W3C WebCrypto Analysis

Kelsey Cairns and Graham Steel

September 10, 2014

Project Goals

- Provide analysis of W3C WebCrypto API
  - Formalize API
  - Analyze model with respect to key safety
  - Assess practical concerns

Modeling

- Model checker searches valid sequences of calls
- Modeling tools:
  - AVISPA – `avispa-project.org`
  - SATMC – `www.ai-lab.it/satmc/`
  - py2if – homemade, purpose built for WebCrypto
- Model attacker has client-side API access
- Tested properties of:
  - `extractable`, `usages`
  - message passing
- Extrapolated to test case properties

Questions We Had

- Can a key be extracted from a client?
- Can a key's `usages` be changed?
- What security properties hold for messages:
  a) Sent from client to server?
  b) Sent from server to client?
- What threats apply to use cases?

Answers We Got

- Can a key be extracted from a client?
  - Unextractable keys are not extractable
- Can a key's `usages` be changed?
  - `usages` are only enforced for unextractable keys
- What security properties hold for messages:
  a) Sent from client to server? – More integrity attacks
  b) Sent from server to client? – More confidentiality attacks
- What threats apply to use cases?
  - Most vulnerable to at least one type of attack

Example

- Client, server share symmetric key s with wrap/unwrap `usages`
- Client wishes to send wrapped key $\{k\}_s$ to server

  Attacker can:

  1. Export k
  2. Replace s
  3. Call client's *unwrap* operation on $\{k\}_s$ to discover k
  4. Polyfill client's *wrap* operation to send raw key

- Attack 3 is eliminated if s does not have unwrap `usage`

Other Things We Thought About

- Review of included crypto algorithms
  - Nearly half have no security proof or known weakness
- No key storage specified
  - Client-side key storage is a liability
- Polyfills
  - Too easy to replace `subtleCrypto` functions
- Client-trusts-server security model

WebCrypto Tracer

- BSD licensed Chrome extension
- Trace client-side calls to webcrypto
- Report activity to user