RTCWeb Security Considerations

IETF 81

Eric Rescorla

ekr@rtfm.com

Consensus (or at least silence) on most security issues

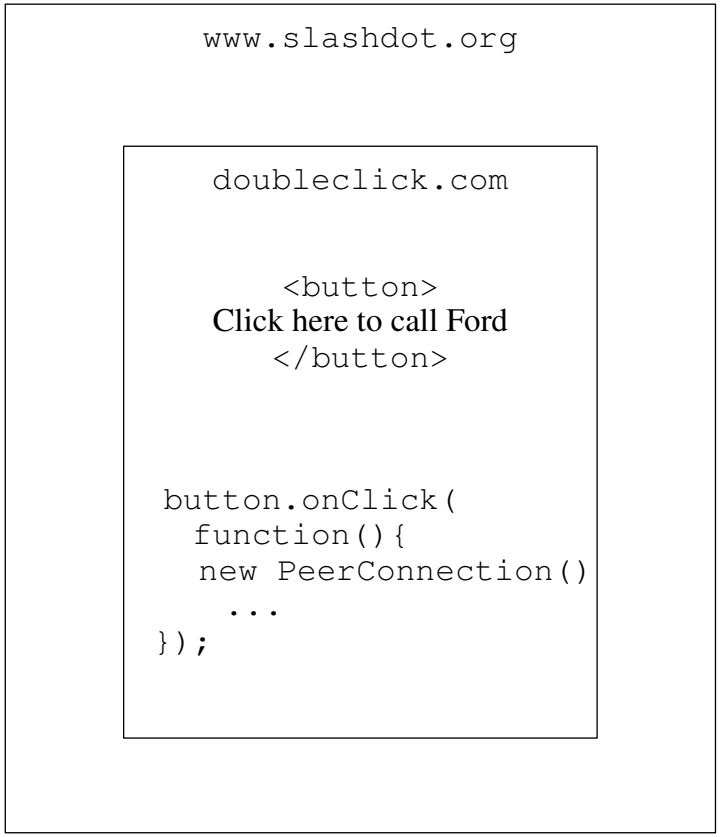
- Addition of this capability *must not* violate core browser security guarantees
- ICE *must* be used to prevent cross-protocol/voice hammer attacks
- User consent *must* be obtained (somehow) prior to providing mic/camera access
  - Scoped to *origin*[*]
- Sites *should* only allow calling from HTTPS pages
  - Browsers *should* forbid calling from mixed content pages
- *Must* provide communications security[*]

