A First Look at Firefox OS Security

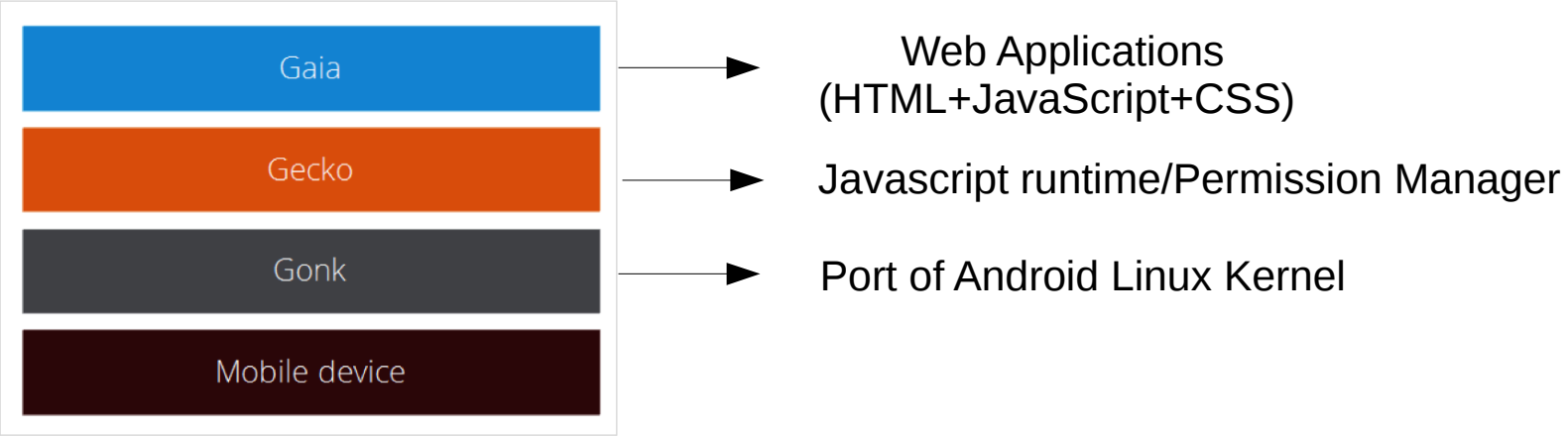
Daniel Defreez*, Bhargava Shastry†
Hao Chen*, Jean-Pierre Seifert†

* University of California, Davis
† Security in Telecommunications, TU Berlin

Outline

- Results from security evaluation of Firefox OS
  - Disclaimer: Only covers work done at TU Berlin
- Overview
  - Mozilla's re-use of system software has a subtle security issue
    - One-click bypass of web content authentication
  - Hosted app code is not signed
    - Overwhelming majority don't use https
  - Apps exchange sensitive data over *http*
    - Intercepted phone numbers, passwords etc.

Background: Firefox OS



- Native applications: No
- APIs: HTML5 Device APIs (standardized+custom)
- Runtime: Javascript interpreter

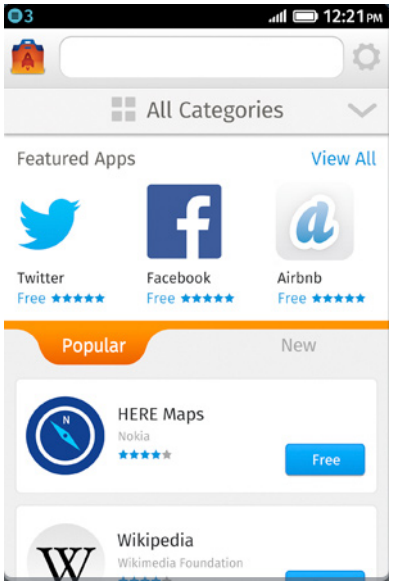
Source: https://developer.mozilla.org/en-US/Firefox_OS/Security/Security_model

Background: Firefox OS Apps

- Three security levels for applications
  - Unprivileged: Limited permission set
    - Can be remotely hosted or installed on device (packaged)
  - Privileged: Access to sensitive permissions
    - Must be packaged, digitally signed after review
  - Certified: Access to telephony stack and device hardware
    - Pre-installed applications developed by platform vendors
    - Third-party developers don't have access to Telephony, Bluetooth, and Camera APIs

