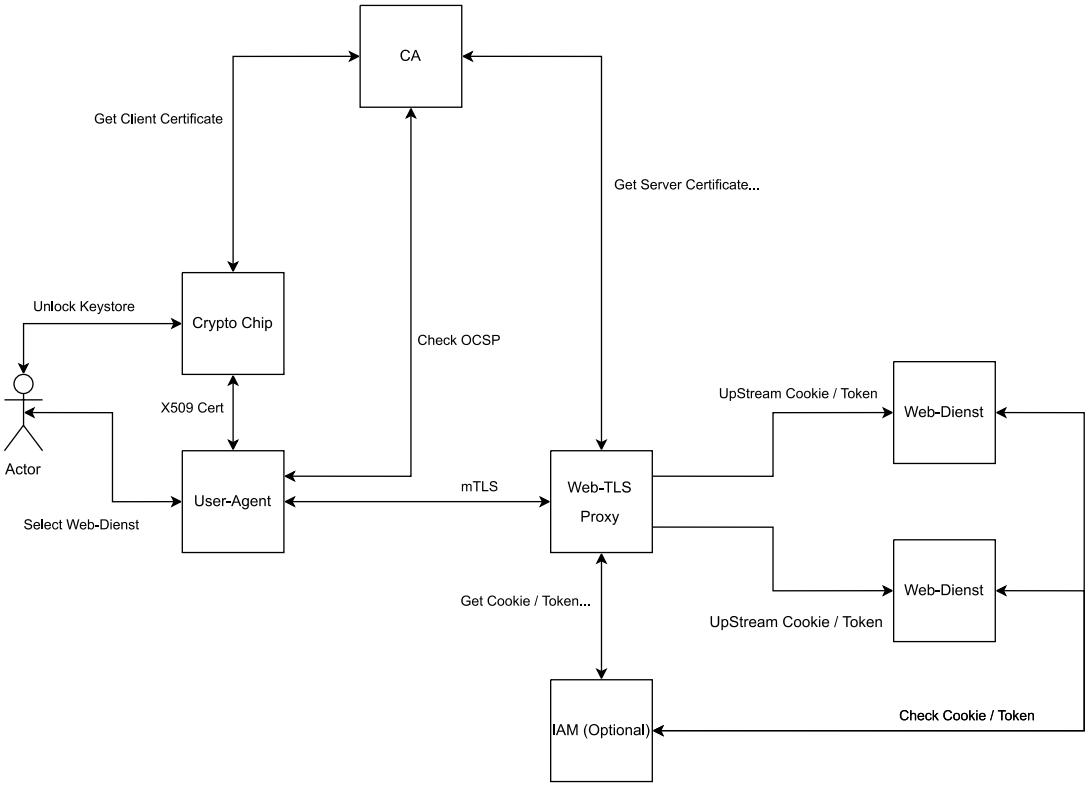
gespeicherten geheimen Schlüssel sollten nicht herausgelesen werden können. Das Auslösen einer kryptographischen Funktion ist mittels Knowledge-Factor (PIN / Password) geschützt.

III Alternativer Lösungsansatz

Bei diesem Lösungsansatz wird die Authentisierung des Users nicht beim Identity Provider (zentral) vorgenommen, sondern von einem Web/TLS-Proxy (dezentral). Die Authentisierung des Users findet auf der TLS-Ebene statt, d.h. der User authentisiert sich mit einem nach ZertES geregelten Zertifikat beim Web/TLS Proxy. Die http-Verbindung zwischen Web-Client und Web/TLS-Proxy wird mit TLS bezüglich Vertraulichkeit/Authentizität und Integrität abgesichert. Die privaten Schlüssel des Users sind auf einem Crypto Chip geschützt.

