WebAuthn

The future of user authentication on the web 🌟

Lucas Garron
CS 253 Guest Talk
2021-11-11
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
WebAuthn at GitHub

Ben Toews (@mastahyeti) implemented U2F.

I wrote most of the WebAuthn implementation.
A few words on Responsibility
DROP ICE
Security and Privacy are not “add-on features”
Passwords (Redux)

“Use bcrypt”
Terribly phishable
HaveIBeenPwned.com
Authentication Factors
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.
navigator.credentials.create(...)
navigator.credentials.get(...)

WebAuthn
Demo Time!

webauthn.io

webauthntest.azurewebsites.net
Try it yourself!

Windows Hello
Fingerprint / PIN (Android)
Touch ID / Face ID (Apple)
Stop thinking about factors
A tour of factors
“We’ve emailed You a login link”.

Security Images

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
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktYmFzZTsgV2luYW1lLmNsYWlkIGFzaGluZw==

-----END OPENSSH PRIVATE KEY-----
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>Doomsday Month</th>
<th>January</th>
<th>February</th>
<th>March</th>
<th>April</th>
<th>May</th>
<th>June</th>
</tr>
</thead>
<tbody>
<tr>
<td>July</td>
<td>11</td>
<td>8</td>
<td>5</td>
<td>10</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>August</td>
<td>31/32^</td>
<td>28/29^</td>
<td>7</td>
<td>4</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>July</td>
<td>11</td>
<td>8</td>
<td>5</td>
<td>10</td>
<td>7</td>
<td>12</td>
</tr>
</tbody>
</table>

^Leap Year

**Doomsday Century**

<table>
<thead>
<tr>
<th>Century</th>
<th>1500</th>
<th>1600</th>
<th>1700</th>
<th>1800</th>
<th>1900</th>
<th>2000</th>
<th>2100</th>
<th>2200</th>
<th>2300</th>
<th>2400</th>
<th>2500</th>
<th>2600</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>5</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 🙏
The experimental non-standard precursor API to WebAuthn. Still used.
CTAP2

Used by your browser/OS to communicate with security keys
≈ WebAuthn + CTAP2
Implementing WebAuthn
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: “trusted device”? 

Add a trusted device

Sign in using built-in authentication (like Touch ID, Face ID or Windows Hello) instead of a password.

Register trusted device

Ask me later
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 WebAuth API by encoding binary data using `base64url` (also known as "websafe" or "urlsafe" base64).

The WebAuth API itself takes input and output values that look almost like JSON, except that binary data is represented as `ArrayBuffer`s. 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:

```bash
npm install --save @github/webauthn-json
```
User Flows

Registration
New device
Re-authentication
Recovery
Cross-browser PA? (e.g. Windows Hello)
AND
Existing registration for PA type?

Trusted device
Would you like to use this as a trusted device?
This will allow you to log in using both
a Windows Hello touch or fingerprint reader
and PA credentials.

Registered
on system

Registered
on local

Registered
on system

Registered
on local

Trust trusted device

Which button?
Register
Success?

Could be a duplicate registration?

Duplicate of security key?

Success?

Trusted device
Would you like to use this as a trusted device?
This will allow you to log in using both
a Windows Hello touch or fingerprint reader
and PA credentials.

Registered
on system

Registered
on local

Registered
on system

Registered
on local

Trust trusted device

Conflict trusted device

Get with current
Registration

Get with current
Registration

Success?

Trusted device
Would you like to use this as a trusted device?
This will allow you to log in using both
a Windows Hello touch or fingerprint reader
and PA credentials.

Registered
on system

Registered
on local

Registered
on system

Registered
on local

Trust trusted device

Success?

Trusted device
Would you like to use this as a trusted device?
This will allow you to log in using both
a Windows Hello touch or fingerprint reader
and PA credentials.

Registered
on system

Registered
on local

Registered
on system

Registered
on local

Trust trusted device

Success?

Trusted device
Would you like to use this as a trusted device?
This will allow you to log in using both
a Windows Hello touch or fingerprint reader
and PA credentials.

Registered
on system

Registered
on local

Registered
on system

Registered
on local

Trust trusted device

Success?

Trusted device
Would you like to use this as a trusted device?
This will allow you to log in using both
a Windows Hello touch or fingerprint reader
and PA credentials.

Registered
on system

Registered
on local

Registered
on system

Registered
on local

Trust trusted device

Success?
A potential solution

Cloud keychains?
Account Recovery

A big unsolved problem.
WebAuthn: A Journey

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