authenticate 2023
THE FIDO CONFERENCE

Tips for Painless Passkeys

Matthew Miller
Technical Lead
Duo Security @ Cisco

Who am I?

- WebAuthn SME with an eye on the Relying Party's experience
- Author of SimpleWebAuthn and py_webauthn libraries, and current maintainer of webauthn.io
- Help drive FIDO2 adoption within the FIDO Alliance TWG, and W3C's WAWG and WACG

That's me!

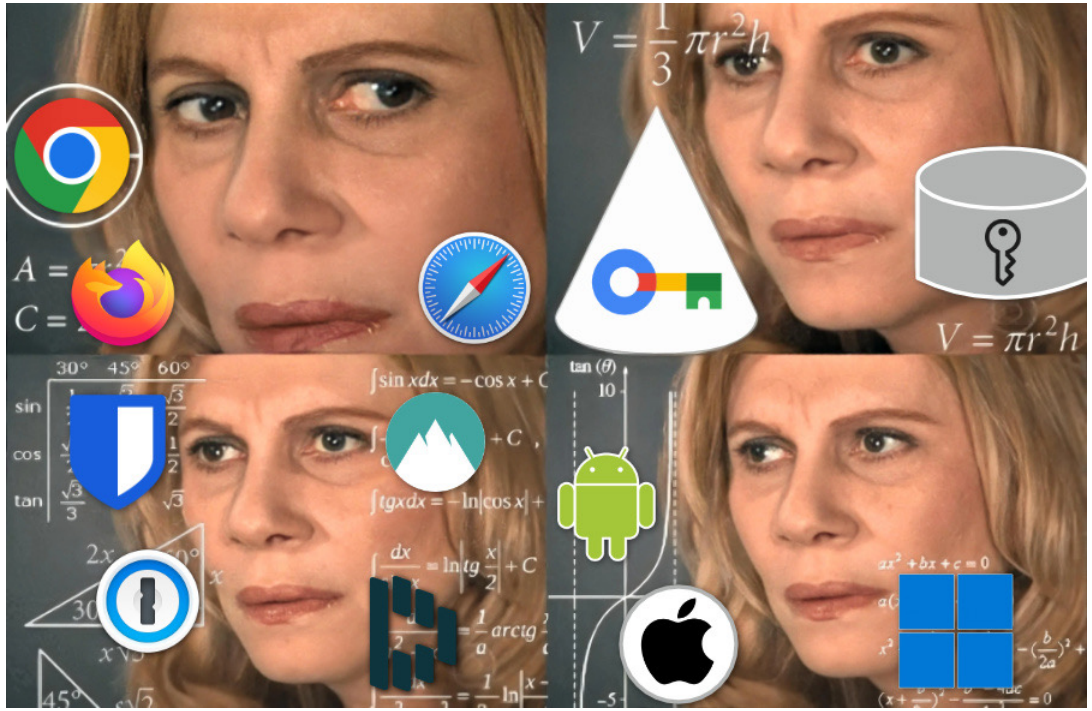
Passkeys are a bit of a double-edged sword

- **Synced passkeys** made WebAuthn available in more places, but also made it harder for Relying Parties (RP's) and users to remember the permutations that will lead to authentication success.
- Questions now shifting from "how can we use this" to **"how can we improve the user experience?"**
  - How can we help users understand **where their passkeys are stored**?
  - How can we follow through on **name change** requests?
  - How can we support **signing in to native apps**?


Passkeys are good
But they can be better

Nicknaming passkeys: Problem

- Browsers and platforms coordinate to offer multiple ways to **access passkeys across devices.**
- **Third-party providers** have entered the space to offer alternative cross-platform use of passkeys.
- How can RP's help users understand where their passkeys are stored?



Nicknaming passkeys: Solutions

From Good...

