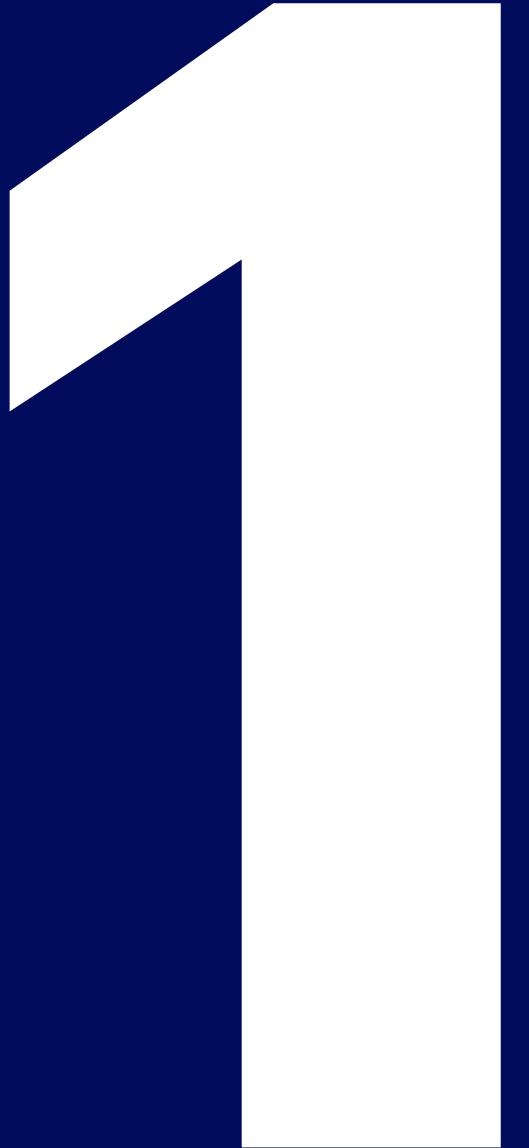
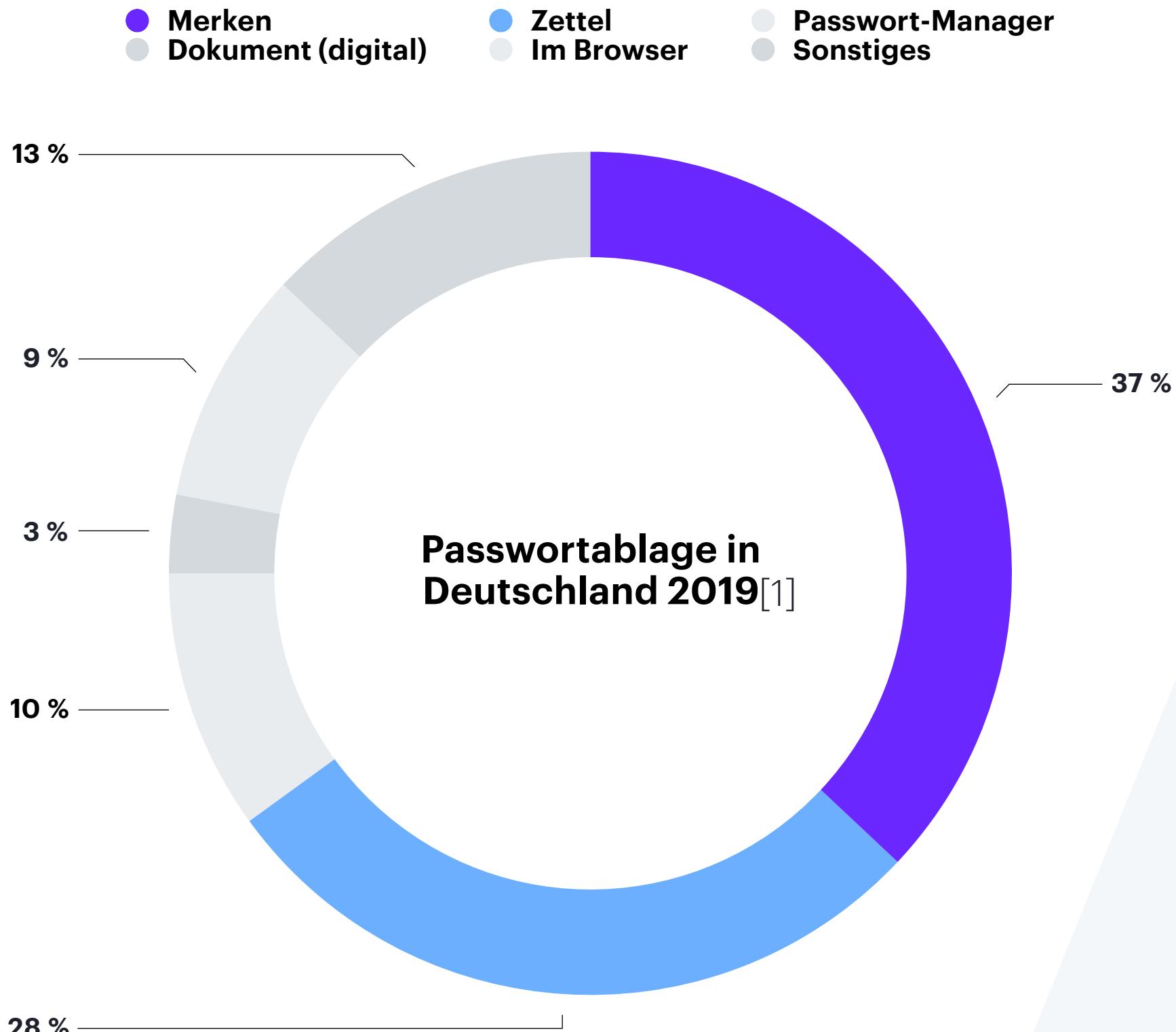


