Software

• Involvement in open source around security and networking (OpenSSL member)
• Interests in security scalability
• Member of Zephyr governance board

Hardware

• “Datapath” architecture for QorIQ and Layerscape SoCs (Networking)
• i.MX apps processors and Kinetis microcontrollers

Focused on new security problems (and solutions) brought on by the emergence of IoT

Based in Québec City, originally from Wellington, New Zealand. (Was not in LoTR)
Agenda

Zephyr
- What, where and why
- Status

IoT security
- Terminology
- Disruption
- Observations
- Where does Zephyr fit into this?
Agenda

Zephyr
- What, where and why
- Status

IoT security
- Terminology
- Disruption
- Observations
- Where does Zephyr fit into this?

See recording of Anas Nashif’s Zephyr talk from Monday
What is Zephyr?

- **Small Footprint RTOS for IoT**
  - As small as 8KB
  - Enables applications code to scale

- **Truly Open Source**
  - Apache 2.0 License
  - Hosted by Linux Foundation
  - Transparent development

- **Cross Architecture**
  - ARM
  - x86
  - ARC
  - Others
Zephyr’s core values

- Modularity
- Security
- Cross Architecture
- Connectivity
- Community Developed
Small OS & RTOS market analysis

Saturated RTOS/Fragmentation
Roll Your Own/No OS
Adoption growth in IoT development
Compromised Devices

Opportunity to build a leading IoT OS
Why Zephyr?

- Strategic Investment
- Best-of-Breed RTOS for IoT
- Permissively Licensed
- Modular
- True Open Source Development and Governance
- Established Code Base
Current platinum members

- NXP
- Linaro
- Intel
- Synopsys
Zephyr project governance

- Governance
- Security
- TSC
- Marketing
- Contributors
Participate!

Benefits of early participation:

- Impact architecture
- Direction
- Marketing / Advocacy
- Decision making
... and scratch that itch
What is “IoT security”?
What is “IoT”? 
What is “IoT”?

- Traditionally-offline “things” → going online
What is “security”? 
Usage

“Add security to the product”

“Secure the edge-node”

“Integrated security, because security is important”
“Add security to the product”
“Secure the edge-node”
“Integrated security, because security is important”

And by security you mean … what exactly?
Does “security” mean...

- Tamper-proof?
- Resistant to side-channel attack?
- Able to perform cryptographic operations fast/efficiently?
- Key-protection and other logical separation?
- Supports secure network protocols?
- Protects content restrictions against misuse?
- Is kept up-to-date through patch updates?
- Reliable/robust in the face of adversarial RF?
- You did some code reviews (this time round)?
Security facets, a less incomplete list

<table>
<thead>
<tr>
<th>Cryptography;</th>
<th>Process and production security</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Software optimization</td>
<td>Compartmentalization/isolation</td>
</tr>
<tr>
<td>• Hardware IP</td>
<td>Digital Rights Management</td>
</tr>
<tr>
<td>• Protocol security, interoperability</td>
<td>IP protection (anti-cloning, ...)</td>
</tr>
<tr>
<td>• Privacy, authentication, non-repudiation</td>
<td>Resistance to side-channel attacks</td>
</tr>
<tr>
<td>Secure non-volatile storage</td>
<td>• Power</td>
</tr>
<tr>
<td>Inline encryption (memory, flash, ...)</td>
<td>• Timing</td>
</tr>
<tr>
<td>Trusted execution (secure boot, ...)</td>
<td>• Electromagnetic emissions</td>
</tr>
<tr>
<td>Key management and protection</td>
<td>Emergency response</td>
</tr>
<tr>
<td>Certification</td>
<td>Security maintenance</td>
</tr>
<tr>
<td>Code quality and review</td>
<td>Attack detection</td>
</tr>
<tr>
<td>Vulnerability analysis</td>
<td>Reliability (quality-of-service, stability, ...)</td>
</tr>
<tr>
<td>Best practice</td>
<td></td>
</tr>
</tbody>
</table>
What is “security”?
What is “security”?

• “Security” on its own can mean almost anything

  “Security” on its own means almost nothing

• It’s almost always context-dependent, in terms of interpretation and importance of those different facets.

• “The minimization of insecurity (or ‘threats’)” ?
What is “IoT security”? 
What is “IoT security”? 

The meeting (perfect storm) of two domains; 

- Device security 
- Network and logical security
What is “IoT security”?