Der Web/TLS-Proxy leitet die http-Anfragen des Web-Clients an den Web-Dienst weiter. Er stellt dabei einen Link zwischen der TLS Session ID und dem vom Web-Dienst gesetzten Cookie her. Das Cookie wird jedoch nicht an den Web-Client weitergeleitet, sondern verbleibt im Session Storage des Web/TLS-Proxy. Über die Session ID wird die TCP/IP-Kommunikation zwischen Web/TLS-Proxy und Web-Client, sowie mit dem Cookie zwischen Web/TLS-Proxy und Web-Dienst verbindungsorientiert. Der Web/TLS-Proxy authentisiert sich gegenüber dem Web-Client auf Basis eines nach ZertES geregelten Zertifikats.

In folgender Graphik ist das Zusammenspiel der einzelnen Komponenten des Verfahrens dargestellt.



Anmerkung: Der Einsatz von nach ZertES geregelten Zertifikaten bildet wegen der Haftung gemäss ZertES und OR im Allgemeinen eine grosse Verlässlichkeit.

Begriff/Abkürzung: Diese Art der Authentisierung wird hier als mutual TLS Authentication without Cookies (mTLSwC) bezeichnet.

IV Vergleich

IV.1 Annahmen

Damit der Vergleich zwischen dem mTLSwC und einem SSO-Verfahren sinnvoll ist, wird angenommen, dass sich alle an der Authentisierung beteiligten Parteien beim SSO-Verfahren mit mTLS auf Basis eines nach ZertES geregelten Zertifikats authentisieren. Die ausgetauschten Tokens werden mit einer nach ZertES geregelten Signatur geschützt.

Es findet keine dynamische Registrierung der Relying Party beim IdP statt. Dies kann z.B. bei OpenID Connect Sicherheitsprobleme verursachen, siehe MLADENOW-MAINKA.

Die Interaktionen des Users mit dem Web-Dienst und dem IdP findet stets über je ein anderes Browser-Fenster statt. Weiter erfolgt die Autorisierung beim Web-Dienst (siehe auch optionales IAM in der Grafik) und nicht beim IdP.

Für die Authentisierung mit OpenID Connect werden zusätzlich folgende Annahmen getroffen:

- Das JSON-Token wird vom IdP mit einer nach ZertES geregelten Signatur versehen, somit bezüglich Authentizität und Integritätsverlust geschützt.
- Alle 3 Parteien (IdP, Web-Dienst, User) gehören zu einer voneinander rechtlich unabhängigen Organisation. Dies ist im eGovernment oft der Fall.

IV.2 Grundsätzliche Unterschiede zwischen mTLSwC und einer SSO-Lösung

In diesem Unterkapitel werden zuerst die grundsätzlichen Unterschiede zwischen dem mTLSwC-Lösungsansatz und einer SSO-Authentisierung dargelegt. Basis des Vergleichs bildet ein Ablauf eines SSO-Authentisierungsverfahrens mit einem Browser, wie es CLAIMS auf Seite 273 dargestellt ist.

IV.2.1 Nachvollziehbarkeit

Die Nachvollziehbarkeit des Authentisierungsverfahrens auf Basis von Log-Einträgen verlässlich zu etablieren, gestaltet sich bei einer SSO-Lösung schwieriger, wenn nicht unmöglich, falls 3 Parteien unterschiedlicher Organisationen mit unterschiedlichen Interessen am Kommunikationsaufbau beteiligt sind.

IV.2.2 Identitätsdiebstahl

Der IdP kann sich als ein bei ihm registrierter User ohne wesentlichen Aufwand beim Web-Dienst authentisieren lassen und im Namen dieses Users Transaktionen veranlassen. Dies kann kaum oder

nicht erkannt werden, insbesondere wenn die Nachvollziehbarkeit nicht verlässlich etabliert worden ist.

Beim mTLSwC-Lösungsansatz ist dies für eine CA auch möglich, doch dies kann festgestellt werden, wenn die Ereignisse beim Web-Dienst registriert (geloggt) werden und der User seine privaten Schlüssel selber generiert, d.h. nicht von der CA erzeugen lässt und lediglich den Zertifikatsaustellung bei der CA beantragt (Certificate Signing Request). Das Zertifikat für den Identitätsdiebstahl stimmt also nicht mit dem Zertifikat des Users überein.