# Sicherheitsevaluation von Multi-Faktor Authentifizierung im Vergleich zur Web Authentifizierungs API

# Motivation

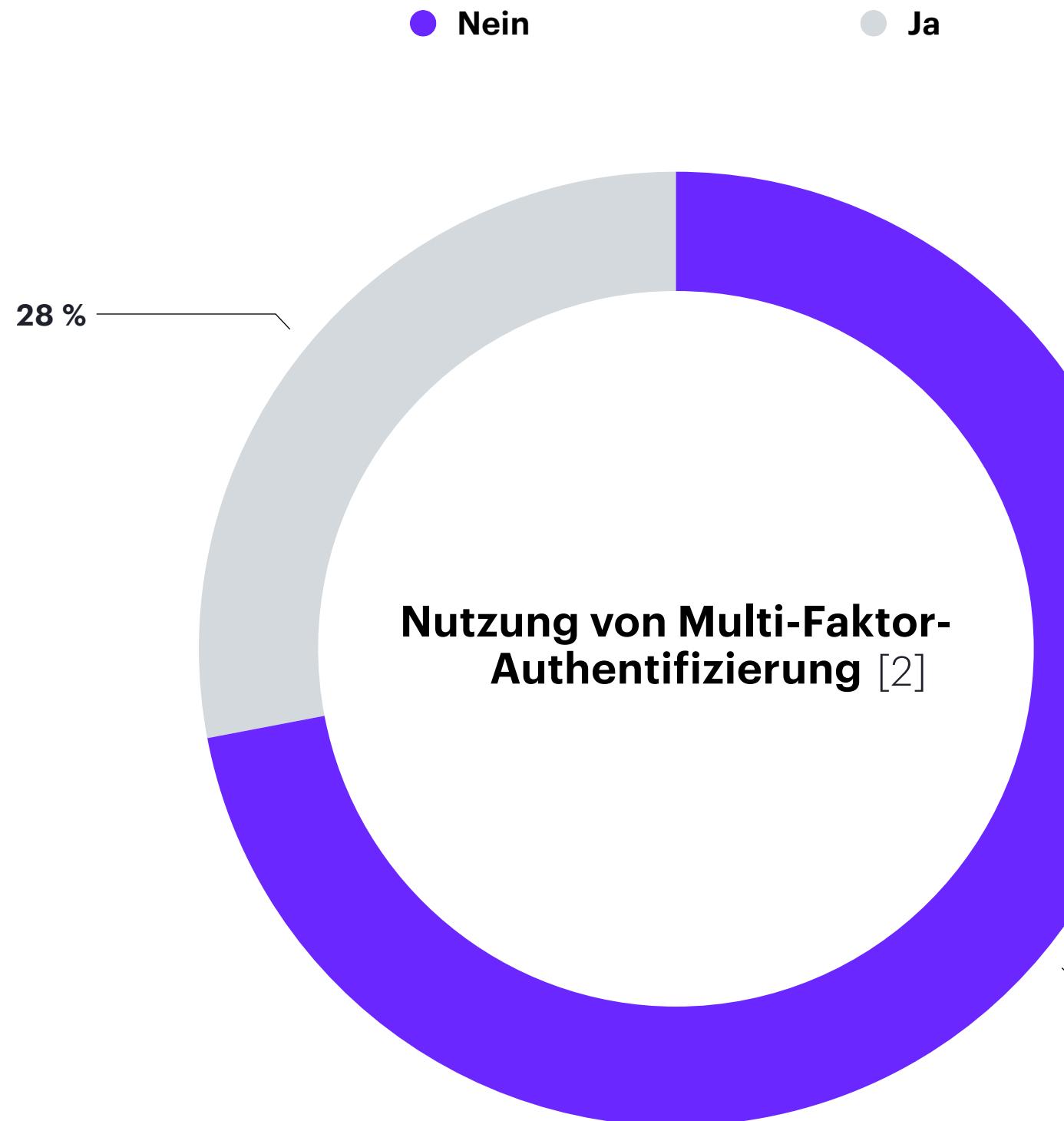


# Passwortablage in Deutschland 2019



> 65% nutzen keinen Passwort-Manager;  
aber Passwörter werden wiederverwendet  
oder beinhalten persönliche Daten

# Nutzung von Multi-Faktor Authentifizierung in den USA 2017



> 70% nutzen keine Multi-Faktor Authentifizierung!

