Passwortlose Logins mit PassKeys

Stefan Schumacher

Chemnitzer Linux-Tage 2025

$Id: CLT2025-Passkeys.tex,v 1.11 2025/03/23 11:42:44 stefan Exp $

Über mich

- CCC2004: Einführung in die Kryptographie
- CLT2008: Sichere Passwörter
- 2013: Stratfor-Passwörter cracken
- CLT2016: Zwei-Faktor-Authentifizierung mit Yubikeys
- CLT2024: Sichere Datenhaltung und Backup in der Cloud

Inhaltsverzeichnis

1. Einführung
2. Standards
3. FIDO2
4. Passkeys
5. libfido2
6. Selfhosting

Passwortsicherheit

- schwache Passwörter, schlechte Passwortsicherheit, schlechte Hygiene
- Stratfor-Hack: Kundendatenbank gestohlen, Passwörter als MD5 gehasht
- https://haveibeenpwned.com/ Passwort-Leaks
- 2FA als »Lösung« mit unterschiedlichem Sicherheitsniveau

Betreiber-Probleme

- Benutzer vergessen Passwörter $\rightsquigarrow$ Konto freischalten bindet Ressourcen
- Benutzer vergessen Benutzername und/oder Domäne $\rightsquigarrow$ Wiederherstellung noch aufwändiger
- Wiederherstellung ist Sicherheitsrisiko (Ist Alice wirklich Alice?)
- Auch ohne Sicherheitsproblematik: Passwörter sind für Betreiber teuer und unpraktisch
- Passwörter abschaffen!
- FIDO2: Authentifikationsmöglichkeit *ohne* Benutzername/Passwort

Inhaltsverzeichnis

1 Einführung

2 Standards

3 FIDO2

4 Passkeys

5 libfido2

6 Selfhosting

FIDO Alliance

Fast Identity Online

- Industrie-Konsortium mit 200 Mitgliedern, u.a. BSI, Infineon, Google, Paypal, Lenovo, Alibaba, NTT DoCoMo, Samsung, Visa, RSA, Intel, ING, Yubico
- Spezifikationsrahmen u.a. für Biometrie, TPM2, Smart Cards, NFC
- Authentifikation mittels asymmetrischer Kryptographie
  Es sollen keine geheimen Daten mehr über unsichere Kanäle laufen
- UAF und U2F $\rightsquigarrow$ Client to Authenticator Protocol (CTAP1)
  Benutzername + Passwort + Token
- Kooperation mit W3C: WebAuthn und FIDO2 (CTAP2)
  möglich: nur noch Passkey, *ohne* Benutzername und Passwort

Exkurs: Zahlungsdienstleister

EU-Recht, EWR+UK

- Richtlinie (EU) 2015/2366 (PSD2)
- Strong Customer Authentication (SCA)
- verlangt 2FA für alle Transaktionen
- *Für Remote-Transaktionen umfasst die Starke Kundenauthentifizierung Elemente, die den Zahlungsvorgang dynamisch mit einem bestimmten Betrag und einem bestimmten Zahlungsempfänger verknüpfen.*
- mit Passkeys so (noch) nicht möglich

