Exposé zur Bachelorarbeit

Klassische User-ID/Passwort Authentifizierung vs. neuartige Loginansätze

Ertugrul Sener

30. Juli 2020

Forschungsfrage:	Inwiefern bietet die Authentifikation ohne Passwort Vor- und Nachteile gegenüber Javascript basierten Web-Anwendungen hinsichtlich Nutzerfreundlichkeit, Datenschutz und Sicherheit?
Bildungseinrichtung:	Hochschule für Telekommunikation Leipzig (HfTL)
Student:	Ertugrul Sener
Matrikelnummer:	S178183
Studiengang:	DAI17
Abgabedatum:	30. Juli 2020

Inhaltsverzeichnis

1 Einleitung und Motivation

Viele Unternehmen setzen auch heute noch auf die BenutzerID, um einen Nutzer zu authentisieren und auf das Nutzerpasswort zur Authentifizierung. Dies wirft vor allem bei der Übertragung einige Datenschutzprobleme auf, die es Angreifern ermöglichen, Daten abzugreifen und damit Identitäten vorzutäuschen, die vom Server anerkannt und authentifiziert werden. Dies gibt dem Angreifer einen Vollzugriff auf den Nutzer und dessen Daten. Der Begriff 'Clientless Login' ist eine Wortneuschöpfung und bezeichnet die Authentifikation an einem Zielsystem ohne den Nutzen von statischen Passwörtern, dabei können sowohl hardware als auch softwarebasierte Schlüssel oder eine Kombination aus beiden bei der Mehrfaktor-Authentifizierung, genutzt werden. Ich werde den heutigen Stand der Technik und den Stand der Forschung darlegen und am Ende anhand von den Kriterien Nutzerfreundlichkeit, Datenschutz und Sicherheit meine eigene Architektur bewerten und mit den bereits existierenden Lösungen vergleichen. Dabei soll die Frage beantwortet werden, inwiefern moderne Methoden Vorteile gegenüber der klassischen Methoden bieten und vorallem wo ihre Schwächen liegen. Prototypisch soll eine Oberfläche hierzu entwickelt werden, um die Authentifikation durchzuführen. Die Hauptmotivation für die Bearbeitung dieser Forschungsfrage liegt einerseits an meinem Interesse für Probleme in der IT-Sicherheit und andererseits an meiner Hauptbeschäftigung im Betrieb bei der Telekom Security im Bereich des Software Engineering. Derzeit arbeite ich mit drei anderen Kollegen an einer Monitoring - Software namens CRD (Controlled Remote Desktop). Mit dieser Anwendung baut man mittels einer nativen Java Anwendung eine Verbindung zu entfernten Rechnern auf. Die Aktivitäten des Rechners werden visuell aufgezeichnet und-/oder in Logdateien gesichert. Gleichzeitig kann der Portalbenutzer (externer Nutzer) sich auf den entfernten Rechner aufschalten und live zusehen. Dadurch, dass jede Anwendung bei jedem Mitarbeiter auf dem Rechner liegt ergeben sich Probleme in der Handhabung mit Updates und Patches. Zusätzlich müssen bei jeder Verbindung die aufgebaut wird, die richtigen Eingabeparameter gesetzt werden. Diese wären Protokoll das genutzt wird und eine Ticket-ID sowie die Auflösung der darauffolgenden Sitzung (sofern grafisch). Aus diesen Gründen sollte eine Konsole (namentlich 'CRD-Konsole') im Webumfeld entwickelt werden, die diese Funktionalitäten zentral auslagert. Für die Authentifikation auf dieses Portal wollte man allerdings nicht nur die übliche User-ID/Passwort Authentifikation implementieren, sondern weitere Funktionalitäten ermöglichen. Zu den ausgewählten Verfahren gehören OTP's (One Time Passwords) und public key Verfahren wie das Webauthn (Web Authentication).