App Review and Publishing

Signed

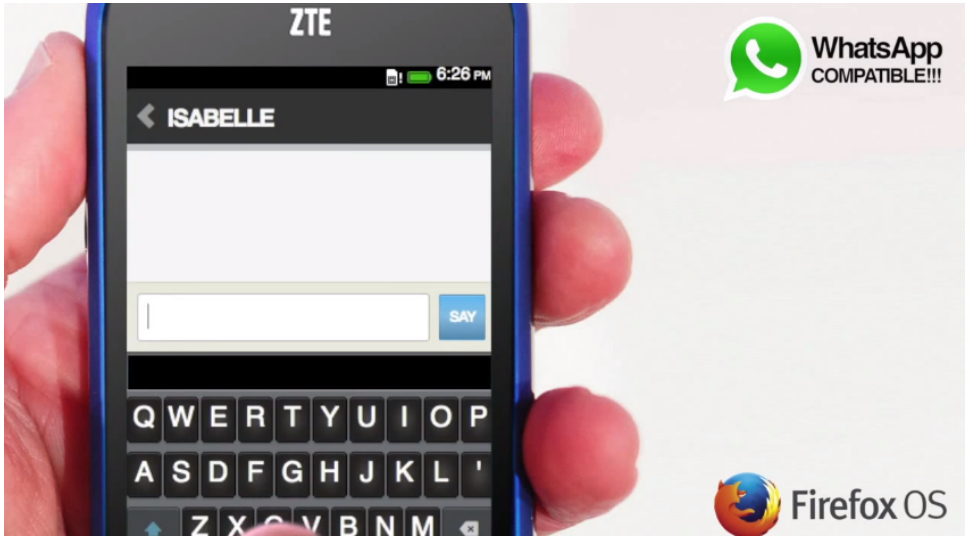


Vulnerable Traffic: Code

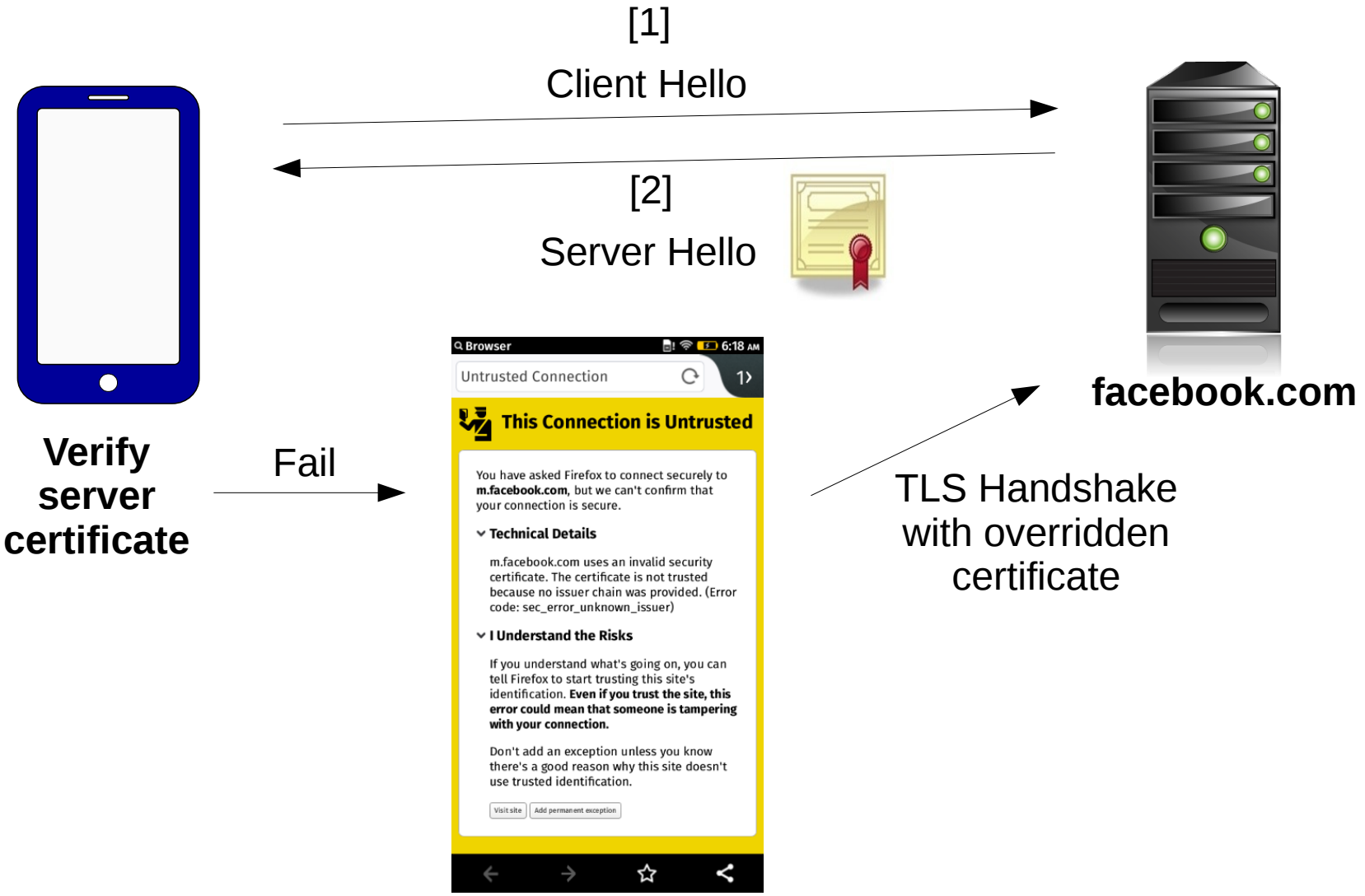
- Unauthenticated installation of unprivileged apps
  - 92% of hosted apps are fetched over http
  - Mozilla *recommends* https
- Proof-of-Concept
  - Added record audio ``feature'' in an unprivileged app
  - Limitations
    - Restricted permission set for unprivileged apps
    - User prompt for sensitive permissions

