The challenge: How do we make security and safety sustainable?

Ross Anderson
Cambridge
How does IoT change safety?

• The EU regulates safety of all sorts of devices
• They asked Éireann Leverett, Richard Clayton and me to examine what IoT means for this
• Once there’s software everywhere, safety and security get entangled
• How will we have to update safety regulation (and safety regulators) to cope?
• We studied cars, medical devices and grid equipment but the lessons are much broader
The Big Challenge

• Established non-IT industries usually have a static approach – pre-market testing with standards that change slowly if at all
• The time constant is typically a decade
• When malicious adversaries can scale bugs into attacks, industries need a dynamic approach with patching, as in IT
• The time constant is then typically a month
Broad questions include...

• Who will investigate incidents, and to whom will they be reported?
• How do we embed responsible disclosure?
• How do we bring safety engineers and security engineers together?
• Will regulators all need security engineers?
• How do we prevent abusive lock-in? Note the US DMCA exemption to repair tractors …
Policy recommendations included

• Pushing vendors to ensure that products can be patched if need be
• Requiring a secure development lifecycle with vulnerability management (ISO 29174, 30111)?
• Creating a European Security Engineering Agency to support policymakers (now: ENISA)
• Extending the Product Liability Directive to services
• Updating NIS Directive to report breaches and vulnerabilities to safety regulators and users
The punch line

- Phones, laptops: patch them monthly, but make them obsolete quickly so you don’t have to support 100 different models
The punch line

• Phones, laptops: patch them monthly, but make them obsolete quickly so you don’t have to support 100 different models

• Cars, medical devices: we test them to death before release, but don’t connect them to the Internet, and almost never patch
The punch line

• Phones, laptops: patch them monthly, but make them obsolete quickly so you don’t have to support 100 different models

• Cars, medical devices: we test them to death before release, but don’t connect them to the Internet, and almost never patch

• So what happens to support costs now we’re starting to patch cars?
Implications for R&D

• Research topics to support 20-year patching
  Include a more stable and powerful toolchain
• Crypto teaches how complex this can be
• Cars teach: how do we sustain all the test environments?
• Control systems teach: can small changes to the architecture limit what you have to patch?
• Android teaches: how do we motivate OEMs to patch products they no longer sell?
Implications for research and teaching

• Since 2016–7 I’ve been teaching safety and security together in the same course to first-year undergraduates
• We’re starting to look at what we can do to make the tool chain more sustainable
• For example, can we stop the compiler writers being a subversive fifth column?
• Better ways for programmers to communicate and document intent might help
The grand challenge for research

• If the durable goods we’re designing today are still working in 2037 then things must change
• Computer science = managing complexity
• The history goes through high-level languages, then types, then objects, and tools like git, Jenkins, Coverity …
• What else will be needed for sustainable computing once we have software in just about everything?
More ...

• Our papers “Making security sustainable” and “Standardisation and Certification in the Internet of Things” are on my web page
  [http://www.cl.cam.ac.uk/~rja14/](http://www.cl.cam.ac.uk/~rja14/)

• Or see “When Safety and Security Become One” on our blog
  [https://www.lightbluetouchpaper.org](https://www.lightbluetouchpaper.org)
  which also has a couple of videos