RTCWeb Security: What Assurances Can/Should We Deliver?

IETF 81

Eric Rescorla

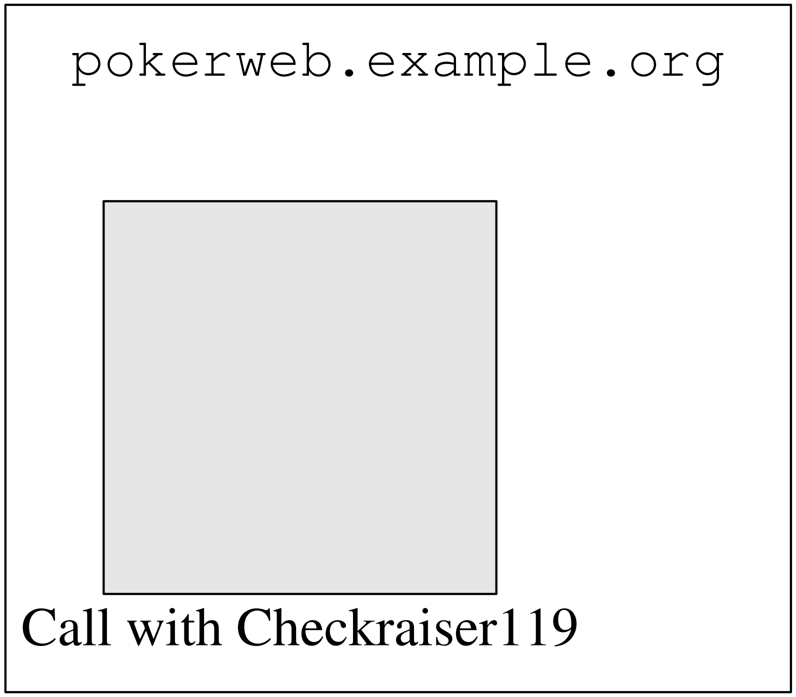
ekr@rtfm.com

Overview

- RTCWeb functionality is too dangerous to enable by default
  - General agreement that users must consent to its use
- But how are they to consent intelligently?
  - What properties do users expect/want?
  - How well do they conform to what we can technically deliver?
- Objective of this discussion
  - Work through a bunch of the common cases
  - Try to answer above questions

Gaming Sites I: A Closed, Anonymous World

- I sign onto PokerWeb looking for a game
  - "Find me someone to play heads-up no-limit"
- Result: I end up in a call with someone...



pokerweb.example.org

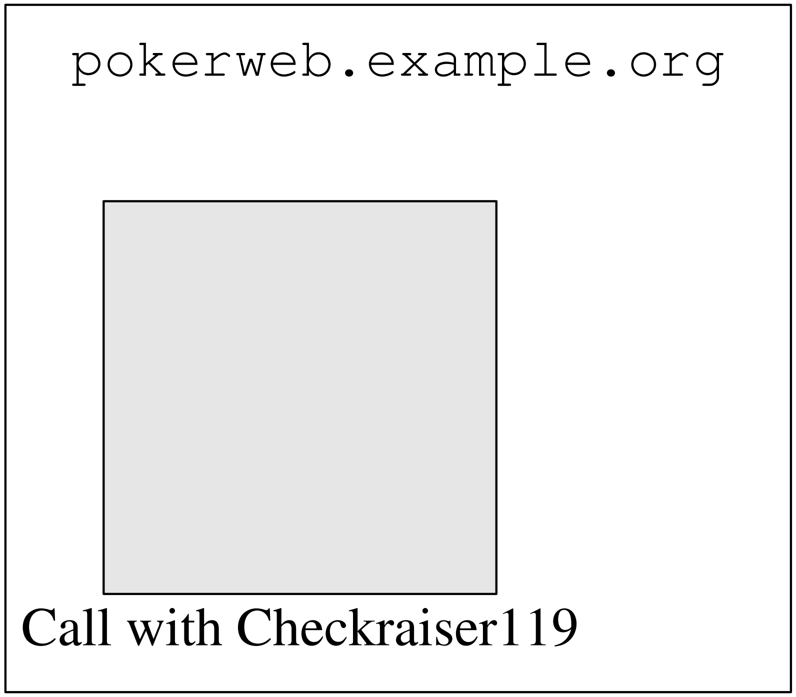
Call with Checkraiser119

User Expectations?

