LOOKING

BEYOND

PERMISSION PROMPTS
Hi

* Serena
* UX designer
* Google Chrome
* Pronounces it “owned” not “poned”
example.com wants to
📍 Use your location

Block  Allow
1. THE POINT OF PROMPTS
In the beginning, the web was for documents.
Over time, we added more capabilities.
example.com wants to 🎤 Use your microphone

Block  Allow

The permission prompt is to make sure you actually intend to let the website use a new capability.
Permissions are for detecting user intent
Q: Did you have a nice experience?

A: no
example.com wants to use your location. Block or allow?
The current way we handle permissions does not scale.
2. MO’ PROMPTS
   MO’ PROBLEMS
serena’s list of gripes 😞

1. Prompts are interruptive
   - Timing is controlled by the site, not by the user
   - Often appears immediately on page load
   - People don’t have enough context to make a decision
serena’s list of gripes 😞

1. Prompts are interruptive

2. Prompts are repetitive
   → Your prompt is not the only one!

Cookies

At website dot com we use cookies for a 😊better😊 browsing experience. Here’s a link to find out more but I bet you won’t click it.

I accept the cookies  “Settings”
serena’s list of gripes

1. Prompts are interruptive
2. Prompts are repetitive

⇢ Your prompt is not the only one!
⇢ We are adding more capabilities still
⇢ Interruptions and repetitions → Satisficing

Panic And Click Anything™
serena’s list of gripes 😞

1. Prompts are interruptive
2. Prompts are repetitive
   - Your prompt is not the only one!
   - We are adding more capabilities still
   - Interruptions and repetitions → Satisficing

   Panic And Click Anything™
serena’s list of gripes 😞

1. Prompts are interruptive
2. Prompts are repetitive
3. Risks are hard to explain

→ I don’t know what a “web serial” is and at this point I’m too afraid to ask

→ Imagine explaining the concept of fingerprinting to someone who is busy with something else
example.com wants to

-Allow your fonts?

A main risk for Fonts API is fingerprinting via the list of installed fonts — not the files themselves.

Imagine explaining that!
serena’s list of gripes 😩

1. Prompts are interruptive
2. Prompts are repetitive
3. Risks are hard to explain

[ long, heavy sigh ]
3. THE THINGS WE TRIED
Oh, prompts are interruptive?

* Introducing: the permission request chip

  → Out of the way

  → Grant rates dropped from 20% to less than 1%
IGNORANCE

INTERUPTION
Oh, risks are hard to explain?

- Automatic permission revocation
- One-time grants
Oh, there’s too many prompts?

* Hide what people don’t want
  - Based on user behaviour (settings opt-in, repeated denials)
  - Based on site behaviour (high-prompt/low-grant sites)

* Chrome Permission Suggestion Service (CPSS)
  - On-device, privacy-preserving ML that predicts likelihood of user grant
  - If highly confident (>95% precision) user will not grant, we show the chip UI

4. THE DEEPER PROBLEM
Why is the user only brought in at the end?
How web APIs work

1. Website tries to use a capability
2. Browser checks for permission
3. If no permission → pop prompt
   → At no point before this does the user need to be involved!
How we currently design software to handle permissions is a “developer-push” model.
Felt, A. P. (2012). *Towards Comprehensible and Effective Permission Systems*
GOAL:
Detect user intent
5. BEYOND PROMPTS
5 concepts

- User selection
- Piggyback off known concepts
- Sandbox risks
- Direct controls
- Embed triggers
User selection

Choose a device to pair with example.com

- Serena’s Pixel Buds Pro
- WONDERBOOM
- Serena’s Pixel Buds
- BT IntuosPro M

Scanning...
Known concepts

Choose file  No file chosen
Sandbox risks
By the very act of turning something on, you learn how to turn it off.
Embed triggers

Something the Chrome team is thinking about: an HTML element that websites can use on their pages to let people “pull” on the capabilities they want, they choose and are ready for it.
6. TO THINK OUTSIDE THE BOX WE MIGHT NEED BIGGER BOXES
What did we learn on the show today?

1. Permissions are for detecting user intent
2. Prompts do not scale
3. Solutions so far seesaw between interruption & ignorance
4. We need user signals earlier in the interaction flow
5. Five concepts beyond just prompting
From *developer-push* to *user-pull*
We can design permission moments directly into the natural flow of using something.
This allows us to enable a more powerful web without sacrificing usability, security or privacy.
By adding new ideas to our toolkit, we can take some pressure off the humble permission prompt.
THANKS!

Further reading—