Vulnerable Traffic: User Data

- 50% of privileged apps make http requests
  - Remember: These apps have been reviewed!
  - Privacy concerns
    - Phone numbers in the clear: ConnectA2, Free SMS



TLS Certificate Overrides



Certificate Override Caching

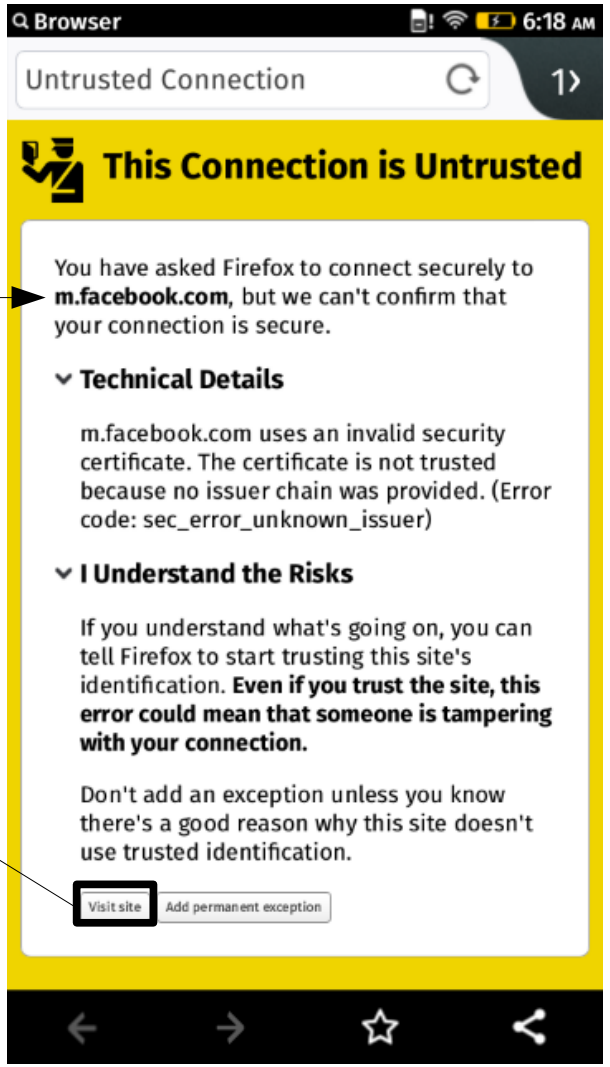
- Although override is ``temporary''
  - Persists until a device reboot on Firefox OS
- Differences between desktop browser and Firefox OS
  - UI and UX inconsistency
  - Architectural difference: Process model
  - Security principal

