Università degli Studi di Trento

Security-By-Contract for the Embedded Internet (or the end of trust as we knew it)

F. Massacci[1] F. Piessens[2], I. Siahaan[1]
Univ. of Trento & Kath. Univ. of Leuven
www.massacci.org
www.s3ms.org

01/23/2009

"The" security models

- Provide Access to Security relevant API only to appropriately trusted widgets/apps
- Trusted = identity is known
- = coming from somebody trusted
- = widget/apps digitally signed
- = signature proxy for accountability
- = trusted community authorizes permissions
- = multiple signatories and signature profiles

- Little, little problem
  - Once I trust you, I'm at your mercy in the protection domain, but you are trusted, aren't you?
  - You can be victim of some cross-site scripting, gmail subject attacks but you will not do intentional "evil", will you?

A Video is worth a 10^10 words…

UK Channel 4 & related blogs

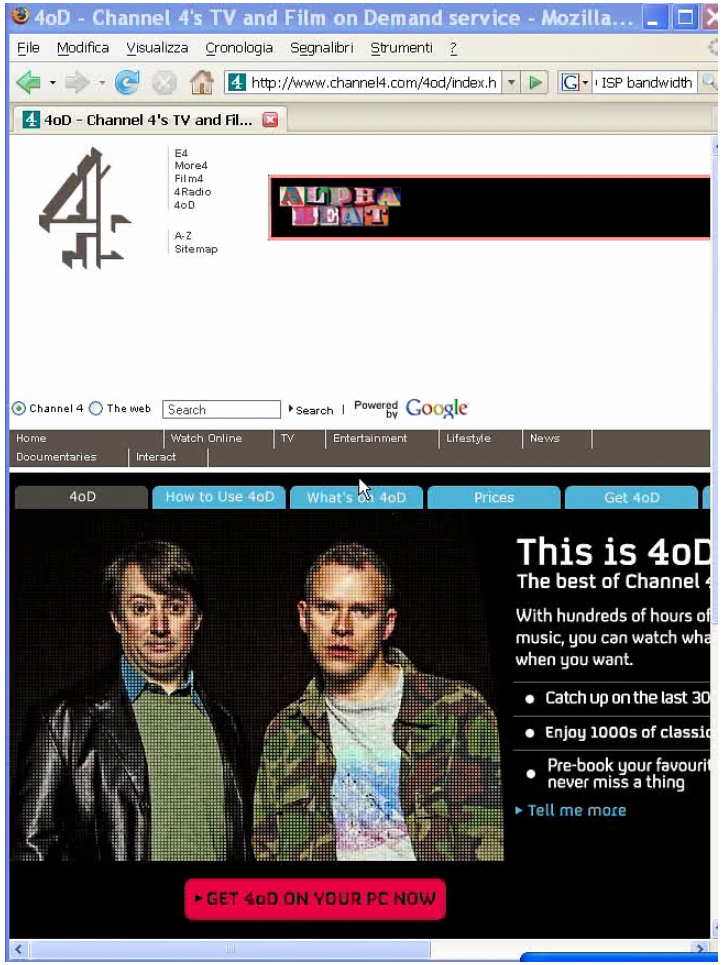
(Videotaped evidence on 9 June 2008 of earlier events)

Showing the death of trusted proof of origin (aka digital signature) as proxy of accountability

A Video is worth a 10^10 words…

UK Channel 4 & related blogs

(Videotaped evidence on 9 June 2008 of earlier events)



Showing the death of trusted proof of origin (aka digital signature) as proxy of accountability

Tu quoque Brute, fili mi…

- Do you know 4oD?
  - A software to view, stream, save and own TV movies
  - You download it from the Internet
  - But it installs on your PC a stealthy P2P servent…
    - which serves movies elsewhere in the world…
- But it's not shady software from rbnexploit.com
  - It's from Channel 4 (or BBC or Sky or) a reputable broadcaster
  - But servent isn't in the FAQ, isn't in the readme….
    - Hidden in the license agreement after N>>1 sections of legalese
  - But your ISP will let you know… the bill….
- Proof of origin was killed as a proxy of accountability
  - Because no DIGITAL claim is attached to a DIGITAL signature
  - And you cannot bootstrap accountability from nothing
  - Yes, but you could sue them… come on, give us a break