# Anzahl der gestohlenen Zugangsdaten in einem Jahr

**1 900 000 000**<sup>[3]</sup>

# “Ein”-Faktor Authentifizierung



# Methoden der Authentifizierung

## Wissen

- > Geheimnis, welches nur der Nutzer kennt
- > Beispiele
  - > Passwörter, PINs
  - > Sicherheitsfragen

## Besitz

- > Ein Besitztum, welches nur der Nutzer hat
- > Hard- oder Software
- > Beispiele:
  - > Chipkarte
  - > (Sicherheits)-schlüssel
  - > App

## Biometrie

- > Körperliches Merkmal
- > Beispiele
  - > Gesicht, Iris, Ohr
  - > Fingerabdruck, ...

## Weitere Methoden

- > ortsbasiert
- > zeitbasiert
- > verhaltensbasiert
- > soziale Authentifizierung

# Sicherheitsaspekte



# Gefährdungen unabhängig von der Authentifizierungsmethode

## Initialisierung

- > Bereits infizierte Geräte können Geheimnisse abhören und weiterleiten
- > Sicherheitskameras, Kollegen oder Webcams können die Daten ebenfalls erlangen

## Transport

- > Abhören der Kommunikation (bspw. HTTP Verkehr)
- > Manipulation der Schnittstellen (USB Anschluss) oder Sensoren

# Wissensbasierte Authentifizierung

- > Das Gehirn hat Schwierigkeiten sich unterschiedliche, sichere Passwörter für jede Webseite zu merken
- > Wiederverwenden desselben Passworts für mehrere Accounts
- > Nutzung von einfach erratbaren Passwörtern und Informationen, die zu Personen gehören
- > Aufschreiben/Verwahren der Passwörter in unsicherer Weise
  
- > Sicherheitsfragen können die Sicherheit verringern, wenn sie wahrheitsgemäß beantwortet werden
- > Erzwungener Passwortwechsel bringt keinen Sicherheitsgewinn
  
- > Unbekannt, ob der Service Provider die Passwörter hashed, salted oder peppered
- > Oft genügt ein Passwort-Hash, um weitere Angriffe durchzuführen

# Besitzbasierte Authentifizierung

- > Gefahr von Verlust, Diebstahl, Beschädigungen oder Vergessen
- > Bei Entwendung ebenfalls benutzbar durch andere Nutzer
- > Drahtlose Übertragungen sind unter Umständen abhörbar
- > Ersatz ist deutlich kostenintensiver als Wissen auszutauschen

# Authentifizierung durch Biometrie

- > Merkmale können sich über die Zeit verändern
- > (Temporäre) Nicht-Verfügbarkeit aufgrund von Verletzungen
- > Probleme der "intra-user Varianz", sowie der FRR und FAR
- > Große datenschutzrechtliche und sicherheitsrelevante Bedenken der Nutzer
- > Merkmale sind kopierbar; ein Ersetzen aber unmöglich

# Multi-Faktor Authentifizierung



# Einmalpasswörter

- > Event-basiert (HOTP) und zeit-basiert (TOTP), beide standardisiert in RFCs
- > Basieren auf "Message Authentication Codes"
- > Nutzung eines "shared secrets"
- > Generation entweder auf dem Client (z.B. mittels Apps) oder auf dem Server
  - > Versand der Passwörter via E-Mail oder SMS
- > Proprietäre Verfahren von RSA oder Yubico existieren ebenfalls

# Sicherheitsschlüssel

- > Können in Hardware oder als Software Lösung realisiert werden
- > RSA SecurID ist der bekannteste Vertreter
  - > Ebenfalls Generation eines OTPs
- > Universal Second Factor (U2F) als offener Standard
  - > Keine Spezifizierung durch das W3C o. Ä.; experimentelle API
  - > Sicherheitsschlüssel können via USB, BLE oder NFC kommunizieren
  - > Nutzung von Public-Key Authentifizierung und des Challenge-Response Verfahrens