Inhaltsverzeichnis

1 Einführung

2 Standards

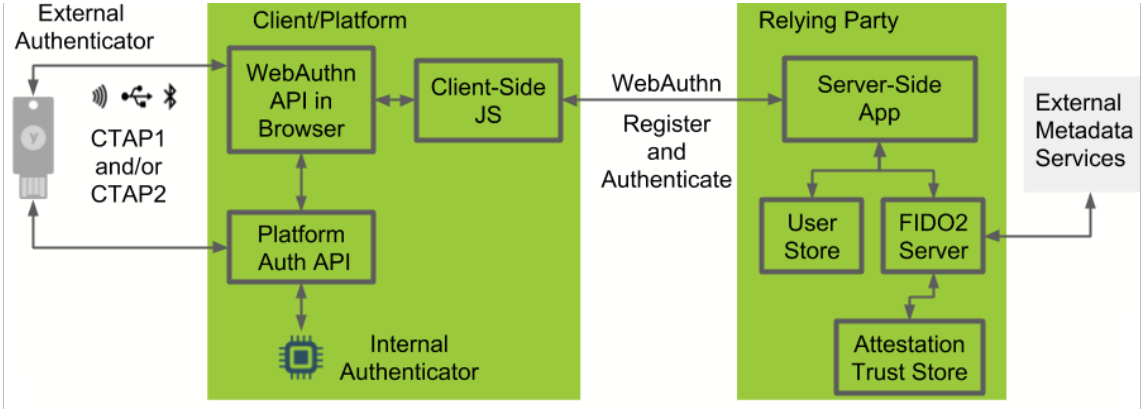
3 FIDO2

4 Passkeys

5 libfido2

6 Selfhosting

Application Layer Architecture



https://developers.yubico.com/WebAuthn/WebAuthn_Developer_Guide/

FIDO2

Registrierung eines Passkeys mit Discoverable Credentials

1. Passkey wird bei der *Relying Party* angemeldet
2. Challenge-Response-Verfahren
   - Zufallsdaten
   - Informationen zur Domain der RP
     Passkey ist an die Domain gebunden!
3. Benutzerinteraktion (*Anwesenheit* + Wissen)
4. Öffentliche Schlüssel werden ausgetauscht
   `secp256k1` ECC $\rightsquigarrow$ Sicherheitsniveau 120 Bit
   es gehen keine privaten Schlüssel übers Netz
   - Passkey speichert öffentlichen Schlüssel der RP
   - RP speichert öffentlichen Schlüssel des Passkey

FIDO2

Phishing-Resistenz

- Passkey authentifiziert sich am RP Server
- RP Server authentifiziert sich am Passkey
- per asymmetrischer Kryptographie
- Phishing-Resistenz *massiv* erhöht
  Passkey funktioniert nur am registrierten System
- Caveat:
  - Registrierung muss korrekt ablaufen
  - Domain-Hijacking
  - Sicherheit der verwendeten Schlüssel (Heartbleed!)
    Hardware Security Modul / Trust Store / TPM

FIDO2

Phishing-Resistenz

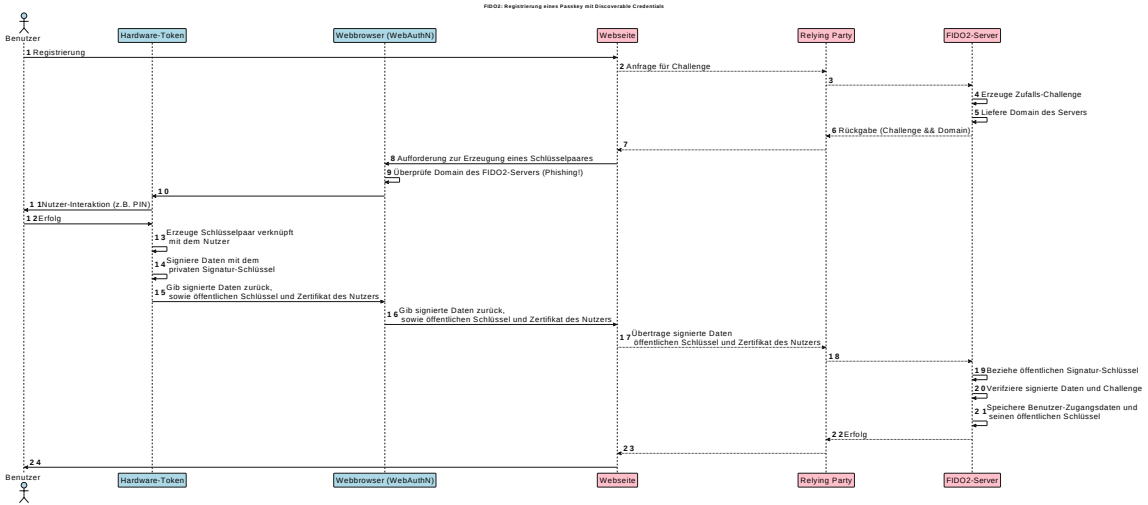
- Passkey authentifiziert sich am RP Server
- RP Server authentifiziert sich am Passkey
- per asymmetrischer Kryptographie
- Phishing-Resistenz *massiv* erhöht
  Passkey funktioniert nur am registrierten System
- Caveat:
  - Registrierung muss korrekt ablaufen
  - Domain-Hijacking
  - Sicherheit der verwendeten Schlüssel (Heartbleed!)
    Hardware Security Modul / Trust Store / TPM

Passwortloses Login

- Discoverable Credentials oder Resident Key genannt
- Relying Party kann einen Passkey einem Konto zuordnen
- RP verknüpft bei Registrierung Passkey mit Konto (per UUID)
- Passkey erzeugt privaten Schlüssel in seiner Hardware und verknüpft ihn mit der Konto-UUID und RPID
- Anzahl der RK-Slots begrenzt!
- mit PIN und/oder Biometrie: 2FA ohne Passwort möglich

Beglaubigte Passkeys

- Attestation: Aufbau einer kryptographischen Zertifikatskette vom Token zum Hersteller in Hardware
- RP kann Zertifikatskette mit kryptographischen Methoden prüfen um Token zu vertrauen
- Interessant für Unternehmen mit hohen Sicherheitsanforderungen
- u.U. komplexere Logistik