Inconsistent UI: Site ID Button

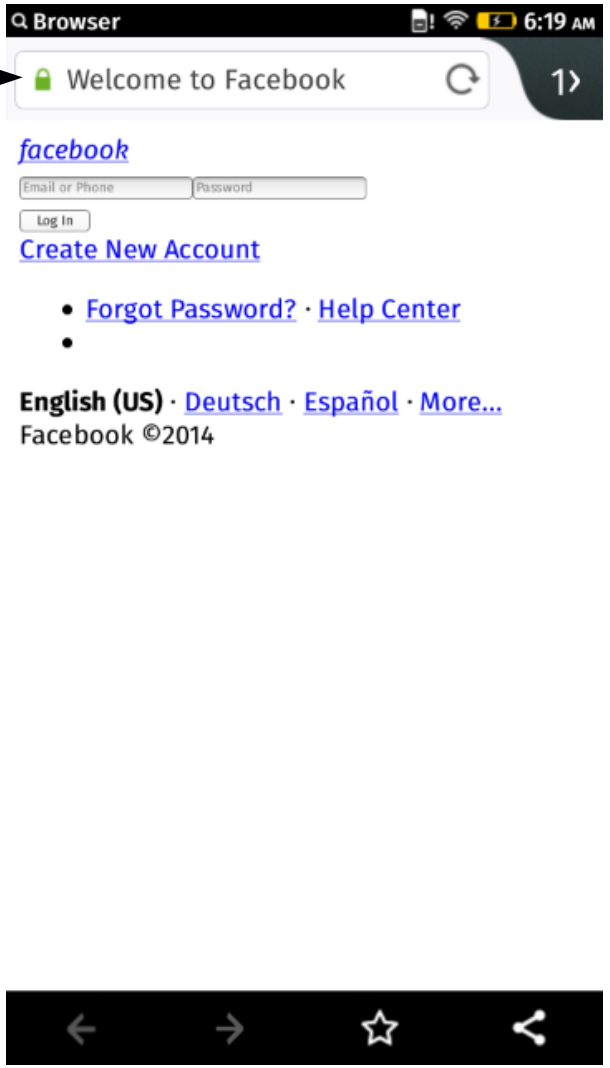
App

Origin

Visit Site

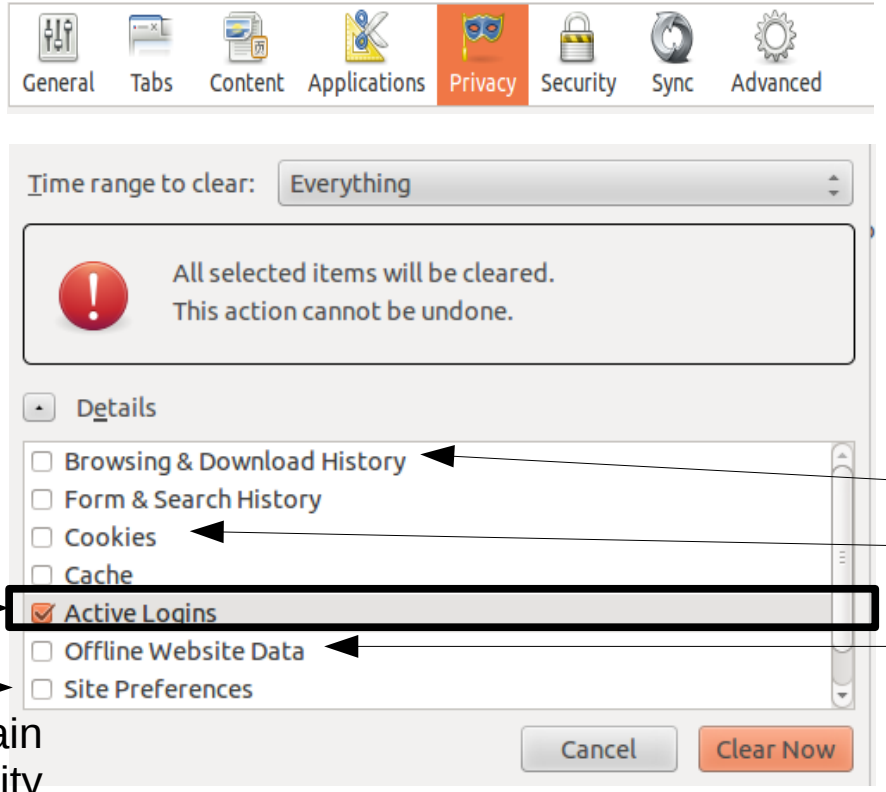


Misleading

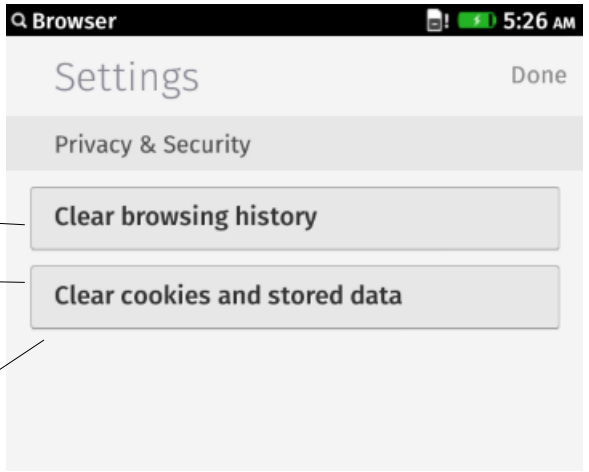