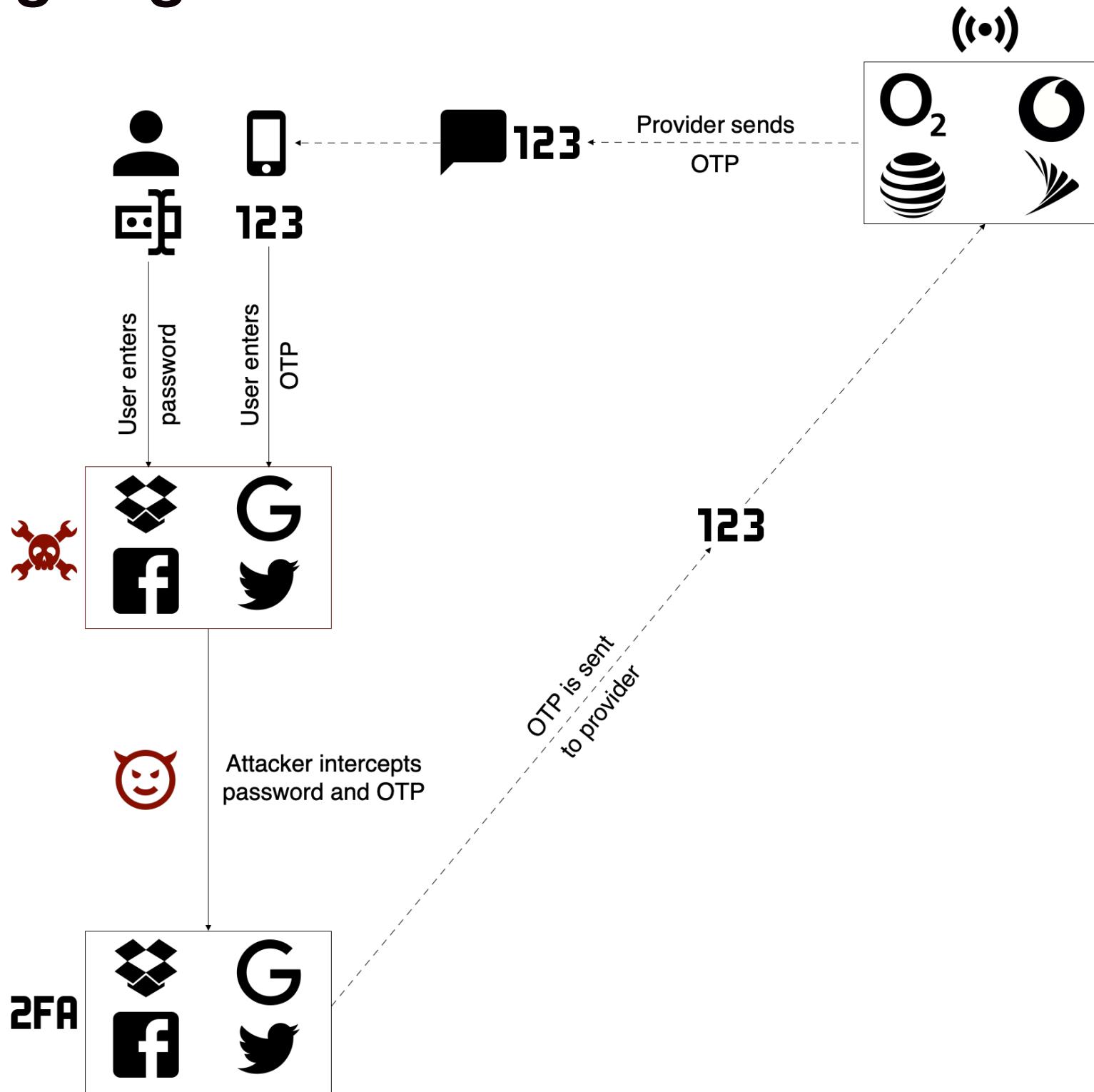
# Sicherheitsaspekte



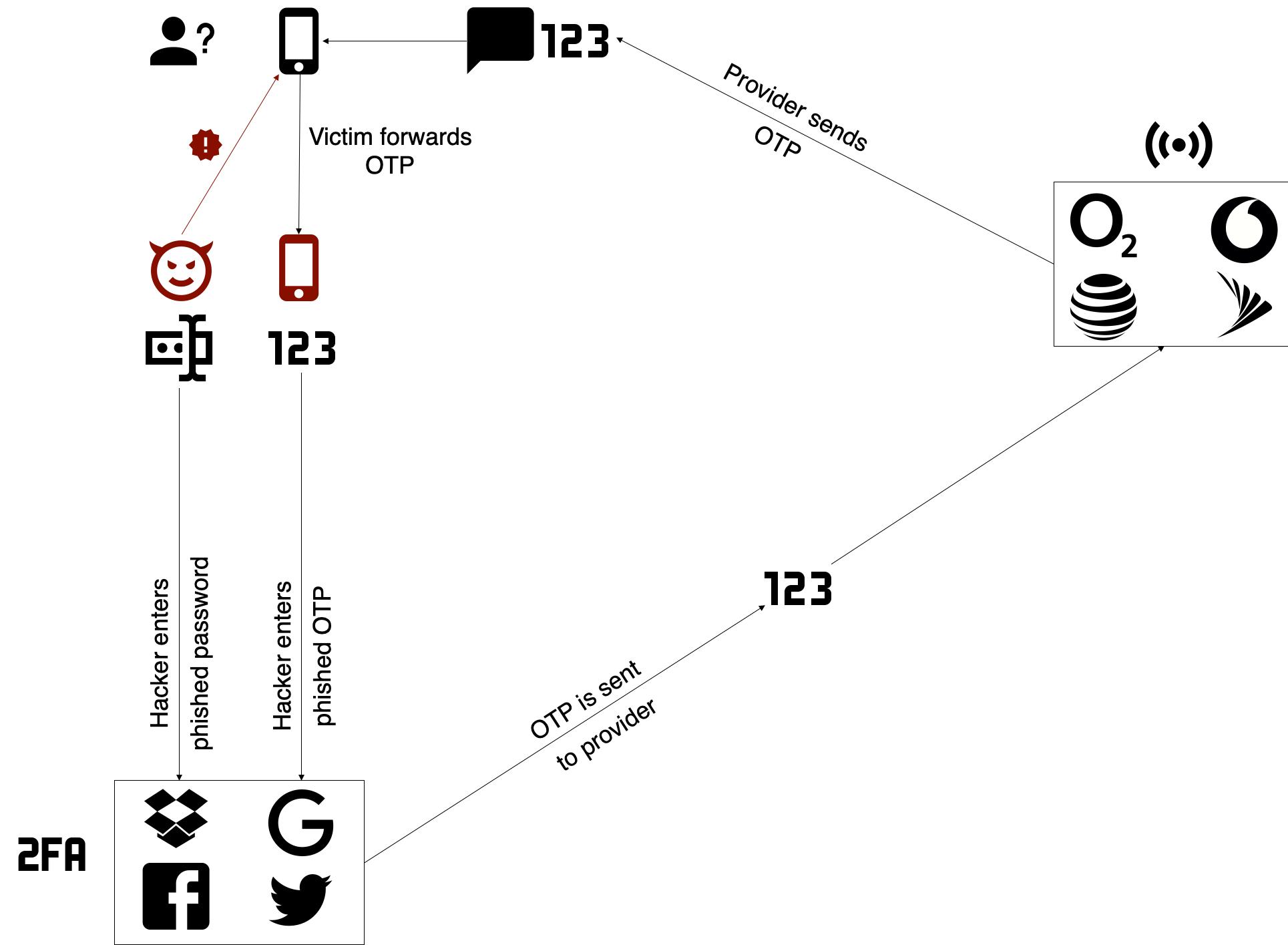
# Einmalpasswörter

- > Die zugrundeliegenden Algorithmen gelten als sicher
- > Kollisionen bei den Hash Funktionen von MD5 und SHA-1 stellen kein Sicherheitsrisiko dar
- > Ausnutzung falscher Konfigurationen
  - > Zu geringe Größe des gewählten Alphabets
  - > Fehlende Invalidierung der Passwörter
  - > Kein Drosseln der Anfragen (Brute-Force Attacken)
  - > Probleme mit der Zeitsynchronisation
  - > Zu großes “look-ahead” Zeitfenster; Erhöhung der Angriffsfläche
- > Fehlende Aufforderung des Faktors bei bspw. des Deaktivieren von MFA