- Site Relationship
  - I am visiting PokerWeb
  - They control the call
- Duration of consent
  - Long-term: I'm going to playing a lot
  - Don't want to consent each time
- Peer identity
  - Anonymous: I have no idea who this person is
  - ... they were just assigned to me by PokerWeb

Gaming Sites II: Repeat Business

- I sign onto PokerWeb looking for a game
  - "Is CheckRaiser 119 on line?"
- Result: I end up in a call with someone...



pokerweb.example.org

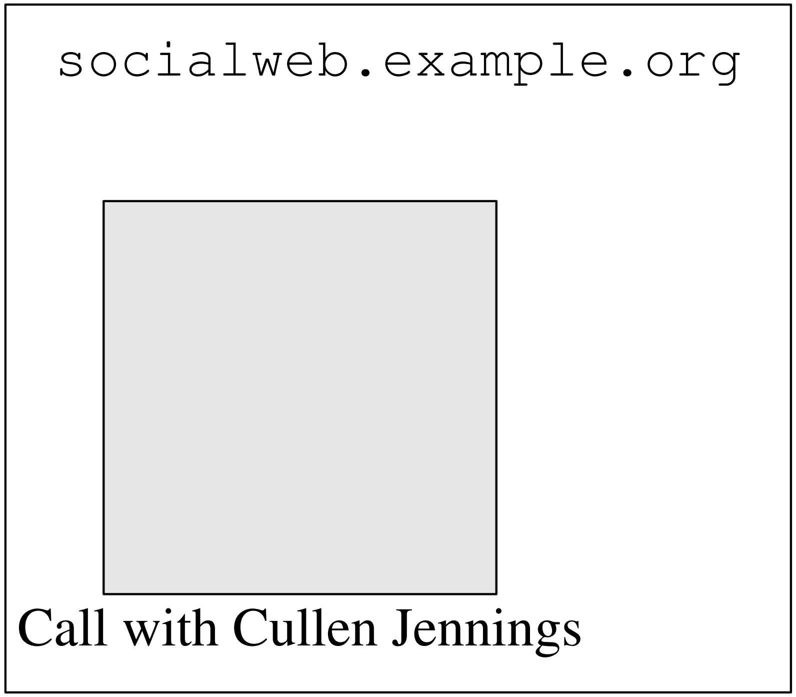
Call with Checkraiser119

User Expectations

- Pseudonymity
  - *I'm playing with Checkraiser 119 (whoever that is)*
  - *... same person I played with before*
- This identity comes solely from PokerWeb
  - Displayed by their user interface (not in the browser chrome)
  - Identity was *assigned* by PokerWeb
    - ∗ Not globally meaningful
- Do users really expect the browser to protect them here?

Calling Services I: A closed, non-anonymous world

- I have an account on SocialWeb
  - ... “friends” with a bunch of my real-world friends
  - Want to call one of my friends