Inadequate UI: Site Security Info

Inadequate UI: Revoking Overrides



Contain security options



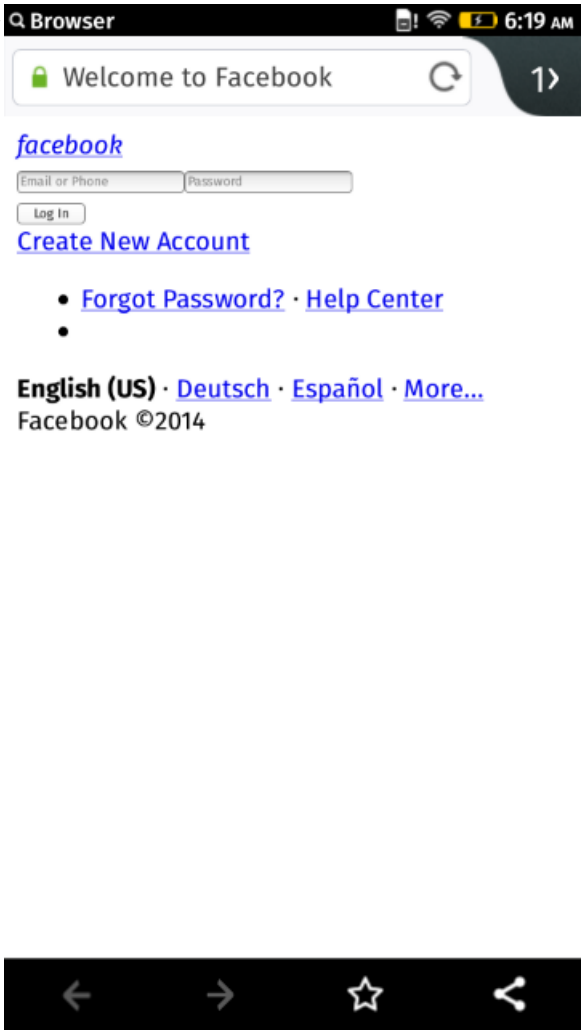
Firefox Desktop Browser

Firefox OS Browser