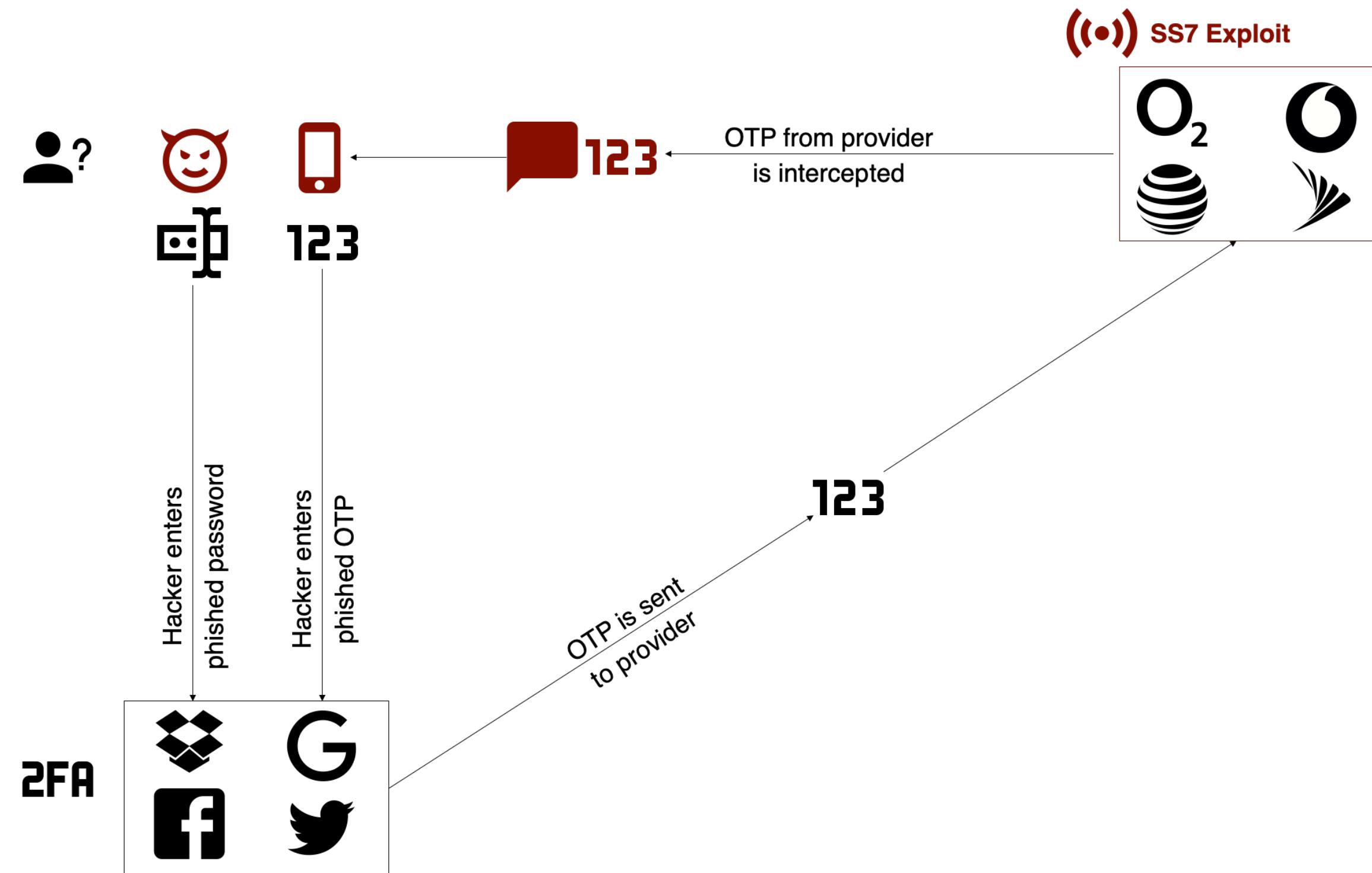
# Erfolgreicher Phishing Angriff trotz 2FA



# Verification Code Forwarding Attack



# SS7 Attack



# Sicherheitsschlüssel

- > Als Besitztum gelten die Bedrohungen von besitzbasierter Authentifizierung
- > RSA Hack in 2011 führte zum Austausch von 40 Millionen Schlüsseln, da der private Schlüssel gestohlen wurde
- > Oftmals keine Möglichkeit Firmware in den Schlüsseln zu aktualisieren
- > Anfälligkeit gegenüber Seitenkanalangriffen
- > Physische Angriffe auf die Speicherchips
- > Zentraler Metadatenserver von U2F ist ein sehr lukratives Ziel