[*] More on this shortly

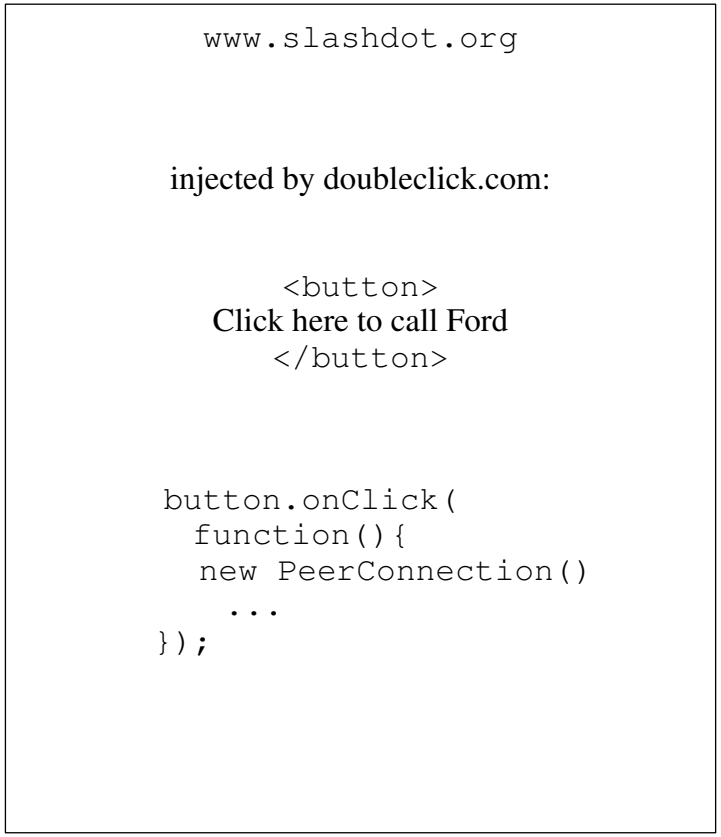
Scope of user consent

- This is not wholly an IETF issue
  - But it bears on the use cases
- Here's what I said at the interim:
  - *Remember: need to avoid in-flow dialogs*
    - *Consent cannot be obtained for each call*
  - *Most likely need to a get approval ahead of time*
    - *E.g., via an application "install" experience for each site*
- Does this work for all use cases?