UX Inconsistency: Overrides Across Apps

Browser app

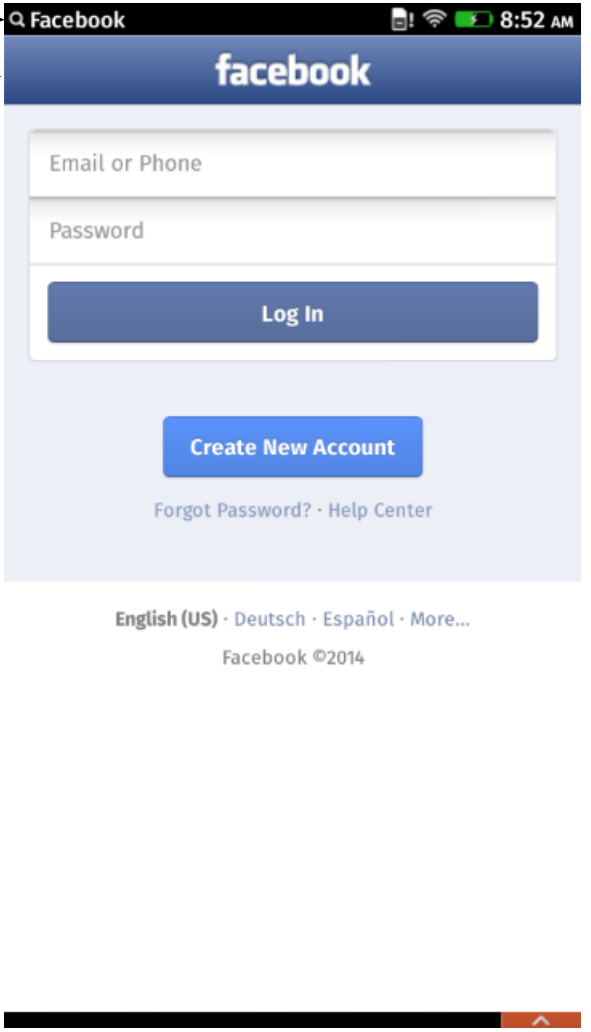
App 1



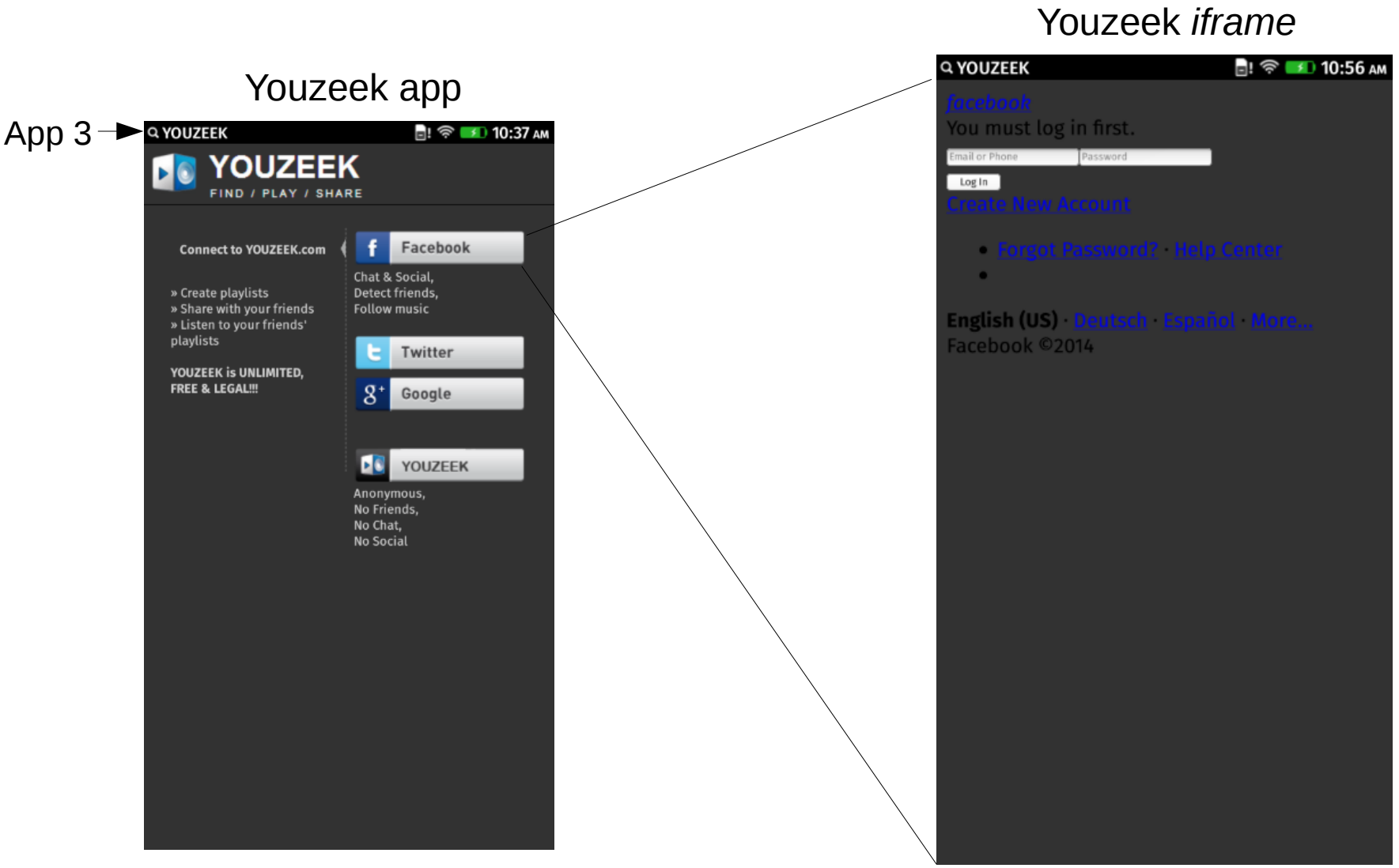
Facebook app

App 2

Security indicator absent!