# Web Authentication API

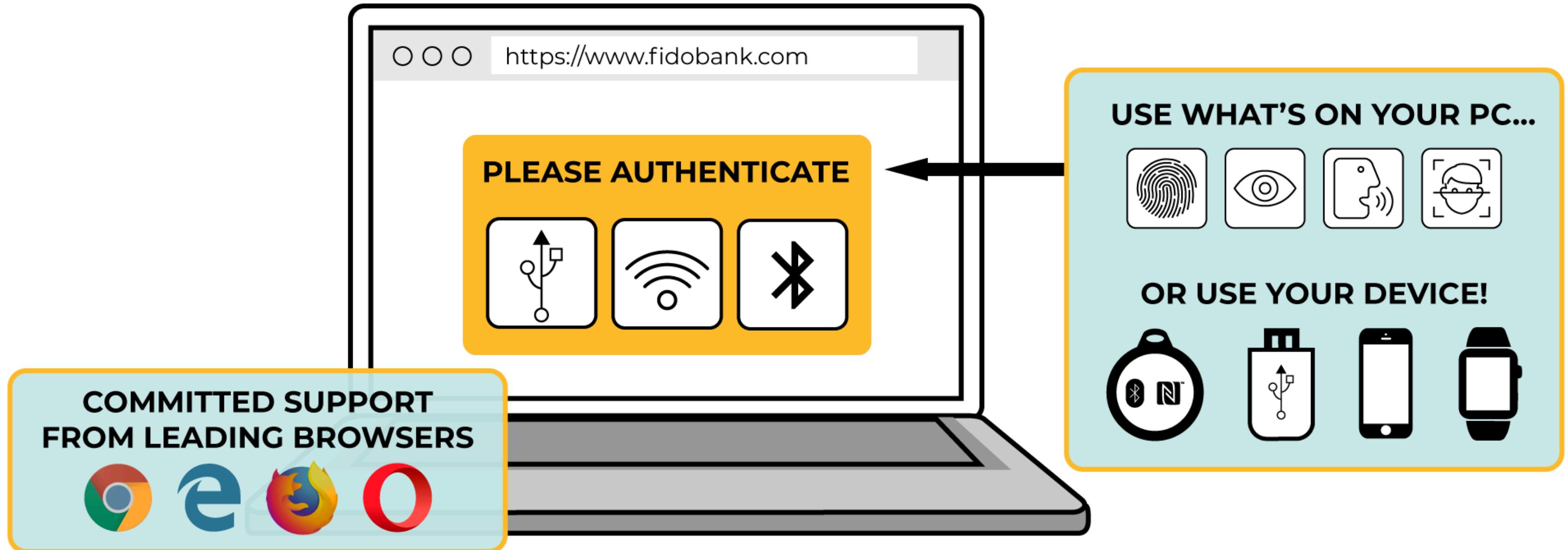
Web Authentifizierungs API; WebAuthn



# **“An API for accessing Public Key Credentials”<sup>[4]</sup>**

**“enabling the creation and use of strong, [...] public key-based credentials by web applications, for the purpose of strongly authenticating users.”<sup>[4]</sup>**

# Web Authentication API

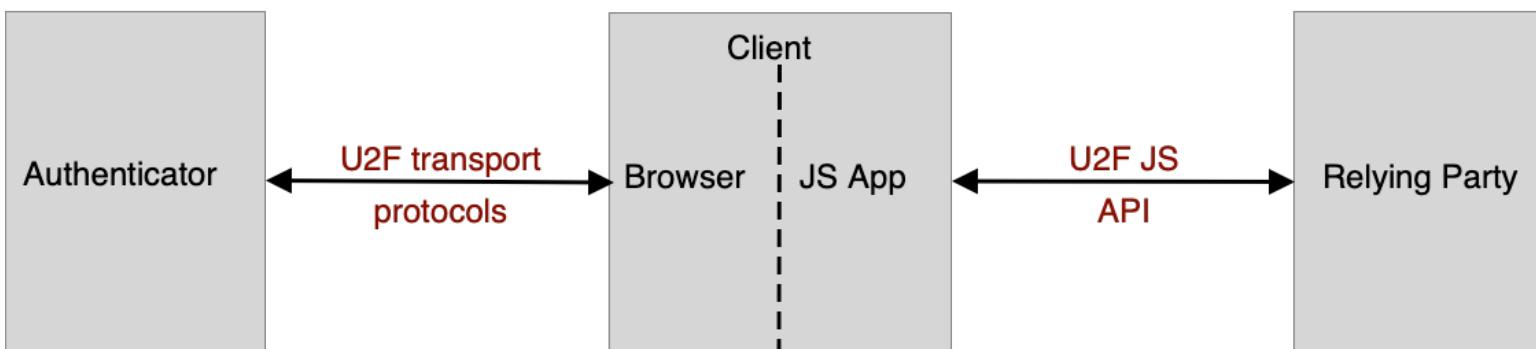


Bildquelle: <https://www.w3.org/2018/04/pressrelease-webauthn-fido2.html.en>; letzter Zugriff am 03.11.2019

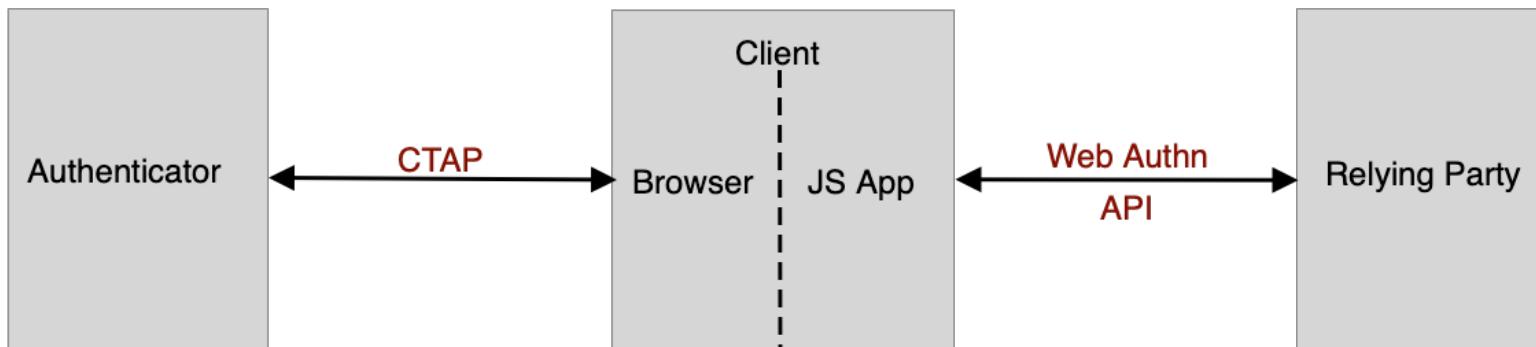
# Web Authentication API

- > Ermöglicht **passwortloses** Registrieren, Einloggen und Multi-Faktor Authentifizierung
- > Basiert auf Public-Key Authentifizierung
- > Teil des FIDO2 Projekts (CTAP & Web Authentication API); spezifiziert als Standard durch das W3C
- > Abwärtskompatibel zu U2F