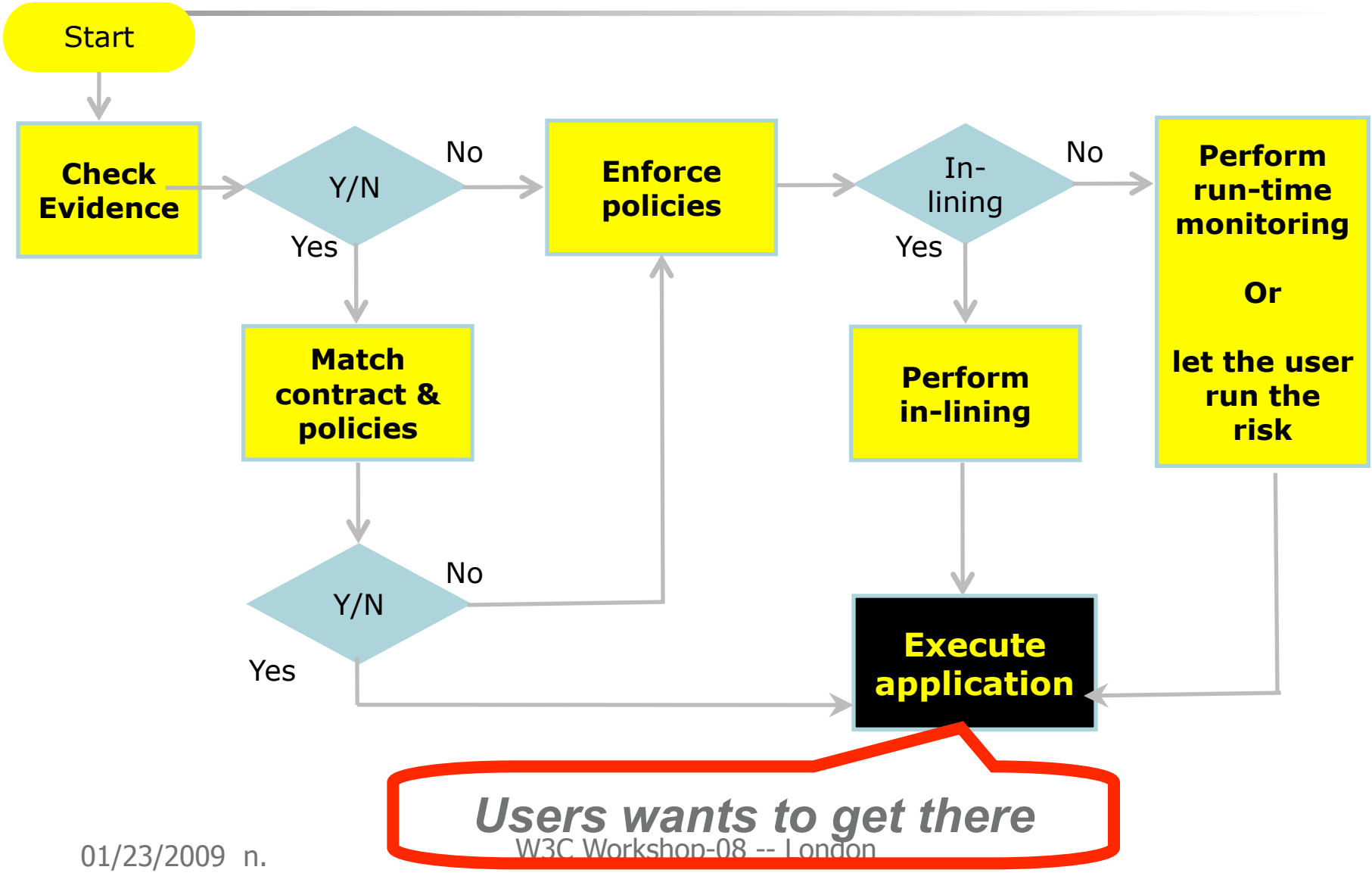
Security-by-Contract

- The Key Idea
  - contract carried by apps/widgets
  - Signature of Code+Contract (now only code)
  - policy specified by a platform
- What's in a code's contract?
  - (security) features of app/widget
  - (security) interactions with its host platform
  - Maybe proof–of-compliance of code
- What's in a platform's policy?
  - Platform contractual requirements on apps
  - Fine-grained resource control

Security-by-Contract - II

- Machine readable contract
  - As opposed to human readable contracts (see 4oD)
  - Embedded in the Manifest
  - (same idea as in widget 1.0)
- E.g. Simple format
  - If BEFORE/AFTER api
  - && Spec#, JML, OCL, Javascript-like conditions on parameters or return value
  - then javascrip simple ops (eg allow for this time)
  - i.e. XACML with state but no need to learn XML language different from javascript

Security-by-Contract - User's View



Security-by-Contract - IV

- What platform stakeholders want is… Policy Enforcement
  - Bad Applications cannot violate the platform policy
  - Operators don't get call at hotline (I never phoned Moldova)
  - Don't need VM owner cooperation to enforce them
  - = inoculate policy into uncontracted application
- What developers want is… Transparency
  - Enforcement mechanisms do not mess Good applications
  - No need of inoculation if contractual compliance
  - Even if inoculated rest is untouched
  - No need to disclose actual source code for inspection by community
- What end users want is… Policy-Contract Matching
  - Knowing whether the application is good for them
  - As they have different "policies" for game, business, etc)
- Formally guaranteed, not just hack+assert

Security by Contract for the Mobile Phones

- S3MS EU project --- www.s3ms.org
  - Applications come with a contract
  - Matching Midlet's contracts with Phone's Policy
  - Inoculation of policies for "untrusted/uncompliant" apps
- Promising results for .NET and Java
  - Enforcement, Transparency, and Checking
  - All formally guaranteed
  - Realistic policies
    - (eg no sms after access to PIM, only connect to this url)
- Demo
  - First Gaming application hacks access to user's PIM and send data to hidden phone (just Italy sorry no Moldova SIM)
  - same application with Security-by-Contract cannot do it

Security by Contract for the Mobile Phones

- S3MS EU project --- www.s3ms.org
  - Applications come with a contract
  - Matching Midlet's contracts with Phone's Policy
  - Inoculation of policies for "untrusted/uncompliant" apps
- Promising results for .NET and Java
  - Enforcement, Transparency, and Checking
  - All formally guaranteed
  - Realistic policies
    - (eg no sms after access to PIM, only connect to this uri)

