WebAuthn

The future of user authentication on the web 🤝

Lucas Garron
CS 253 Guest Talk
2019-11-06
About Me

Lucas Garron
@lgarron

Mathematician, cuber, dancer, coder. I want the web to win. @GitHub websec, formerly @GoogleChrome usable security. Immigrant. He/him.

📍 Mountain View ⚫ garron.net
You may know me from:

Chrome DevTools Security
badssl.com, hstspreload.org
Speedcubing, Dancing
GitHub supports Web Authentication (WebAuthn) for security keys

GitHub now supports Web Authentication (WebAuthn) for security keys—the new standard for secure authentication on the web. Starting today, you can use security keys for two-factor authentication on GitHub with even more browsers and devices. And, since many browsers are actively working on WebAuthn features, we’re excited about the potential for strong and easy-to-use authentication options for the entire GitHub community in the future.

Register a new security key in your GitHub settings

More browsers, devices, and biometric options

Previously, GitHub supported physical security keys using the experimental U2F API for Chrome. WebAuthn is the standards-based successor. You can now use physical security keys on GitHub with:

- Windows, macOS, Linux, and Android: Firefox and Chrome-based browsers
- Windows: Edge
- macOS: Safari, currently in Technology Preview but coming soon to everyone
- iOS: Brave, using the new YubiKey 5Ci
A few words on Responsibility
Security and Privacy are not “add-on features”
Passwords (Redux)

“Use bcrypt”
Terribly phishable
HaveIBeenPwned.com
Factor

Something you ______.
Factor
Something you know.

Example:
Password
Factor

Something you have.

Example:

Security Key
Factor

Something you are.

Example: Fingerprint
Classical “Factors”
Stop thinking about factors
WebAuthn is supposed to help you...

Stop thinking about factors
WebAuthn
WebAuthn

A browser API for many authentication factors.
WebAuthn

navigator.credentials.create(...)
navigator.credentials.get(...)
Demo Time!
webauthn.io
webauthntest.azurewebsites.net
Try it yourself!

Windows Hello
Fingerprint (Android)
Touch ID (Chrome macOS)
Stop thinking about factors
A tour of factors
“We’ve emailed You a login link”.
Not a user auth factor.
Useless against “Meddler in the Middle” attacks
This is why you shouldn’t use texts for two-factor authentication

Researchers show how to hijack a text message

By Russell Brandom | Sep 18, 2017, 1:17pm EDT

Reddit Got Hacked Thanks to a Woefully Insecure Two-Factor Setup

The tech community has known about the risk of using SMS in two-factor authentication for years. Reddit appears to have missed the memo.

Why you are at risk if you use SMS for two-step verification

Do two-step verification the right way to keep hackers at bay.

Matt Elliott | July 31, 2017 4:27 PM PDT
TOTP
Time-based One-Time "Password"
HOTP
Hash-based
One-Time
“Password”

(no one uses this)
PAKE
Password Authenticated Key Exchange
(uncommon on the web)
Different security strengths
Client Certificates

Select a certificate
Select a certificate to authenticate yourself to client.badssl.com:443

BadSSL Client Certificate (BadSSL Client Root Certificate Authority)

Show Certificate  Cancel  OK
Push notifications
Something you... can do?

## The Doomsday Rule

<table>
<thead>
<tr>
<th>Weekday</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

### Doomsday Month

<table>
<thead>
<tr>
<th>January</th>
<th>February</th>
<th>March</th>
<th>April</th>
<th>May</th>
<th>June</th>
</tr>
</thead>
<tbody>
<tr>
<td>31/32*</td>
<td>28/29*</td>
<td>7</td>
<td>4</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>

\*Leap Year

<table>
<thead>
<tr>
<th>July</th>
<th>August</th>
<th>September</th>
<th>October</th>
<th>November</th>
<th>December</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>8</td>
<td>5</td>
<td>10</td>
<td>7</td>
<td>12</td>
</tr>
</tbody>
</table>

### Doomsday Century

<table>
<thead>
<tr>
<th>1500</th>
<th>1600</th>
<th>1700</th>
<th>1800</th>
</tr>
</thead>
<tbody>
<tr>
<td>1900</td>
<td>2000</td>
<td>2100</td>
<td>2200</td>
</tr>
<tr>
<td>2300</td>
<td>2400</td>
<td>2500</td>
<td>2600</td>
</tr>
<tr>
<td>3 (Wed)</td>
<td>2 (Tue)</td>
<td>0 (Sun)</td>
<td>5 (Fri)</td>
</tr>
</tbody>
</table>
Under the hood
Anybody: So what's the difference between #WebAuthn, CTAP2, FIDO2, and U2F?
Me: Behold the holy #FIDO2 trinity and be blessed🙏

Diagram:
- WebAuthn is not FIDO2
- FIDO2 is CTAP
- CTAP is not WebAuthn
- FIDO2 is U2F
- U2F is not WebAuthn
The experimental non-standard precursor API to WebAuthn. Still used.
CTAP2

Used by your browser/OS to communicate with security keys
FIDO2

≈ WebAuthn + CTAP2
Implementing WebAuthn
User-Facing Terminology

Two-factor authentication

Security key
When you are ready to authenticate, press the button below.

Use security key
User-Facing Terminology

For now: “security key”

Security key
When you are ready to authenticate, press the button below.

Use security key
User-Facing Terminology

In the future: “using your device”?
Configuration

User presence vs. user verification

Resident key vs. non-resident key

Platform vs. roaming
@github/webauthn-json

webauthn-json is a client-side Javascript library that serves as convenience wrapper for the the WebAuthn API by encoding binary data using base64url (also known as "websafe" or "urisafe" base64).

The WebAuthn API itself takes input and output values that look almost like JSON, except that binary data is represented as ArrayBuffers. Using webauthn-json allows the data to be sent from/to the server as normal JSON without client-side processing.

Usage

1. Replace calls to navigator.credentials.create() with create(), and navigator.credentials.get() with get().
2. Encode/decode binary values on the server as base64url.

Example

Install using:

```
npm install --save @github/webauthn-json
```
Account Recovery

A big unsolved problem.
WebAuthn: A Journey

Worth adopting, but there's a long way to go.