## U2F



## FIDO2



- > Fast IDentity Online (FIDO)
- > Gründung 2013
- > Mitglieder (u. a.)
  - > PayPal, Google, Microsoft, Mastercard, VISA, Samsung, BSI
- > Spezifizierungen UAF, U2F, CTAP und des ersten Entwurfs der Web Authentication API als FIDO 2.0

# Web Authentication API

- > Authentifikatoren können rein in Software oder extern per USB, Bluetooth oder NFC realisiert werden
- > Unterstützt von Chrome, Edge, Firefox, Safari, Chrome & Firefox für Android
  
- > Keine Unterstützung durch den Internet Explorer; Problematisch für den Enterprise Bereich
- > Viele Android Browser ohne Unterstützung der API
- > Keine Unterstützung von iOS Safari, nur Drittanbieter App “Brave Browser” besitzt Support für die API

# Sicherheitsaspekte



# Sicherheitsaspekte und Nutzbarkeit

## > Keine bekannten, erfolgreichen Attacken

- > Formell verifiziert
- > Sicherheitsprobleme sind nur auf Protokollebene vorhanden
  - > Fehlende Spezifizierung der Nutzung eines sicheren Zufallszahlengenerators
  - > Unterstützung von RSASSA PKCS#1 v1.5 (=> Bleichenbacher Angriff)
  - > Keine Punkt-Komprimierung auf der elliptischen Kurve gefordert
  - > "Schlechte" Auswahl der elliptischen Kurven (Barreto-Naehring) mit reduzierter Anzahl von Bits
- > Keine Möglichkeit von Backups des Schlüsselmaterials
  - > Es sollten daher immer mindestens zwei Schlüssel mit einem Account assoziiert werden

# Vergleich zu anderen MFA Lösungen

- > Erweiterung des U2F Protokolls, d.h. besitzbasierte Sicherheitsbedrohungen treffen ebenfalls zu
  - > Durch Spezifizierung durch das W3C besteht eine erhöhte Unterstützung der Webbrowser
- > Im Vergleich zu z. B. TOTP fehlt in manchen Webbrowsers die Unterstützung und Interoperabilität
- > WebAuthn ist resistent gegenüber Phishing, viele andere MFA Lösungen, bis auf U2F, nicht.
- > WebAuthn bietet keine Backup Möglichkeiten
- > Keine Lösung bietet Schutz vor bspw. Session Hijacking

# Zusammenfassung und Fazit



# Zusammenfassung und Fazit

## Generell gilt

- > Sensibilisierung der Nutzer gegenüber Gefahren
- > Bewusstsein für sicheren Umgang mit Zugangsdaten schaffen

## Bestehende MFA Lösungen

- > MFA kann die Sicherheit erhöhen, ist bis auf U2F aber nicht resistent gegenüber Phishing
- > SMS Verkehr ist abhörbar und unsicher
- > Probleme mit Passwörtern bestehen weiterhin

## Web Authentifizierungs API

- > Kann Passwörter komplett durch Public-Key Authentifizierung ersetzen
- > Bietet MFA oder passwortloses Registrieren und Login durch Public-Key Authentifizierung an
- > Aktuell nicht ausreichend unterstützt, vor allem in mobilen Betriebssystemen und Webbrowsersn
- > Wenige Webseiten haben die Web Authentifizierungs API implementiert

# Quellen

1. Christian Friemel. Trotz „Collection #1-5“: Beim Passwortschutz lernen deutsche Internet-Nutzer nur langsam dazu. Mar. 27, 2019. URL: <https://newsroom.web.de/2019/03/27/trotz-collection-1-5-beim-passwortschutz-lernen-deutsche-internet-nutzer-nur-langsam-dazu> (letzter Zugriff am 03.11.2019).
2. Olabode Anise und Kyle Lady. State of the Auth: Experiences and Perceptions of Multi-Factor Authentication. Nov. 7, 2019: URL: <https://duo.com/blog/state-of-the-auth-experiences-and-perceptions-of-multi-factor-authentication> (letzter Zugriff am 03.11.2019).
3. Kurt Thomas et al. “Protecting accounts from credential stuffing with password breach alerting.” In: 28th USENIX Security Symposium. USENIX Security ’19. Santa Clara, CA, USA: USENIX Association, Aug. 2019, pp. 1556–1571. ISBN: 978-1-939133-06-9.
4. Dirk Balfanz et al. Web Authentication: An API for accessing Public Key Credentials Level 1. Mar. 4, 2019. URL: <https://www.w3.org/TR/2019/REC-webauthn-1-20190304> (letzter Zugriff am 03.11.2019).