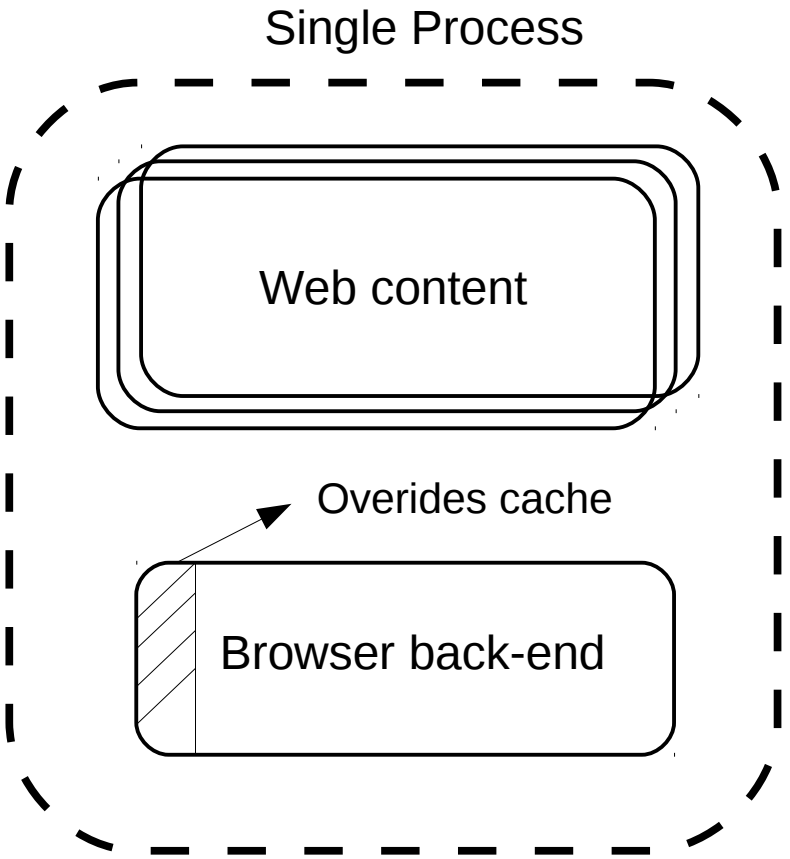
UX Inconsistency: Overrides Across Apps



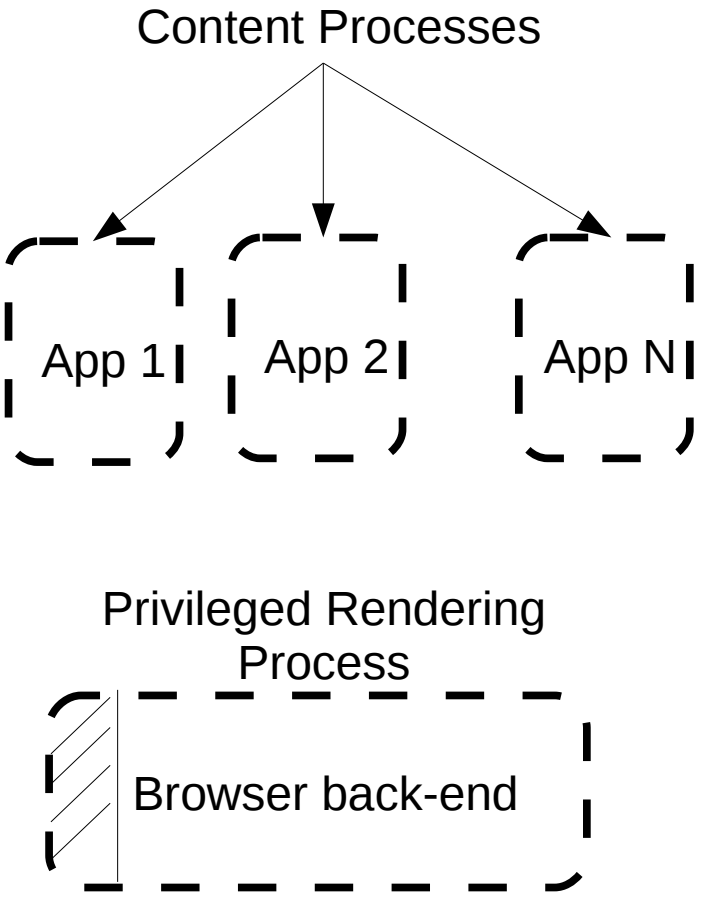
Persistent and Cross-App

- Why this happens?
  - Single process (desktop) vs. Multi-process (FFOS)
  - Security principal