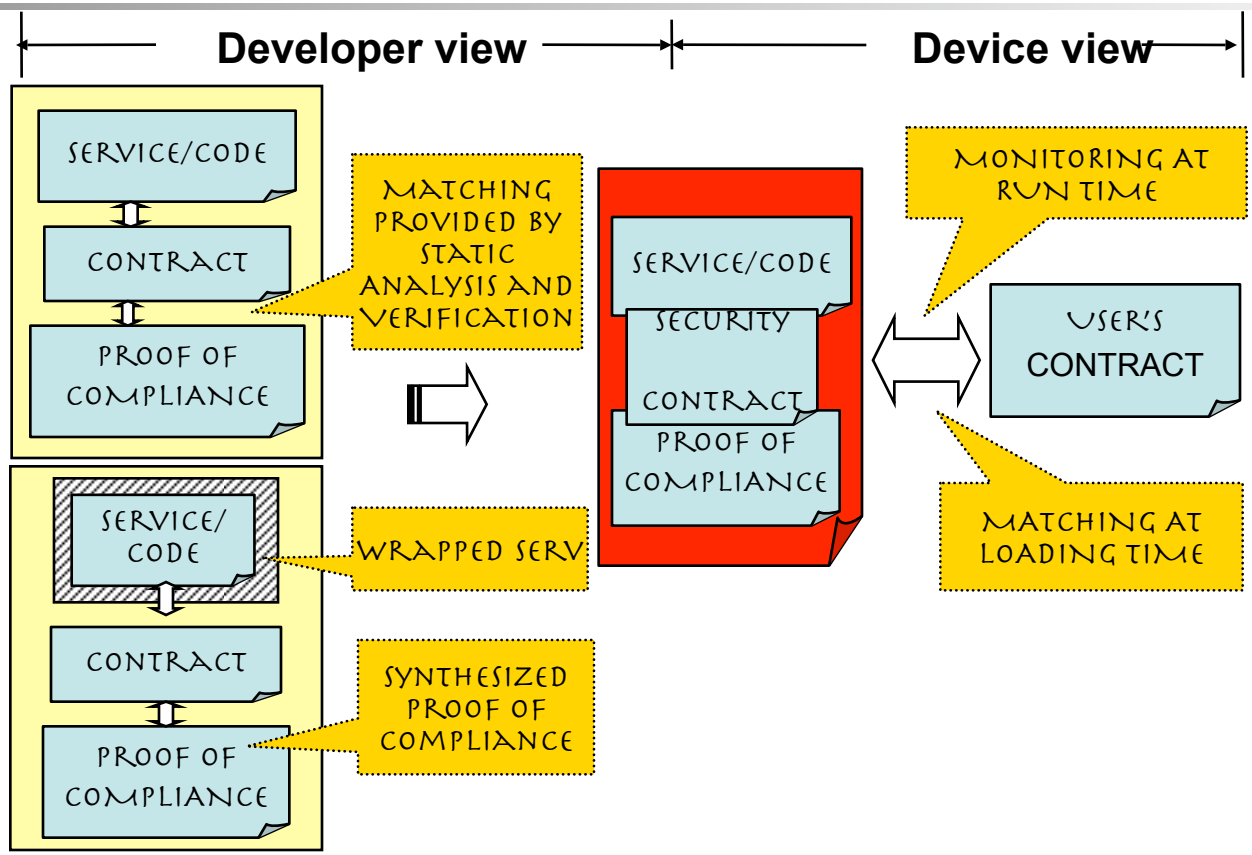
- Demo
  - First Gaming application hacks access to user's PIM and send data to hidden phone (just Italy sorry no Moldova SIM)
  - same application with Security-by-Contract cannot do it

Lots of things still to be done…

- Research-wise
  - From managed apps to webapps
  - Testing by contract?
  - Concurrency of threads?
  - Evidence/Proof generation for inlined monitors?
- Standardization-wise
  - Which are the security features?
  - Simple but expressive way to describe contracts?
  - Users presentation and questions?

"The" User Model

- I want to use services with lots of functionalities = access to Device API
  - Eg Roaming Monsters app by send SMS with friends
- But I want "control" of costly functionalities
  - Happy birthday widget shouldn't send a SMS to premium number in Moldova
- If I'm privacy aware, I want "control" access to my data
  - Chess Playing midlet has no business with my GPS
  - Mobile Maps has no business with my agenda
- At a level of details I can understand


Developer view
Device view
SERVICE/CODE
CONTRACT
PROOF OF COMPLIANCE
MATCHING PROVIDED BY STATIC ANALYSIS AND VERIFICATION
SERVICE/CODE
SECURITY CONTRACT
PROOF OF COMPLIANCE
MONITORING AT RUN TIME
USER'S CONTRACT
MATCHING AT LOADING TIME
SERVICE/ CODE
WRAPPED SERV
CONTRACT
PROOF OF COMPLIANCE
SYNTHESIZED PROOF OF COMPLIANCE