Ad Hoc Calling from Embedded Advertisements



Option A: Ad in an IFRAME

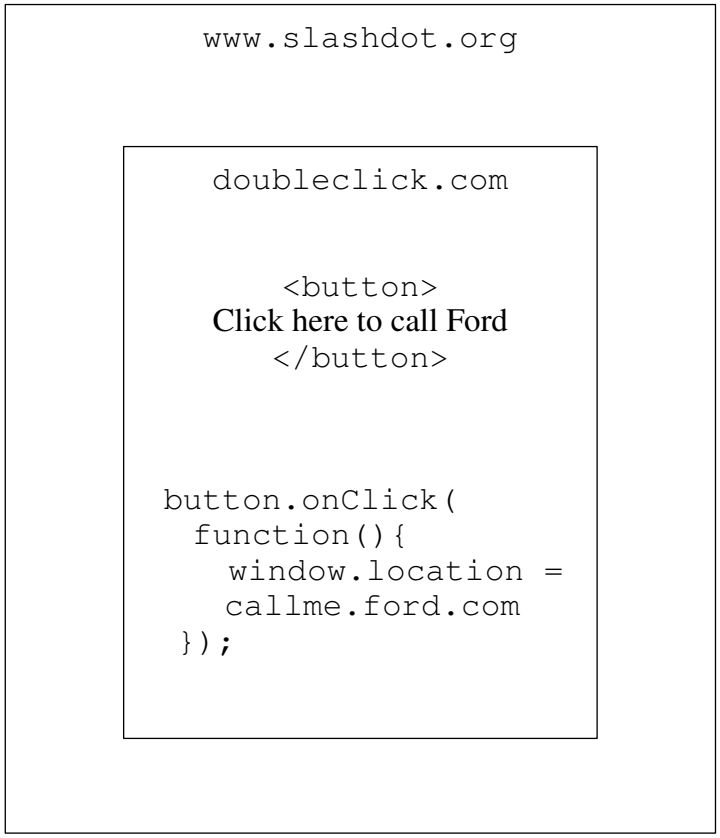


Option B: Injected ad

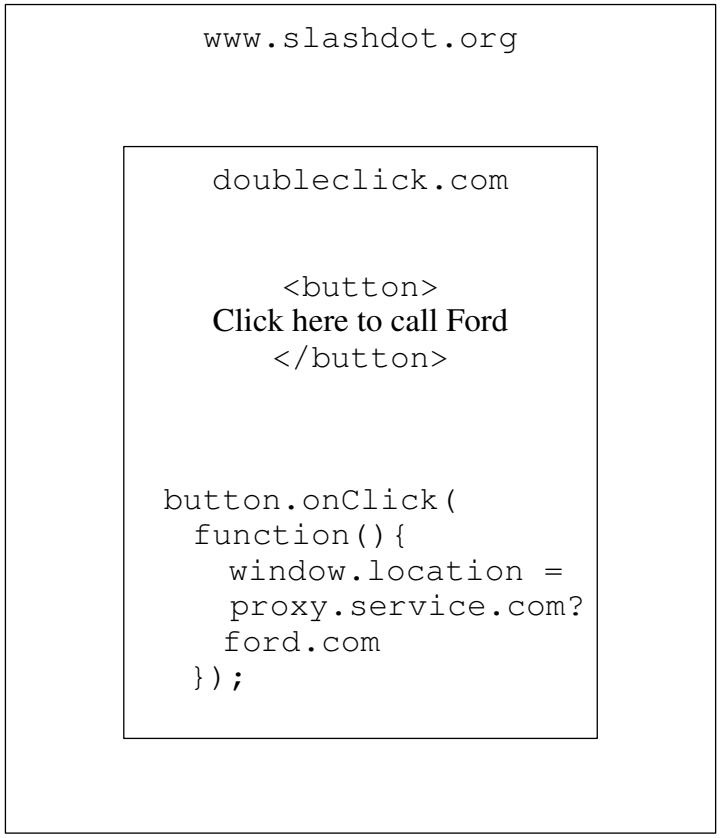
Threat Impact of Ad-Hoc Calling

- In neither case is the site calling the API anyone the user has a relationship with
  - Option A: Slashdot; Option B: DoubleClick
  - They don't even know about DoubleClick
- "Click here to let Commander Taco access your camera and microphone"
  - We would rather not have users click here
- OK, so that's not going to work

Ad Hoc Calling from Embedded Advertisements (II)



Option C: Call from target's site



Option D: Proxied call from service

OK, so that's a little better

- Option C: "Are you willing to let Ford use your camera and microphone"
  - We're now into "click here to screw yourself" territory
  - And what about "F0rd"? Or "Ford models"?
- Option D: "Are you willing to let this calling service use your camera and microphone"
  - Could maybe do this upfront
  - How does the calling site decide whether to complete the call?
    - Maybe it's got its own dialog?
- Again, not completely an IETF issue, but our guidance probably appreciated

What about communications security?

- **Must** provide security against message recovery and message modification
  - For both media (voice/video) and data
  - All the usual protocols work fine for this part
- What about threats by the calling service itself?
  - Controls nearly all the UI
  - Browser needs to protect the user from the calling service
    - But direct interaction is difficult
- Potential attacks by the calling service

  Retrospective: The calling service is is non-malicious during a call but is subsequently compromised (preventable)

  During-call: The calling service is compromised during the call it wishes to attack (hard to prevent)

Protecting Against Retrospective Attack

- Assume attacker has access to encrypted media stream
- If calling service has access to traffic keys, attack is trivial
  - Even worse in Web contexts because of extensive logging
  - Hard to believe service can adequately "forget" keys it has seen
    * Most sites log requests at many different locations
- Right approach: asymmetric key-based exchange between the endpoints
  - Secure against retrospective attack even if mediated by calling service
  - APIs **must not** allow calling service to subsequently extract traffic keys
  - Best if it provides perfect forward secrecy (PFS)

Protecting Against During-Call Attack

- Need to have asymmetric key exchange
  - Otherwise passive attack is trivial...
  - Defeating asymmetric key exchange requires MITM attack
- Defenses against MITM
  - Keying material verification
    - Third-party authentication service (we know this won't work)
    - Out-of-band fingerprint exchange
    - Short authentication string
  - Key continuity
    - Verify that the same key is used for each call

Allow unencrypted RTP at all?

- Practically all existing standards-based VoIP implementations uses RTP
  - With no cryptography
- If we want to interoperate with those deployments, we must support RTP
- How likely is interop in any case?
  - Interop already requires ICE—not widely deployed
  - For PSTN interop you're likely to have to SBC anyway
- Basic choice: limited interop versus security all the time

Positions With Significant Support at the Interim (in my opinion)

- DTLS-SRTP all the time
  - MUST implement DTLS-SRTP
  - MUST NOT do RTP or SDES
  - Backward compatibility not so good
- DTLS-SRTP + RTP and SDES-sorta
  - MUST implement DTLS-SRTP; MUST be the default
  - MUST implement RTP
  - MAY implement SDES
- UI requirements (see next slide)

UI Requirements (draft-kaufman-rtcweb-security-ui)

- UAs MUST provide an indication of the security characteristics of audio and video
  - MUST include the cipher suite
  - SHOULD provide an indication of PFS or not
- UAs MUST provide a mechanism for verifying keying material if a secure channel is available
  - MUST provide a binding to the far station identity (e.g., fingerprint, SAS)
- General consensus on this stuff

Relevant Drafts

draft-rescorla-rtcweb-security-00

draft-johnston-rtcweb-media-privacy-00

draft-kaufman-rtcweb-security-ui-00