IV.2.3 Verfügbarkeit

Mit Einbezug eines weiteren, für das Funktionieren notwendigen Servers für die Authentisierung beim SSO werden die Authentisierung und die damit verbundenen Dienste weniger als bei mTLSwC-Lösung verfügbar. Es liegt in der Natur der Sache, dass die Verfügbarkeit mit steigender Anzahl Komponenten sinkt. Siehe dazu auch CALCULATING AVAILABILTY, RAUSAND-HOYLAND.

IV.2.4 Aufwand der Benutzeradministration

Bei beiden Lösungsansätzen muss/sollte die Berechtigung des Users beim Web-Dienst zugeordnet werden. Aus Sicht des Web-Dienstes fragt sich nun, welchen Vorteil ein SSO-Lösungsansatz bietet.

IV.2.5 Token Handling

Beim SSO-Ansatz muss zusätzlich ein oder mehrere Token (Claims) gehandhabt werden, z.B. die Signatur des Token geprüft werden. Die Schnittstelle zwischen Signaturprüfung und Applikation kann fehleranfällig sein, was Möglichkeiten für einen Hackerangriff eröffnet, siehe z.B. BREAK-SAML.

Zudem muss der Browser bei mTLSwC keine Cookies verwalten, insbesondere sicher aufbewahren.

IV.2.6 Benutzerinteraktion

Der Benutzer muss beim SSO-Ansatz 2-mal ein Zertifikat eines Servers (Web-Dienst, IdP) eingehend prüfen, während bei mTLSwC nur einmal.

IV.2.7 Registrierung

Der Web-Dienst muss sich nicht bei einem Server (IdP) registrieren lassen.

IV.2.8 Mehr zu prüfen

Wenn mehr zu prüfen ist, dann bietet dies mehr Möglichkeiten für Fehler und somit für mehr Sicherheitslücken. Oder: Je mehr Kontrolleschritte in einem System durchzuführen sind, desto mehr Fehlerquellen bestehen.

IV.2.9 Anmerkung zu TLS

Der Verbindungsaufbau bei TLS ist anfällig für eine Trapdoor, welche ohne Source Code Analyse nicht/kaum festgestellt werden kann, siehe hierzu SOCIETY BYTE.

IV.3 Zusätzliche Unterschiede zwischen mTLSwC und OpenID Connect

IV.3.1 Mischung von Programmen und Daten

Bei mTLSwC werden für das Errichten der gesicherten Kommunikation (TLS) lediglich Daten ausgetauscht. Es bietet nicht wie bei OpenID Connect die Möglichkeit, Programme (JavaScript) und Daten beim Aufbau der gesicherten Kommunikation zu vermischen.

IV.3.2 Cross Site Request Forgery (CSRF)

Weil für den Verbindungsaufbau bei SSO ein Cross Site Request (CSR) durch die RP an den IdP erforderlich ist, ist die RP in „guter“ Position, einen CSRF-Attacke zu initiieren oder zu unterstützen (direkt oder indirekt). Wie die Attacke genau funktioniert, ist u.a. in STUTTARD/PINTO und in SULLIVAN/LIU erläutert. Folglich sind Abwehrmassnahmen gegen eine solche Attacke erforderlich. U.a., weil sich der Benutzer bei einem „falschen“ Web-Dienst angemeldet hat, oder der Web-Dienst gehackt wurde. Eine Reihe von Massnahmen gegen eine CSRF ist bei OWASP-CSRF und in OAUTH SECURITY, Kapitel 4.7, zu finden.

Beim dezentralen Ansatz sind keine Cross Site Request für den gesicherten Verbindungsaufbau zum Web-Dienst erforderlich. Trotzdem empfiehlt es sich auch dort, einen Schutz gegen eine CRSF einzubauen. Z.B., wenn der Benutzer zur bestehenden Verbindung eine weitere zu einem Web-Dienst in einer anderen Domäne aufbaut.

