“Yes, your smart toaster really will be spying on you for the government”

–Steve Ranger, ZDNet.com
Identity

- Usually what people think of with “security”
- Today’s focus, a single problem: identity
- Toaster talks to cloud, how does each know the other is the right party?
Secrets

- **tl;dr** We need secrets, on both sides
- toaster needs something adversary doesn’t have, and the cloud needs something, too
- Anyone getting these secrets can pretend to be that entity
TLS and the Web

- Server knows a private key
- Generally, browser doesn’t know anything
- Yes, this is a problem
- Passwords handle this on the web, that doesn’t really work very well
TLS and the Web (cont.)

- To avoid having to log-in to your toaster, it needs its own secrets
- Network secret vs per-device secret
- Network is amazingly common
- If I get that, I can be any toaster I want
The Problem

- Secure communication needs secrets to be stored
- PSK common in IoT, needs fewer resources
- Key install at provision time
- If key leaks, devices can be spoofed
- Vulnerability in code allowing read can reveal these to attacker, can even happen over a network
Secrets

- Threat model shows this is an important area of attack
- “normal” TLS needs to authenticate, that is a secret
- IoT modes such as PSK and bare keys need to store secrets as well
My Four Solutions

- These are arm™ specific (I work for Linaro)
- Somewhat Zephyr specific, but not really
- Aside from “don’t bother”, I want to hear other ideas
Outside Ideas

- TrustZone™
  - Larger CPUs. Common in Arm “A” CPUs
  - Example of OP-TEE implemented with this
  - Not really helpful with “M” CPUs
Outside for ‘m’

- TrustZone™ for Armv8-M
  - Allows a trusted environment to separate secrets
  - But only available on Armv8-M cores
Linux-style

- Privilege Separation, Linux-style userspace
  - Really wants an MMU, even this is heavy with small memory
  - Lots of code on the wrong side of the barrier: network stacks, etc. Doesn’t satisfy least privilege
What Else Can We Do?
The Four

- Tell everyone to only use Armv8-M
- Improve Privilege Separation
- Separate CPUs
- A Hypervisor-ish approach
Only v8m

- Tell people to only use Armv8-M
- Lots of existing chips, and new designs still using Armv7-M
- Likely to always be a cost issue
Privilege Separation

- Move more code to userspace (network)
- Gets complex
- Still lots of code to trust
- Wants an MMU and more RAM
Separate CPU

- Security on its own core (in same SoC)
- Temptation to put more code on it (network)
- Can be a practical solution
Hypervisor-ish

- Use what we have in Armv7-M
- Run Zephyr in user mode
- It makes hypervisor calls when needs privilege ops
- IRQs are complex, need split handling
- mbed uVisor as example (but it does much more)
Recommendations

- Well
- All four
Recommendations

- Support Armv8-M where it can be used, best overall
- Make user/kernel separation better, at least will help larger systems and those with MMUs
- Define a dual-CPU example with security on other CPU
- Prototype the hypervisor, determine if effective/practical
Thank You

#HKG18

HKG18 keynotes and videos on: connect.linaro.org
For further information: www.linaro.org