Look at multiple signals from registration:

- AAGUID
- Browser user agent
- Backup flags
- transports
- authenticatorAttachment

```
"1d2832da-..."
"Mozilla/5.0 (Macintosh..."
be:true, bs:true
["internal", "hybrid"]
"platform"
```

} "BitWarden"

...To Better

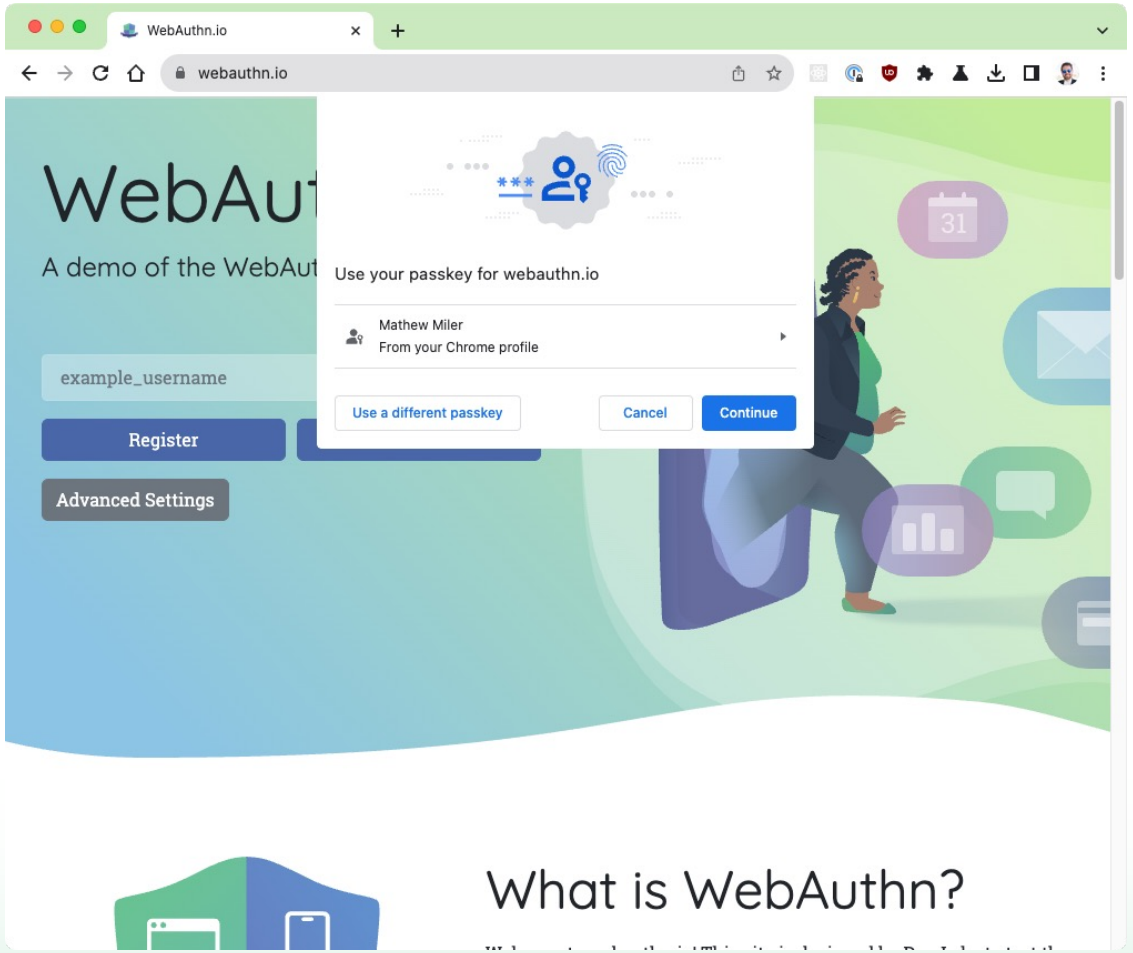
New **authenticatorDisplayName** value returned in the credProps extension:

```
{
  "credProps": {
    // Not actually a reality, but coming soon?
    "authenticatorDisplayName": "BitWarden - Miller Time Vault"
  }
}
```

https://github.com/w3c/webauthn/pull/1880

Updating passkey metadata: Problem

- **People often change** names, usernames, email addresses, phone numbers.
- A passkey's **user.name** and **user.displayName** are **decided during registration** and stored within authenticators as metadata.
- How can an RP help a user consistently see their updated name during passkey auth?



Updating passkey metadata: Solutions

From Good…

Technically feasible in a few steps:

- Prepare registration options with the new value for **user.name**, and same value for **user.id** and **rp.id**.
- Omit the existing credential from **excludeCredentials**.
- Call **navigator.credentials.create()**.
- Hope the user uses the same authenticator as before…

…To Better?