FIDO2: Registrierung eines Passkey mit Discoverable Credentials
Benutzer
Hardware-Token
Webbrowser (WebAuthN)
Webseite
Relying Party
FIDO2-Server
1 Registrierung
2 Anfrage für Challenge
3
4 Erzeuge Zufalls-Challenge
5 Liefere Domain des Servers
6 Rückgabe (Challenge && Domain)
7
8 Aufforderung zur Erzeugung eines Schlüsselpaares
9 Überprüfe Domain des FIDO2-Servers (Phishing!)
10
11 Nutzer-Interaktion (z.B. PIN)
12 Erfolg
13 Erzeuge Schlüsselpaar verknüpft mit dem Nutzer
14 Signiere Daten mit dem privaten Signatur-Schlüssel
15 Gib signierte Daten zurück, sowie öffentlichen Schlüssel und Zertifikat des Nutzers
16 Gib signierte Daten zurück, sowie öffentlichen Schlüssel und Zertifikat des Nutzers
17 Übertrage signierte Daten öffentlichen Schlüssel und Zertifikat des Nutzers
18
19 Beziehe öffentlichen Signatur-Schlüssel
20 Verifiziere signierte Daten und Challenge
21 Speichere Benutzer-Zugangsdaten und seinen öffentlichen Schlüssel
22 Erfolg
23
24
Benutzer
Hardware-Token
Webbrowser (WebAuthN)
Webseite
Relying Party
FIDO2-Server

Inhaltsverzeichnis

1 Einführung

2 Standards

3 FIDO2

4 Passkeys

5 libfido2

6 Selfhosting

Passkey

als Datei

- Passkey kann als Datei abgelegt werden
- und normal im Dateisystem liegen (unsicher!)
- Import in KeePassXC ab 2.7.7 / KeePassium 2.0 möglich
- Schutz der Passkeys durch Zugangschutz der KeePassXC-Datenbank
- Sicherung der KeePassXC-Datenbank normal möglich, z.B. in Cloud

Passkey

im mobilen Endgerät

- Datei wird im Handy/Tablet/Laptop (TPM2 bzw. Secure Enclave) abgelegt
- mit den User-Credentials verschlüsselt (Windows-Hello, Google-Konto oder Apple ID)
- per Biometrie gesichert (Face-ID oder Fingerabdruck)
- und in die Hersteller-Cloud (Apple Cloud/Google-/Microsoft-Konto) gesichert
- Apple/Google/Microsoft kümmern sich um Backup
- Und haben die Herrschaft über die Passkeys!
- Verlust von AppleID/Microsoft-/Google-Konto $\rightsquigarrow$ Verlust der Zugänge?

Passkey

im mobilen Endgerät

- Apple iOS: kann KeePassium zum speichern/abrufen von Passkeys nutzen
- Keepass-Datenbank kann unter meiner Kontrolle bleiben
- Trotzdem Backup/Restore nötig!
- KeePassXC kann Passkeys als Datei importieren
- 14.10.2024: Working Draft on Secure Credential Exchange Format / Protocol

Passkey

als Hardware-Token

- Passkey in einem sicheren Hardware-Token
- Export aus dem Hardware-Token (fast) unmöglich $\rightsquigarrow$ kein Backup!
- Update der Firmware ebenfalls nicht möglich!
- Alle Operationen mit dem Passkey finden im Hardware-Token statt
- nur das Ergebnis wird nach außen kommuniziert
- Schutz durch PIN und/oder Fingerabdruck
- FIPS-Zertifizierung möglich (aber nicht wirklich ratsam für Normalverbraucher)
- Sicherste Form des Passkey!

THETIS
THETIS
CRYPTNOX
plug-up
Security
KEY

Hardware-Token

Kosten

- Yubikey Security Key: 30€, 100 Resident Keys
- Yubikey 5: 60€-90€, 100 Resident Keys
- Nitrokey Passkey: 30€, 50 Resident Keys
- Thetis A: 25€-35€, 50 oder 200 Resident Keys
- Token2.com Pin+ 25€, 300 Resident Keys
- Anzahl der speicherbaren Passkeys variiert
- Qualität der Hardware auch

Risiken

- Zugang zurücksetzen: Welchen Rücksetzkanal gibt es? Mail? SMS?
  Jeder Zugang ist nur so sicher wie der schwächste Zugang!
- Digitale Souveränität: Wer hat meine Credentials? Irgendein Non-EU-Unternehmen?
- Backup, Backup, Backup. Mehr als 1 Passkey notwendig.
- Passkey als Marketing-Buzzword!