<table>
<thead>
<tr>
<th>Device Security</th>
<th>Network Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure non-volatile storage</td>
<td>Cryptographic s/w and h/w</td>
</tr>
<tr>
<td>Inline encryption (memory, flash, ...)</td>
<td>Protocol security &amp; interoperability</td>
</tr>
<tr>
<td>Trusted execution (secure boot, ...)</td>
<td>Usability and clarity</td>
</tr>
<tr>
<td>Key management and protection</td>
<td>Code quality and review</td>
</tr>
<tr>
<td>Certification</td>
<td>Best practice</td>
</tr>
<tr>
<td>Vulnerability analysis</td>
<td>Emergency response</td>
</tr>
<tr>
<td>Process and production security</td>
<td>Security maintenance</td>
</tr>
<tr>
<td>DRM &amp; IP protection (anti-cloning, ...)</td>
<td>Attack detection</td>
</tr>
<tr>
<td>Resistance to side-channel attacks</td>
<td>Reliability (quality-of-service, stability, ...)</td>
</tr>
</tbody>
</table>
IoT Security – when assumptions collide

**Device security**
- Implementation + certification are static
- Threat model is physical

**Network security**
- Patched early and often, via network
- Threat model is “the network”

**Risk multipliers**
- Widely deployed
- Physical and network accessibility
  - Large attack surface
  - High attack incentive

**Defense de-multipliers**
- Commodity pricing
- Finding and fixing bugs will be hard
  - Minimization of engineering investment
  - Reactive security down, zombies up
Traditional MCU-based engineering

Oriented around **device-security** (if at all);
- Industrial, medical, automotive, …
- **Non-networked**
- Heavily engineered for a **static state of optimal security**
- Once that’s done, ship it!

*(And then move on to something else...)*
Conventional computing complexity

AP-based and even MCU-based systems are more and more complex, resembling server, network, and smartphone systems.
Conventional computing complexity

MPU-based and even MCU-based systems are more and more complex, resembling server, network, and smartphone systems.

*Things will go wrong!* Reactive security (vulnerability handling, incident response) is needed in the microcontroller/IoT ecosystem.
Reactive security for MCUs / IoT

Is Device Lifecycle Management (DLM) the answer?
Reactive security for MCUs / IoT

Is Device Lifecycle Management (DLM) the answer?

Not really, that’s mostly limited to;

• Installing a vendor’s “Root of Trust” (RoT)
• Being locked-in to that vendor’s code/patch-signing services
• The mechanics of deploying updates “Over The Air” (OTA)
Reactive security is well-understood in traditional networked computing;
- Servers
- High-end networking
- Smart-phones
- Desktops
- [...] 

Can we adopt the same methods?
Reactive security for MCUs / IoT

There are some complications with conventional vulnerability-handling (CVE, CPE, etc.)

- The MCU/MPU and its software is often “hardware” to a host
- SoC subsystems often contain firmware too
- One product’s host OS is another product’s subsystem firmware
- CPE isn’t flexible about this hierarchical view
- Multiple vendors involved, supply-chain complexities
Certification for IoT?

Various things have been proposed, but:

- Limit themselves to evaluating the implementation
- Don’t account for the (post-production) process
- Works against responsible code maintenance
- Collapse the supply-chain
Certification for IoT?

Various things have been proposed, but:

- Limit themselves to evaluating the implementation
- Don’t account for the (post-production) process
- Works against responsible code maintenance
- Collapse the supply-chain

And if we certified the software process?
Where does Zephyr fit into this?
Certified/certifiable (audited/auditable, ...)

Merge is usually hard and expensive;
- Upstream doesn’t minimize $\text{diff}(A,A^+)$
- $\text{delta}(A,B)$ doesn’t account for re-certification difficulty

Upstream
- A
- B

Downstream
- Users
- OEMs
- Certified products

Merge
Certified/certifiable (audited/auditable, ...)

Upstream
- Mainline devel
- Stable/LTS
- Hardened tree

Downstream
Hardened “downstream” is coupled to mainline work
- Feedback for security impact of mainline changes
- Creates incentive for a better mainline
- Minimize throttling of mainline development
Where does this happen?

- Governance
- Security
- TSC
- Marketing
- Contributors
Summary

• RTOS upstream to be maintained as production-worthy and current, i.e. reactive security in “real time”.

• Vulnerability handling needs a refresh for “LITE”-type tech.

• Security quality (certifiability, auditability, safety, …) integrated into the project, without bogging down the mainline.

• Drive best-practice for IoT security, practicing what is preached.
Thank you!
Thank you!

(And don’t forget to scratch!)
SECURE CONNECTIONS
FOR A SMARTER WORLD