2 Das Problem mit der Bequemlichkeit

Laut des IT-Grundschutz-Kompendiums vom Bundesamt für Sicherheit in der Informationstechnik könne ein Benutzer aus Bequemlichkeit oder pragmatischen Gründen bewusst auf komplizierte und unhandliche Kryptomodule verzichten und Informationen stattdessen im Klartext übertragen. [6] Dies stellt ein hohes Sicherheitsrisiko für Unter-

nehmen, aber auch Privatpersonen dar, da Benutzer nicht gewillt sind ihre Passwörter durch komplizierte Verfahren zu erzeugen und in regelmäßigen Abständen vollkommen randomisiert zu setzen. An neuartige Ansätze des Logins in nativen oder webbasierten Anwendungen stellen sich dadurch völlig neue Herausforderungen. So müssen neue Authentifizierungsmöglichkeiten nicht nur sicher sein, sondern auch komfortabel genutzt und bedient werden können, da sie sonst von den Endnutzern gemiedern oder umgangen werden. Wichtig ist eben auch, dass die breite Masse Zugriff auf die Ressourcen hat, die es zur Nutzung dieser Verfahren braucht. Man denke nur an die ganzen betrieblichen Passwörter, bei denen zum Monatsende nur eine Zahl an der letzten Stelle des Passwortes geändert wird. Laut einer Statistik von 2019, der "Global Data Risk Report From the Varonis Data Lab", gaben 38% aller Nutzer an ein Passwort im Unternehmen zu nutzen, dass sie nicht (oder nur geringfügig) ändern. Außerdem wird laut dieser Statistik alle 364 Tage ein Data-Breach aufgrund von unsicheren Passwörtern in einem mittelständigen Unternehmen stattfinden. Die monatliche Ablaufzeit von Passwörtern in Unternehmen scheint also nicht ganz den Effekt zu erzielen, der ursprünglich damit geplant war, da die Arbeiter die Bequemlichkeit über die Sicherheit stellen.

3 Authentifizierungsstrategien

3.1 UserID/Passwort

Egal ob auf dem Mobilgerät, dem Computer oder der Spielekonsole: Häufig wird ein Nutzer dazu aufgefordert, "[...] seine Identität durch die Eingabe eines Passwortes nachzuweisen"[2]. Diese Bestätigung spielt eine ëssenzielle Rolle"[2] im sowohl Privatleben als auch der Unternehmenswelt. Mit einem Passwort wird die Identität eines Nutzers nachgewiesen. Im Beispiel einer Online-Plattform registriert sich der User mit seiner E-Mail Adresse (vgl. User-ID) und seinem Passwort auf einer Webseite. Darauffolgend erhält dieser einem Bestätigungslink zum Identifikationsnachweis. Nach dem Klick auf den Hyperlink, stellt die E-Mail Adresse des Nutzers "die neue Identität des Nutzers [dar]"[2]. Mit diesen Daten muss sich der Nutzer dann bei jedem neuen Anmeldevorgang authentifizieren. Für eine allgemeine Authentifizierung gibt es drei Möglichkeiten, unzwar die des Wissens, des Besitzes oder körperlichen Merkmale. [7] Das Wissen in unserem Beispiel könnte ein Kennwort sein, das nur dem Nutzer bekannt ist, ein Beispiel für die Authentifizierung mit einem Besitz findet sich im Unternehmen der deutschen Telekom sehr schnell: Die SmartCards, über die wir uns gegenüber unseres Rechners authentifizieren. Zuletzt dienen biometrische Daten wie der einmalige Fingerabdruck oder die Gesichtserkennung der Authentifikation.