IV.3.3 Click Jacking (UI-Redress)

Da mindestens 2 Websites eim gesicherten Verbindungsaufbau mit einer SSO-Browser-Lösung benötigt werden, macht es den Browser, beziehungsweise den Benutzer anfällig für Click Jacking Attacken durch die RP.

Die Attacke ist in STUTTARD/PINTO, Schwenk und ZALEWSKI, sowie mögliche Gegenmassnahmen beschrieben.

V Quellenangaben

V.1 Technik

BREAK-SAML	On Breaking SAML: Be Whoever You Want to Be, Juraj Somorovsky, Andreas Mayer, Jörg Schwenk, Marco Kampmann, and Meiko Jensen
CALCULATING AVAILABILTY	Hoda Rohani, Azad Kamali Roosta, Calculating Total System Availability, https://www.delaat.net/rp/2013-2014/p17/report.pdf
CLAIMS	Dominick Baier et. al, A guide to Claims-Based Identity Access Control, Microsoft, 2nd Edition, 2011. Kann vom Internet heruntergeladen werden.
DIG	NIST Special Publication 800-63-3, Digital Identity Guidelines
JSON	JavaScript Object Notation (JSON) Data Interchange Format, RFC 7159
JSON-SIG	JSON Web Signature, RFC 7515
KRAUTWALD	Julian Krautwald, Single Sign-On – OpenID Connect(ing) People, Master Thesis, Ruhr Universität Bochum, 2014
MLADENOW-MAINKA	Vladislav Mladenow, Christian Mainka, OpenID Connect Security Consideration, Ruhr Universität Bochum, 2017
OAUTH	The OAUTH 2.0 Authorization Framework, RFC 6749, IETF
OAUTH SECURITY	OAuth Security Best Practices, https://tools.ietf.org/id/draft-ietf-oauth-security-topics-13.html
OAUTH BEARER	OAUTH 2.0 Authorization Framework: Bearer Token Usage, RFC 6750, IETF
OAUTH THREAT	OAUTH 2.0 Threat Model and Security Consideration, RFC 6819, IETF
OPENID CONNECT	OpenID Connect Core 1.0 incorperating errata set 1, OpenID Foundation
OWASP-CSRF	Cross-Site Request Forgery Prevention Cheat Sheet, https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html#synchronizer-token-pattern
RAUSAND HOYLAND	Marvin Rausand, Arnlijot Hoyland, System Reliability Theory, Second Edition, Wiley, 2003
RISTIC	Ivan Ristic, Bullet Proof SSL and TLS, Feisty Duck, 2014
SCHWENK	Prof. Dr. Jörg Schwenk, Netzwerksicherheit 3, Ruhr Universität Bochum, 5. Auflage, 2017
SOCIETY BYTE	Daniel Muster, Völlig Diffuse Sicherheit bei Internetanwendungen. https://www.societybyte.swiss/2017/07/25/voellig-diffuse-sicherheit-bei-internet-awendungen/, Juli 2017 - Englische Version unter: http://www.it-rm.ch/files/Trapdoor_SSL_TLS_II.pdf, July 2018
STUTTARD /PINTO	Darfydd Stuttard, Marcus Pinto, Hacker's Handbook, 2nd edition, Wiley, 2011

SULLIVAN/LIU	Bryan Sulllivan, Vincent Liu, Web Application Security, McGraw Hill, 2012
ZALEWSKI	Michal Zalewski, Tangled Web, dpunkt Verlag, 2013

V.2 Vorschriften

BGEID	Bundesgesetz über elektronische Identifizierungsdienste vom 27. September 2019. Gegen dieses Gesetz kam das Referendum zustande. Die Volksabstimmung darüber steht noch aus.
ZertES	Bundesgesetz über Zertifizierungsdienste im Bereich der elektronischen Signatur und anderer Anwendungen digitaler Zertifikate vom 18. März 2016, SR 943.03