Process Model



Firefox Browser

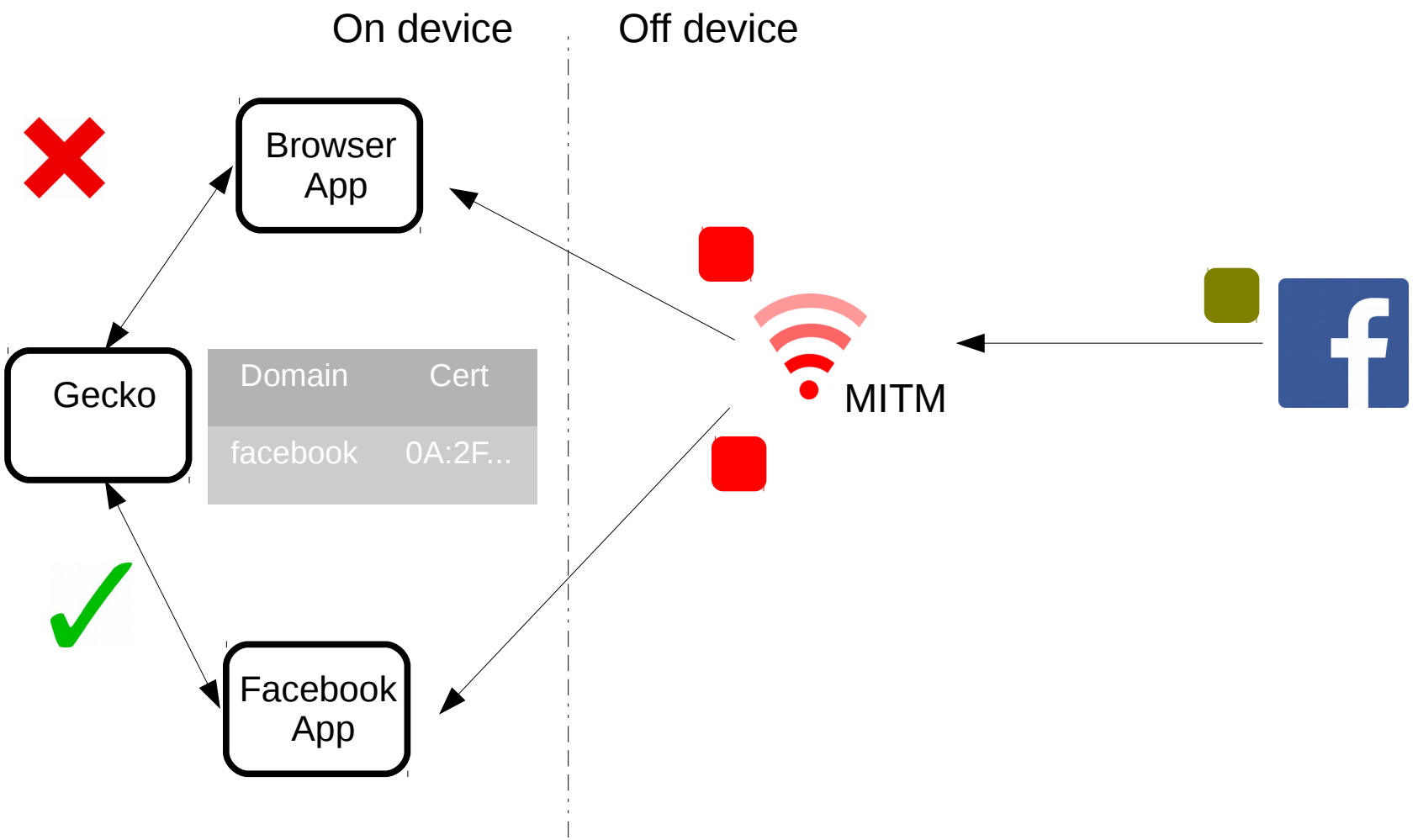


Firefox OS

Security Principal

- What is the unit of isolation?
  1. facebook.com? [Same origin policy]
  2. {Facebook App, facebook.com}?
- Legacy Gecko identifies [1]
  - No notion of an ``app''

Threat Scenario



Summary

- TLS override certificate caching
  - Certificate overrides once approved, persist!
  - No UI to revoke overridden certificates
  - Security indicators either misleading or absent
  - Overrides apply across applications
- Vulnerable web traffic
  - Unauthenticated installation of unprivileged apps
  - Unencrypted data sent by privileged apps

Disclosure

- Mozilla notified of bugs
  - Not fixed yet, no reason explicitly stated
- Platforms affected
  - All Firefox OS releases thus far
  - Bugs persist in upstream code as of February 2014

Future Work

- Scratched the surface so far
- Other side-effects of retrofitting web apps
  - Different process model
  - Multiple security principals
- How risky is mobile web browsing vs. desktop browsing?

Thank You!

Questions?

Threat Scenario