3.2 Einmalkennwort

3.2.1 Timerbasiert

Ein Passwort (auch Kennwort genannt) ist per Definition ëine Zeichenfolge, die zur Authentifizierung verwendet wird. Damit soll die Identität einer Person [...] auf eine Res-

source nachgewiesen werden."[4]. Ein Einmalkennwort ist ein Kennwort, das nur ein einziges Mal für eine Authentifizierung genutzt werden kann. Man unterscheidet drei Arten von Einmalkennwörtern (künftig OTP, für Öne Time Password"). Bei zeitbasierten OTPs wird die aktuelle Systemzeit gehasht und an den Server übermittelt. Je nach Toleranz wird diese Zeit mit der Systemzeit des Servers verglichen (unter Beachtung der Toleranz) und bei Erfolg dem Clienten der Login gewährt. Laut Margaret Rouse, mehrfach ausgezeichnet für ihre Publikationen im IT-Umfeld in der New York Times, USA Today und Washington Times und ehemalige Lehrerin für Informatik und Technologieintegration biete das TOTP Verfahren zusätzliche Sicherheit für den Nutzer, da selbst bei Erhalt des Passwortes das TOTP nicht in die Hände des Angreifers gelangt und nach einer gewissen Toleranzzeit verfällt. ([5] Eine Beispielapplikation für die Nutzung eines TOTP - Ansatzes für zusätzliche Sicherheit ist der "Google Authenticator", welches in jedem gängigen App Store zu finden ist.

3.2.2 Ereignisbasiert

Ereignisbasierte OTPs besitzen einen Ereigniszähler, der bei jeder versuchten Authenzifizierung einen Zähler auf Server und Clientseite synchronisiert inkrementiert. Sollte der Zähler asychnron werden bzw. der Server einen anderen Wert gespeichert haben als der Client bei der nächsten Authentifizierung sendet, wird der Authentifizierungsvorgang abgebrochen. Man findet diese Funktionalität wortwörtlich beschrieben in Googles Time and event based one time password Patent [3] in folgendem Wortlaut "[...] the characteristics of an event can be the value of a counter that is incremented each time the user pushes a button on the token"[3]. Für diesen Prozess wird der HOTP Algorithmus genutzt, der im RFC4226 näher beschrieben ist. Diese Art der Authentifizierung kann zum Beispiel für die E-Mail Verifikation und damit die Identifikation (wie zuvor erläutert) genutzt werden.

3.2.3 Challenge-response

Challenge-response basierte OTP Verfahren bedienen sich an komplizierten mathematischen Verfahren. Das heißt, es erfolgt ein ACK (Acknowledge bzw. Initialanstoß zur Authentifizierung). Der Client, berechnet die Response mithilfe der mathematischen Formel und sendet das Ergebnis an den Server. Sollte es einen Match geben, erhält der Client eine Response vom Server, der seine Echtheit bestätigt. Synchronisationsprobleme kann es bei diesem Verfahren entgegen der ereignis oder timerbasierten OTP-Verfahren nicht geben, da die Berechnung dieses 'Schlüssels' vollkommen auf der Clientseite funktioniert. Der Server überprüft diese Rechnung nur mit seinem eigenen Wert, stellt aber keine weiteren Rechnungen oder Umformungen mit diesem Wert an. Der Hauptvorteil dieses Verfahrens ist, dass unabhängig von der Zeit und einem speziellen Ereignis eine Anfrage gestellt werden kann. Der Server kann also seine 'Challenge' abschicken und muss keine 'Response' innerhalb einer festgegebenen Zeit erhalten, um authentifizieren zu können. Dieses Verfahren gilt als besonders sicher, da es auf Serverseite keinen Algorithmus gibt, der sich vorausberechnen lässt.