socialweb.example.org

Call with Cullen Jennings

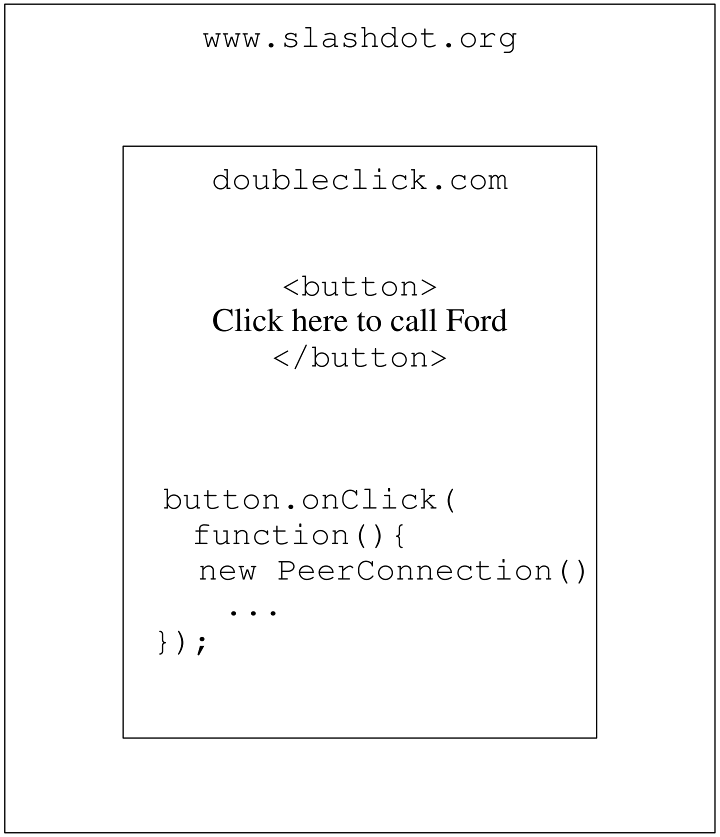
User Expectations

- Site Relationship
  - I am visiting SocialWeb
  - They control the call
- Duration of consent
  - Long-term: I'm going to playing a lot
  - Don't want to consent each time
- Peer identity
  - I know who I am talking to
  - I added them to my friend list

Calling Services II: An open non-anonymous world (i.e., federated)

- I have an account on SocialWeb
    - But I have friends on AntisocialWeb
    - ... who I want to call
- I don't want to get into this one just yet...

Ad Hoc Calling from Embedded Advertisements



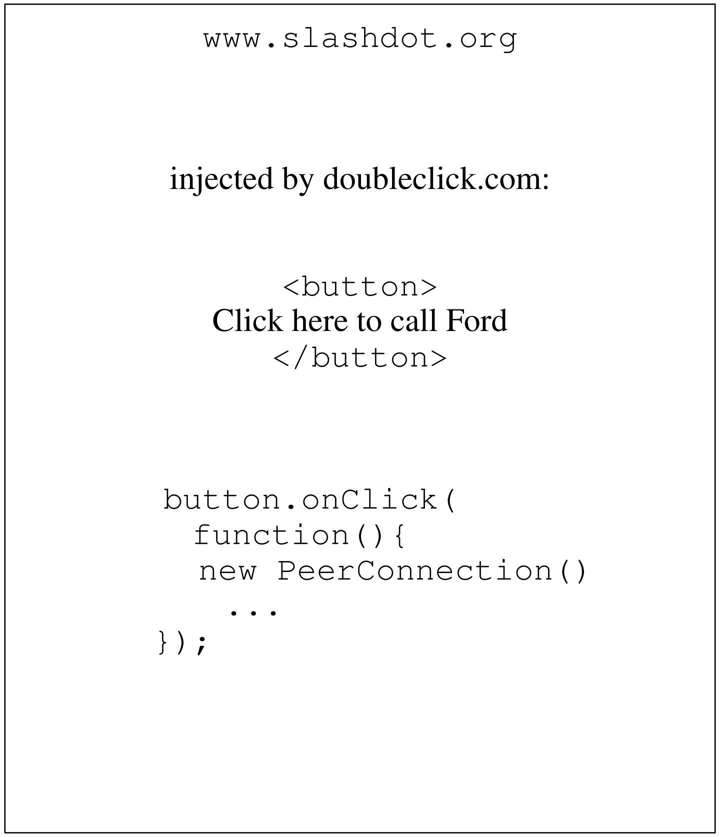
www.slashdot.org

doubleclick.com

```
<button>
Click here to call Ford
</button>
```

```
 button.onClick(
   function(){
   new PeerConnection()
     ...
});
```

Option A: Ad in an IFRAME



www.slashdot.org

injected by doubleclick.com:

```
<button>
Click here to call Ford
</button>
```

```
 button.onClick(
   function(){
   new PeerConnection()
     ...
});
```

Option B: Injected ad

User expectations

- When I place this call I'm talking to Ford
- Not giving Ford long-term access to my camera and microphone

- But I'm on Slashdot...
  - Do I think Slashdot has endorsed this?

