

# Standardisation and Certification in the 'Internet of Things'

Éireann Leverett, Richard Clayton  
and Ross Anderson  
Cambridge

# What will the IoT change?

- Privacy made the early running with the smart TV and the Cayla doll – but your phone already hears everything and is full of adware
- Denial-of-service was next with the Mirai botnet – but we already have botnets
- But safety looks like the real pressure point
- Phones and laptops don't kill many people directly; cars and medical devices do...

# How does IoT change safety?

- Eireann Leverett, Richard Clayton and I did a project for the European Commission
- The EU has complex regulatory regimes for the safety of all sorts of devices
- How will these have to change once there's software everywhere?
- We looked specifically at vehicles, medical devices, and electrotechnical equipment
- But the lessons are more widely applicable!

# EU problem statement

- We regulate safety in many industries
- The “Internet of Things” puts computers and communications everywhere
- This creates new safety risks around security
- Indeed, the two are the same in the languages spoken by most EU citizens (sicurezza, seguridad, sûreté, Sicherheit, trygghet...)
- How do we update safety regulation (and safety regulators) to cope?

# Background

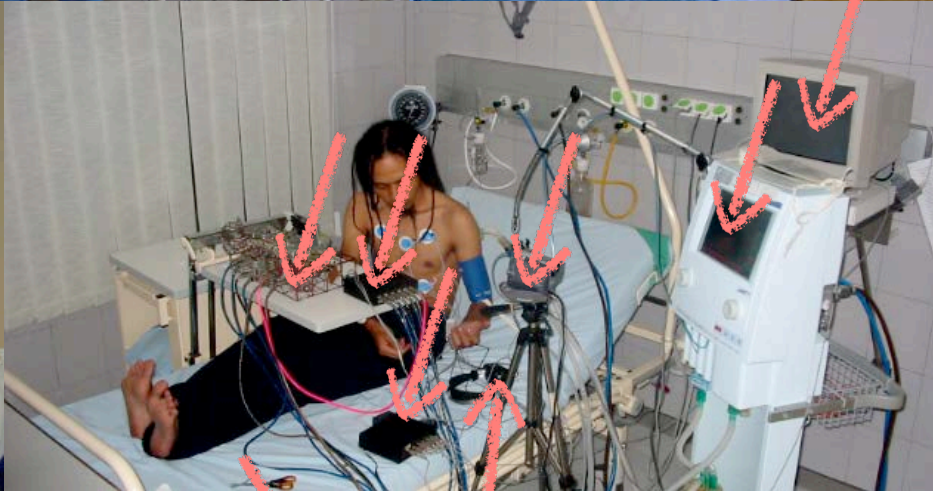
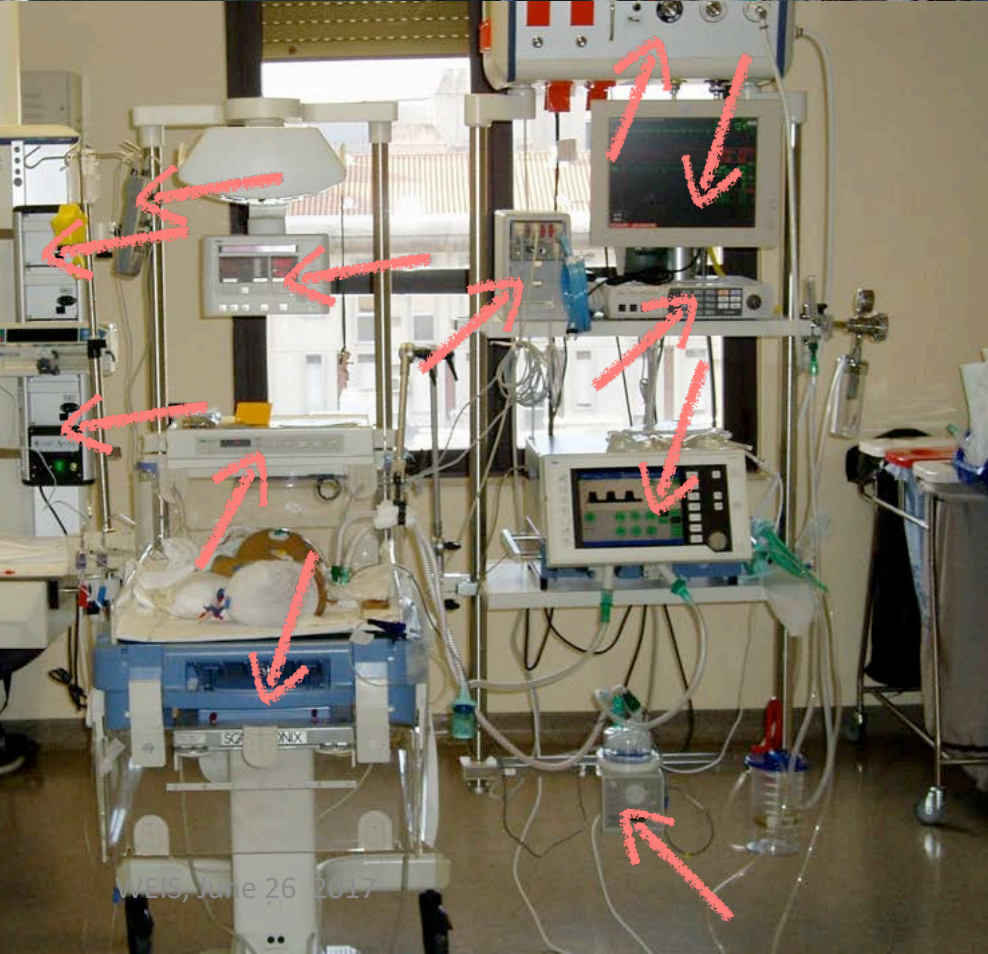
- Markets do safety in some industries (aviation) way better than others
- Cars were dreadful until Nader's 'Unsafe at Any Speed' fired up the public, got insurance industry involvement and led to the NHTSA
- In the EU, we got the Product Liability Directive 85/374/EES, Framework Directive 2007/43/EC on type approval, and much much else
- Some broad principles, plus many detailed rules