3.3 Web Authentication

Webauthn ist kurz für die 'Web Authentication'. Zur Verfügung gestellt wurde dieser Standard der Authentifikation 2018 von der FIDO Alliance und dem W3C. [1] Sie ermöglicht eine passwortlose (bzw. benutzerdatenlose, es wird also auch keine User-ID benötigt) Authentifikation durch Tokens (Sicherheitsschlüssel). Für dieses Verfahren wird zunächst ein Buffer aus kryptografischen random Bytes generiert, dass der Verhinderung von 'Bruteforce' (vom webauthn - Guide auch als 'reply attacks' beschrieben) - Angriffen dienen soll. Web Authentication nutzt das vorhandene public-key-Verfahren für Webseiten. Der Standard definiert allerdings nicht welche Art von Schlüssel genutzt wird. Unter den Möglichkeiten zählen der ÜSB security key"[1] oder der "built-in fingerprint sensor"[1]. Webauthn basiert auf vielen bereits vorhandenen Abhängigkeiten der Informatik wie die Standards von HTML5, ECMAScript, COSE (CBOR Object Signing and Encryption COSE, RFC8152) oder dem Nutzen von der Base64url encoding.
Zusammengefasst hat der FIDO2 Standard mit CTAP (dem Protokoll für externe Authentifikationen mit Mobilgeräten) und Webauthn (der Schnittstelle bzw. ÄPI") vorhandene Funktionen definiert, mit der native Authentifizierungsmethoden wie das public-private Key Verfahren auf die Webseite übertragen werden können. Wichtige Beispiele für Web Authentication sind der Yubikey, der USB-Token oder unsere biometrischen Daten (FaceID oder TouchID), die wir täglich in jedem AppStore nutzen, der diese Daten verschlüsselt an eine Webseite bzw. einen öffentlichen Store übermittelt. In unserem Use-Case wollen wir die Authentifizierung anhand von Webauthn als Multi-Faktor nutzen, also als zusätzliche Sicherung neben einem Passwort. Denn wie oben beschrieben, kann das Wissen eines Menschen durch Data Breaches oder menschliches Versagen (Gutglauben) leicht abhanden kommen. Mit einem Yubikey oder einem USB-Token befindet man sich allerdings auf der Ebene des Besitzes, wodurch ein potenzieller Angreifer es schwerer hat an die Daten zu kommen. Biometrische Daten und diese public-private-Key Verfahren gelten nämlich allgemein als sehr sicher. Wodurch es technisch sehr schwer bis mathematisch (in gegebener Menschenzeit) fast unmöglich ist, die Algorithmen hinter ihnen zu knacken.

4 Bewertung der Ansätze

Bei der Bewertung der neuartigen Ansätze lässt sich sagen, dass die Technik für diese Ansätze sehr viel älter ist und meist auch aus dem nativen (klassischen) Bereich der Software Entwicklung kommt. Mit Schlüsselverfahren war man auch vor der Erfindung von Webauthn vertraut. Neu ist die Möglichkeit, nun auch biometrische Daten für die Authentifikation im Web zu verwenden oder seinen privaten Schlüssel weiterhin auf seinem Gerät (ob nun Smartphone oder Desktop PC) zu sichern und nur den öffentlichen Schlüssel an Server zu übermitteln, um sich mit seinem privaten Schlüssel zu authentifizieren. Die mathematischen Konzepte hinter Webauthn und der Einmalpasswörter basieren auf Primzahlen und werden (sofern Quantencomputer nicht in wenigen hundert Jahren zum Standard werden) wohl nicht geknackt werden. Das macht diese

