HASPOC - a secure platform for embedded systems

OUR VISION: HASPOC, a trusted, cost and resource efficient virtualized COTS (ARMv8) platform with formally proven and Common Criteria certified security properties.

The HASPOC platform enables secure and trusted solutions for embedded systems by isolating critical services through virtualization. This reduces the trusted computing base drastically, a prerequisite for high assurance. Furthermore, the HASPOC platform helps to reduce costs by securely sharing hardware among multiple services. The technology is – in addition to specific security products like crypto equipment, secure mobile phones and firewalls – applicable in a wide range of areas such as critical infrastructures, SCADA systems, mobile communication networks, vehicular, avionics and medical systems, cloud application platforms and in the Internet of Things.

The HASPOC ARMv8 Hypervisor

The HASPOC ARMv8 hypervisor is a virtualization platform that with high assurance provides strict isolation and controlled communication between guest operating systems. Key characteristics are:

- Full multi-core system virtualization – guests can be hosted without any modifications.
- Peripheral access with strict isolation – a guest can be given exclusive access to a peripheral.
- Multi-core with strict separation – a guest can execute on one or more cores but a core cannot be shared between two guests.
- Static configuration after boot – guest configurations and parameters cannot be changed after hypervisor initialization.
- A framework for secure hypervisor services – services include high performance inter-guest communication, multi-core power management and controlled shared access to resources and peripherals.
- Small size and footprint – the hypervisor binary is no larger than 64KB and consists of no more than 10.000 lines of code
- Example configurations are provided for various Linux flavors (Debian, Ubuntu, Android) and bare metal guests.

Secure Boot

The secure boot solution handles initialization and configuration of the HW platform in addition to cryptographic verification of firmware integrity (including hypervisor and guest firmware) to ensure a secure start of the HASPOC hypervisor.

The secure Boot is based on ARM Trusted Firmware and adapted for Common Criteria and formal verification. It is divided into target specific and generic portable modules. It supports the Hikey 96board, the Juno reference hardware and the Fast Virtual Platform. Key characteristics are:

- Simplified and smaller than ARM Trusted Firmware
- Trust anchoring conforms to ARM Trusted Firmware model
- Compliant with ARM Trusted Firmware Runtime Services
- Minimalistic cryptographic module RSA2048 and SHA256
- Modular design allowing extensive testing
- Extensible boot format supports large objects
- ROM resident binary 60 kbyte

Formal verification

In an ongoing effort the security of HASPOC is being formally verified which also reduces its trusted computing base. The following is covered by this work:

- Integrity and confidentiality with controlled communication are formally verified; in particular that guests on a HASPOC platform execute as if running on separate machines in a distributed setting (“bisimulation”)
- Implicitly verified properties are
  - Secure code: lack of overflows, null pointer dereferences etc.
  - Functional correctness of boot and hypervisor handlers
  - Correct hardware configuration
  - Separation properties of the underlying hardware
- Verification is performed on binary code level
- Detailed models for ARMv8 architecture + proof machinery have been developed

Prepared for Common Criteria (CC) evaluation

The Common Criteria (CC) assurance methodology (ISO 15408) is applied to the platform in order to meet the assurance requirements for an EAL6 evaluation of products using the platform. This is formally described by the following documents, as mandated by the Common Criteria standard:

- A Security Target (ST) which defines the HASPOC platform as the Target of Evaluation (TOE) and describes the security problem solved by the platform. The ST is conformant to Common Criteria version 3.1 revision 4.
- A Functional Specification (FSP) which gives a semi-formal specification of the external interfaces of the TOE.
- A TOE design (TDS) document which describes the HASPOC platform modular design including the subsystems and the modules of the TOE, the security functionalities provided and the interaction between them.
- A Security Policy Model (SPM) document which describes a formal model of the security policies enforced by the HASPOC Platform.
- An Architecture (ARC) document which describes the security functionalities provided by the HASPOC platform.
- Test guidelines which give guidance on how a product using the HASPOC platform can be tested in order to fulfill the CC requirements at EAL6.