# When cars get hacked



# Background (2)

- Traditional car makers moving to autonomy in steps (adaptive cruise control, automatic emergency braking, automatic lane keeping...)
- Challengers like Google, Tesla moving faster
- Tesla has already moved to regular upgrades and the others are racing to follow
- One problem: the test rig (the 'lab car') is big, expensive, and gets recycled for new models
- So how will we patch a 2017 car in 2037?







# Background (3)

- The Medical Device Directives (90/385 EEC, 93/42/EEC, 98/79/EU) are now being revised
- Research by Harold Thimbleby: in the UK, hospital safety usability failures kill about 2000 p.a. (about the same as road accidents)
- Priority: get regulators to do post-approval studies and adverse event reporting
- At present devices are typically approved on paperwork alone
- Many devices needed replacement at Y2K...

# Background (4)

- Usability failures that kill are typically blamed on the nurse (if noticed at all)
- But attacks are very much harder to ignore – a wifi tampering demo in 2015 led the FDA to blacklist the Hospira Symbiq infusion pump
- They balked at recalling 300 similar products
- Software upgrades can break certification!
- Proper safety / security lifecycle is needed

# Background (5)

- ENISA reports that the energy sector has one of the highest rates of attacks on CNI
- UK experience: after alarms about smart meter security, GCHQ engaged with the CNI threat but not the lower-level ones
- Examples of what goes wrong:
  - STS rollover in 2024 affects 400m utility meters
  - Hardening ICS protocols will take 20 years or more

# 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 will 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 ...

# Institutional Players

- Dozens of European regulators (+ hundreds in Member States)
- Standards bodies (UNECE, ETSI, CEN, CENELEC)
- Safety labs (KEMA, EuroNCAP, ...)
- Security labs (CLEFs, Underwriters' Labs, commercial pen testers, ENCS, academics ...)
- Other custodians of the many safety and security standards including NIST, IEEE, IEC
- Other principals, e.g. insurance industry

# Policy recommendations included

- Requiring vendors to self-certify, for their CE mark, 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
- Extending the Product Liability Directive to services
- Updating NIS Directive to report breaches and vulnerabilities to safety regulators and users



# Translating this to engineering

- 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 doing this project I've started 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?
- Need better ways for programmers to communicate and document intent

# Conclusions

- We had a hard look at what's going wrong with the security of durable goods
- Security will be more about safety in future rather than mostly about privacy
- There are many institutional factors, such as liability, regulation and whether the capability to patch resides in one company or several
- But once safety-critical durable goods can be attacked online, it's patch or scrap

 WILEY

# Security Engineering

Ross Anderson

SECOND EDITION

A Guide to Building Dependable  
Distributed Systems