Security: Quo Vadis?

Hannes Tschofenig
Recent work on communication security makes it harder for MITM attackers to inspect traffic. And there is a lot more encrypted communication in general (thanks to Let’s Encrypt and other efforts).
Security protection needs to happen at the endpoints

Think also about “zero trust networking”.
Layers and layers of isolation
Exception Levels, TrustZone
Adding secure world hypervisor

Figure from Arm® Firmware Framework for Armv8-A
System on Chip Design with secure enclave (Corstone)

Conceptually similar to TPMs and iSIMs but better integrated into the rest of the SoC design.
Confidential Computing: aims to reduce the ability for the owner/operator of a platform to access data and code inside TEE (Intel SGX, AMD SEV-SNP and ARM CCA).

Alternative solution: Privacy-Preserving Computation (e.g. via homomorphic encryption and multi-party computation).

https://confidentialcomputing.io
Isolation requires more code and more complex setup
Reference implementation of low-level software

https://www.trustedfirmware.org
Secure Software Development with
- testing,
- fuzzing,
- static (and potentially dynamic) analysis
- formal methods.
Lots of hardware security mechanisms to deal with programming language issues

Memory encryption, Pointer authentication, Stack limit checking, XN, MPUs, Memory Tagging Extension
Morello: CHERI (Capability Hardware Enhanced RISC Instructions)
Large number of guidance documents being published

How to make sure that vendors follow the guidance? Certification
What are the attacks we are still seeing?
• Stupid mistakes
• Side channel attacks
• Fault injection attacks
• Firmware rootkits
• Social engineering attacks
• Ransomware
Despite all these security technologies, why do we still have attacks?

Or: What should we do differently (better)?
My list

- Too many unfinished libraries
- Complexity causes problems for developers
- Technology deployment takes a long time
- Hacking is more rewarding than securing