Backup

ideal

- 3 Hardware-Passkey-Token
- an mindestens 2 verschiedenen Orten
- praktikabel?
- Für jede Kontoregistrierung zum Bankschließfach?

Backup
praktikabler

- 2 Hardware-Passkey-Token zu Hause
- falls ein HW-Key ausfällt
- Passkey-Datei im KeePassXC-Passwortmanager
- Passwort-Datenbank auf verschiedenen Systemen in der Cloud synchronisieren
  Rclone, CryFS, GocryptFS ...
- Die Cloud-Zugänge nicht vergessen ...
- Emergency-Sheet drucken und wegschließen

Sicherheitsempfehlung 2FA

1. Passkey als Hardware-Token
2. Passkey als Datei in einem Passwortmanager
3. einzigartiges, zufallsgeneriertes Passwort in einem Passwortmanager && TOTP (auf einem anderen Gerät)
4. alles andere, wie Passwort + Mail-TAN

Inhaltsverzeichnis

1. Einführung
2. Standards
3. FIDO2
4. Passkeys
5. libfido2
6. Selfhosting

libfido2

```
stefan@X201:\> lsusb | grep -i key
Bus 001 Device 014: ID 1050:0116 Yubico.com Yubikey NEO(-N) OTP+U2F+CCID
Bus 001 Device 015: ID 1ea8:f825 Thetis Security Key(F825)
Bus 001 Device 016: ID 349e:0026 TOKEN2 FIDO2 Security Key(0026)
Bus 001 Device 017: ID 1050:0407 Yubico.com Yubikey 4/5 OTP+U2F+CCID

stefan@X201:\> fido2-token -L
/dev/hidraw9: vendor=0x1050, product=0x0116 (Yubico Yubikey NEO OTP+U2F+CCID)
/dev/hidraw10: vendor=0x1ea8, product=0xf825 (Thetis Security Key(F825))
/dev/hidraw12: vendor=0x349e, product=0x0026 (TOKEN2 FIDO2 Security Key(0026))
/dev/hidraw14: vendor=0x1050, product=0x0407 (Yubico YubiKey OTP+FIDO+CCID)
```

libfido2

```
stefan@X201:\> fido2-token -L -r  /dev/hidraw2
Enter PIN for /dev/hidraw2:
00: dKbqkhPJna890siSSsyDPQCYqlMGpUKA5fyklC2CEHvA= webauthn.io
01: z1aIHBB85WV+lBfX+eMEfzSDMu42KCYWF/EeuErzPwAQ= pocketident.local.lan
02: paQIHBB85WV++eMEfzSKCYWzPwXMk05cdgEF3rHUDzcQ= sso.arbeitsagentur.de
03: cKMSUq5cekOuOV/BTx7lBfXzcaQ3R1n8ggVu6kPSJ0t8= google.com
04: MsFZhlkcdnSX/RF8JPXzi7XoGXDMu42c4YgcTnSF6ugh= amazon.de
05: e7P7yR6i8tF/EeuErYfXdf56mmwvM41O8dgXvCG4qgg8= apple.com
07: U28gMTYuIE3DpHIgMTI6MDY6NTguQ0VUIDIwMjUKqDge= github.com
08: e7P7yR6i8tF/EeuErYfXmYWZiNDNjNWNiY2Y1ZCAgLQo= gitlab.com

stefan@X201:\> fido2-token -I -c  /dev/hidraw2
Enter PIN for /dev/hidraw2:
existing rk(s): 26
remaining rk(s): 274
```

Inhaltsverzeichnis

1 Einführung

2 Standards

3 FIDO2

4 Passkeys

5 libfido2

6 Selfhosting

Passkeys nutzen

- `https://webauthn.io/` Passkey testen und Bibliotheken
- `chrome://settings/securityKeys`: Passkey verwalten
- LUKS2 FDE mittels Passkey (systemd) (Grub kann nur LUKS1!)
- OpenSSH ab 8.3 (mit libfido2-Support)
- GnuPG ab 2.2
- WebApps: Native Unterstützung oder
- Anbindung via OpenIDConnect mittels OIDC-Provider
- pocket-id.org: sehr simpel, unterstützt *nur* Passkeys
- Authentik, Authelia, Kanidm, Keycloak

Passkey für Immich

- Nginx PM: TLS und Reverse Proxy
- PocketID: Passkey-Login
- OIDC-Protokoll: Login-Vermittlung
- Immich: Fotos-Hosting

Fragen?

- stefan.schumacher2 <at> dataport.de
- https://mastodon.social/@0xKaishakunin
- @schumaste:matrix.org