Verfahren, die teilweise auf Hashing beruhen extrem sicher. Im Gegensatz zur Übermittlung eines Passwortes im Klartext ist es Angreifern extrem schwer an den privaten Schlüssel des Nutzers zu gelangen, da dieser auf dem Gerät selbst liegt. Sollte das Gerät kompromittiert werden, schlagen allerdings beide Ansätze, ob nun klassich oder modern, fehl. Hinsichtlich der Bequemlichkeit lässt sich sagen, dass das timerbasierte One Time Password die angenehmste Art ist eine zwei Faktor Authentifizierung durchzuführen. Heutzutage ist jeder im Besitz eins Smartphones und kann problemlos Apps herunterladen die ein zeitbegrenztes OTP generieren. Mit diesen kann man sich (für eine bestimmte Zeit) authentifizieren. Ein Angreifer müsste sich, um einen erfolgreichen Angriff durchzuführen, in dem Augenblick indem eine Anfrage gestellt wird, auf dem Rechner des Nutzers befinden bzw. zu seinem Passwort gleichzeitig noch das Smartphone (oder Vergleichbares) gekapert haben, um die Zahl / den Schlüssel auszulesen und sich innerhalb dieser Zeit statt des Nutzers einzuloggen. Auf vielen Mobilgeräten würde diese Auslese schon alleine am Rechtesystem und der Rechteverwaltung für mobile Applikationen scheitern. Ein Hauptproblem teilen sich allerdings sowohl neuartige als auch ältere Ansätze zur Authentifizierung: Die Identifikation findet nicht immer oder nur teilweise statt. Ein Server kann also anhand eines Passwortes validieren, dass der Eingeber der Besitzer dieses Passwortes oder des privaten Schlüssels ist. Allerdings kann er bei beiden Ansätzen keine Garantie geben, dass dies auch der ursprüngliche Ersteller des Accounts war und somit die Berechtigung auf diese Webseite wirklich besitzt. Davon wird allerdings zunächst einmal aufgrund der Authentifikation ausgegangen. Wenn es um die Privatsphäre geht, muss man natürlich bei der Eingabe von biometrischen Daten der Empfängerseite (dem Server) vollends vertrauen hinsichtlich sorgfältigen und gewissenhaften Umganges mit den Daten. Aufgrund der hohen Bequemlichkeit und Sicherheit dieser Methoden allerdings, verzichtet man bewusst auf einen Teil der Privatsphäre. Am Ende funktionieren diese Konzepte immer so, dass eines der drei Kriterien nicht vollständig oder in Perfektion erfüllt werden kann. Es herrscht das "pick two of them Prinzip, welches man aus vielen Teilbereichen der Informatik bereits kennt. Somit können neue Authentifizierungsverfahren einiges mehr an Sicherheit (durch Besitz) und Bequemlichkeit bieten, fordern allerdings einen entscheidenden Teil der Privatsphäre.

5 Quellen und Inhaltsverzeichnis

Literatur

[1] Alexei Czeskis (Google) Dirk Balfanz (Google). *Web Authentication: An API for accessing Public Key Credentials Level 1*. 4. März 2019. URL: https://www.w3.org/TR/webauthn/ (besucht am 28. 05. 2020).

[2] Christian Forst. „Sichere Authentifizierung - Teil 1: Klassische Methoden“. In: *Conplore Redaktion* (14. Mai 2014).

[3] David M'Raihi Google. *Time and event based one time password*. 2006. URL: https://patents.google.com/patent/US9258124B2/en (besucht am 29. 05. 2020).

[4] ibau. *Passwort, Erklärung zu Passwort und Anwendung, Sicherheitslücken bei Passwörtern*. Kein Datum. URL: https://www.ibau.de/akademie/glossar/passwort/l (besucht am 21. 05. 2020).

[5] Colin Steele Margaret Rouse. „time-based one-time password (TOTP)“. In: *Searchsecurity* (Juli 2019). URL: https://searchsecurity.techtarget.com/definition/time-based-one-time-password-TOTP (besucht am 28. 05. 2020).

[6] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz-Kompendium*. 2018. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2018.pdf?__blob=publicationFile&v=7 (besucht am 24. 05. 2020).

[7] Bundesamt für Sicherheit in der Informationstechnik. *Zwei-Faktor-Authentisierung für höhere Sicherheit*. Kein Datum. URL: https://www.bsi-fuer-buerger.de/BSIFB/DE/DigitaleGesellschaft/OnlineBanking/Zwei_Faktor_Authentisierung/Zwei-Faktor-Authentisierung_node.html (besucht am 21. 05. 2020).