Common Themes

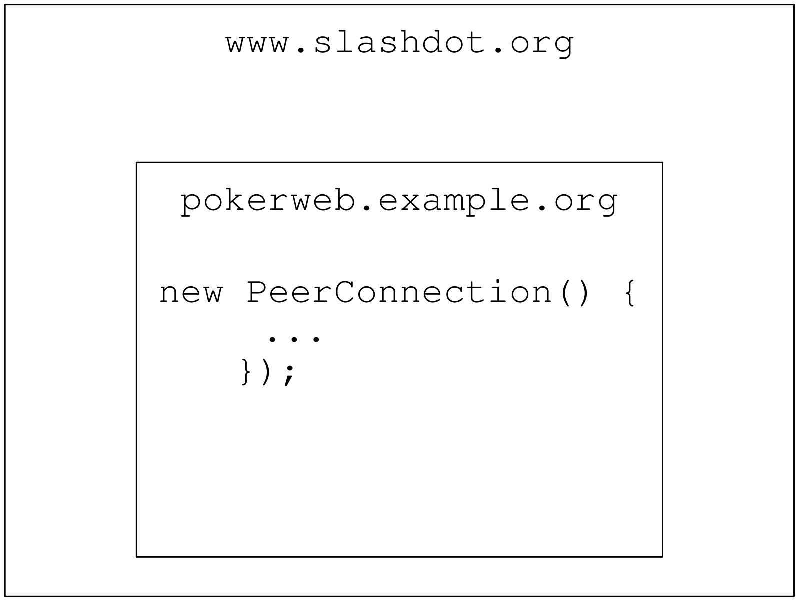
- Consent
  - How to make long-term consent grants secure
  - How to safely give short-term consent
- Authenticating the person you are talking to
  - Anonymous peers
  - Pseudonymous peers
  - Peers with long-term identities

Making long-term grants secure

- Basic problem: site controls interface
  - They can initiate a call to anyone
  - But I don't want them to bug my house
    - These are semi-contradictory
- How do I allow someone to make calls without letting them make calls?

Partial Digression: Network Attackers

- Assumption: I've authorized PokerWeb
- I'm in an Internet Cafe and visit any URL
  - Attacker injects IFRAME pretending to be PokerWeb
  - But calls go to him



www.slashdot.org

pokerweb.example.org

```
new PeerConnection() {
     ...
    });
```

- Result: attacker has bugged your computer

User Expectations

- It's safe to authorize PokerWeb and then surf the Internet
  - Without being bugged
- Including on insecure networks
  - This may include your home network
- Unfortunately, this is not true here...

Potential Long-Term Consent Security Features

- Live with it (require clear UI)
- Require user interaction with browser chrome for all calls
  - User interaction alone is not enough because of clickjacking
- Require user interaction for calls to “new” peers
- Require JS to be delivered over HTTPS (only stops network attacks)

- None of these are that great

Short-Term Consent

- Need some mechanism to allow immediate calls
  - To people you have no previous relationship with
- Conflicting requirements
  - Low-impact
  - Not something users will just click through
  - Can we do anything to help here?

Characterizing Short-Term Consent

- User doesn't really know who they are calling
  - And if they do, it's "Ford"
- We don't have the technical means to give this kind of identity
  - Best-case scenario is an authenticated domain name
  - Do you want to call `www.ford.com`?

API impact of short-term consent

- One natural design is to show “self” picture a la Facetime
  - Here’s your video, do you want to set up the call
- But this implies *some level* of device access prior to permissions grant
  - Step 1: display video to user with call start button
  - Step 2: start call
- Out of scope?

What about the site I'm visiting?

- Adam Barth: the user thinks he's on Slashdot
  - Even though Slashdot neither placed the ad nor is the called party
  - Only vaguely conscious of ad networks
- Should the top-level site get to have an opinion?
  - Protect the user?
  - Protect its reputation?
  - What about privacy?

Enforcing Pseudonymity

- Assumption: if you care who you're talking to you're going to use SRTP
- "This is the same person" enforced by PokerWeb
  - Identity could be owned by someone else tomorrow
- What can the browser enforce? Cryptographic continuity
  - This is the same *machine* I talked to last time
  - Not that great a substitute

Verifying who you are talking to

- Assumption: if you care who you're talking to you're going to use SRTP
- Assumption: we only sort-of-trust the calling site
  - (Alan Johnston, Matthew Kaufman)
- Need to be able to cryptographically verify the other side
  - PKI plausible for some applications (especially when you're calling an organization)
  - Verify keying material (fingerprint, SAS) through side channel [draft-kaufman-rtcweb-security-ui-00]

Executive Summary (mostly opinion)

- Can't completely eliminate threats from long-term trusted sites
  - Without undue impact on user experience
  - Basic principle: minimize threats where possible and allow user verification of state
    - ∗ Trust but verify
- Ironically, short-term consent is in some sense more secure
  - Likely user will have to directly grant consent
  - ... in browser chrome
  - ... and only for this call
  - You may not really know who you're talking to but that's not too bad