Securing Web App Distribution:

Source Code Transparency

What do we want to achieve?

- Enable web apps that don’t trust the server
  - Web apps that use client-side encryption / E2EE
  - Web apps that don’t send any data to the server

Prior Art

- Web Extension
  - Check that source code matches the version on GitHub
  - Check that source code is signed by third party
- Service Worker
- Binary Transparency(?)
- SRI (but we want RI)

Source Code Transparency

- Publish (a hash of) the source code in a transparency log
- Browsers check the source code they receive against the log
- Ensures that everyone gets the same code
- Auditors can then check the source code with the help of reproducible builds, SBOM(?), CSP, SRI, etc.

Aside: Certificate Transparency

- Transparency log of all TLS certificates
- We could piggy-back on top of this

(More) Concrete Proposal

- Create a signed Web Bundle
- Send the hash to a (new) Source Code Transparency log
- Create an X.509 extension indicating that SCT should be used
- When present: browser goes and check the hash in SCT log

Challenges

- Allow updates immediately? Or only after inclusion in the log?
- Every page (that the user might land on) should be checked
- What about error responses?

Thanks! Questions?