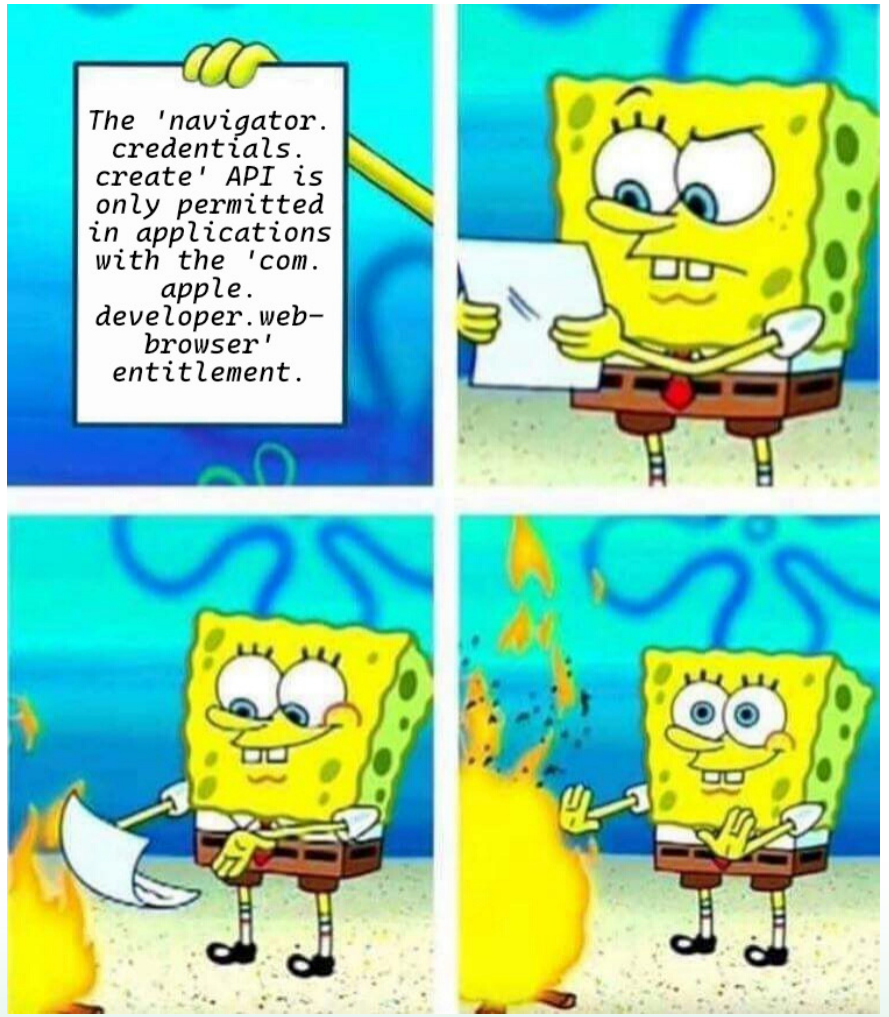
It turns out it's a tough problem to solve!

Also touches on the problem of how an RP that allows self-service passkeys administration can keep authenticators synced with deletion events.

Ongoing discussion in W3C WAWG: https://github.com/w3c/webauthn/issues/1967

Using passkeys in native apps: Problem

- Many existing native apps handle login in embedded webviews with **varying levels of WebAuthn support**.
- **WebAuthn API invocation errors** can be difficult to detect and build around.
- Sometimes webviews **lie**!
- How can an RP best incorporate passkeys auth into their native app?



Using passkeys in native apps: Solutions

From Good…?

- Support username + password + phishable 2FA? (*boooooo*)
- Open the default browser and handle authentication in an evergreen browser?
- 

…To Better

Implement recommended webviews:

- Apple: ASWebAuthenticationSession
- Google: Chrome Custom Tabs
- Microsoft: Edge WebView2

Or implement FIDO2 auth natively:

- Apple: Supporting passkeys
- Google: Credential Manager

Q & A

Where to find me

Mastodon: @iamkale@infosec.exchange

WACG: https://www.w3.org/community/webauthn-adoption/

Passkeys.dev: https://passkeys.dev

Libraries (GitHub):

- MasterKale/SimpleWebAuthn
- Duo-Labs/py_webauthn

authenticate 2023

THE FIDO CONFERENCE

Thank you.

Signature Sponsors:

1Password | Microsoft | yubico | Google

authenticatecon.com