Benchmarks

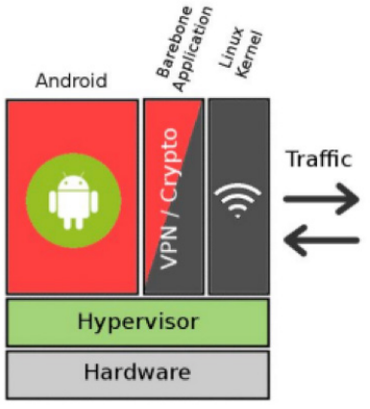
The Secure Boot start time including cryptographic signature verification is comparable to that of ARM Trusted Firmware without signature verification. In our measurements Secure Boot and ATF require 4.866s and 4.941s respectively to load a 15MByte binary. For the hypervisor we have used a number of different methods and benchmark suites (*CoreMark, NBench, SciMark, LINPACK, dhrystone, perf, memtester and kselftest, Caliper, lmbench, hackbench and tinymembench, dbench, fio, iperf, nperf, cachebench and ARM-Memspeed*) to measure the following:

- Memory performance overhead of a virtual guest is around 2% of a native guest.
- Latency performance is generally within 3% of bare metal performance. A notable exception to this is latency for user-space process context switch (lmbench_ctx) which has much higher latency due to a constant rescheduling penalty of +30 µs.
- Network performance shows up to 15% lower throughput than bare metal. The performance degradation is mostly due to the rescheduling issue explained above.
- CPU performance overhead is around 0.5% of a native guest.
- IO performance overhead is within 2% of a native guest, performing notably better in some tests.
- A Linux virtual Ethernet driver implemented using the hypervisor inter-guest communication service was measured to have a 750 Mbps TCP bandwidth.
- The hypervisor scores slightly better (62.0%) compared to native guests (58.9%) on Linux Test Project.

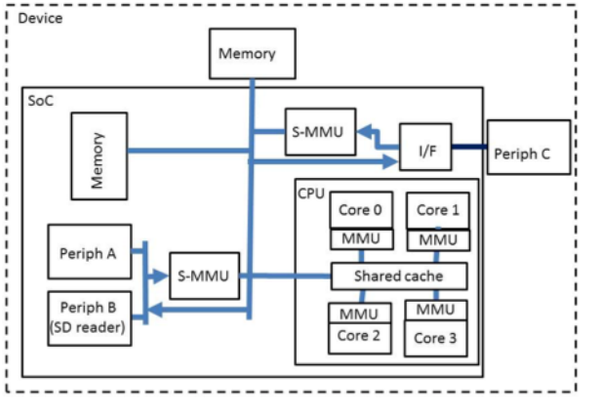
All benchmarks have been measured using a HiKey board and Ubuntu Server 15.04.

A demo application with HASPOC inside

Often sensitive resources of an organization are made available to members' smartphones and tablets via VPN. The security of such tunneling traditionally depends on the security of the mobile device's OS. HASPOC allows to reduce the amount of software to be trusted. As shown in the figure, the VPN service and its credentials can be placed in a separate guest. A specific network guest is responsible for all traffic and can only communicate with the rest of the system via the VPN guest. Even if the commodity OS and the network guest are compromised, it is still guaranteed that VPN credentials are protected and that all information leaving the device is encrypted. With a bandwidth of 140 Mbit/s between commodity OS and network guest, the user experience is comparable to using Android directly.



HASPOC compliant HW platforms



The HASPOC solution is suitable for platforms that comply with the following requirements illustrated in the diagram:

- ARMv8-A architectures with multiple cores and two-stage MMU
- System MMUs distributed in accordance with the desired partitioning
- ARMv8 General Interrupt Controller v2 or upwards
- For trust anchoring: write-protected root key hash (e.g. in eFuse) and ROM with HASPOC boot code
- Examples for supported platforms: Juno and HiKey boards and various development systems from Qualcomm and Nvidia

Open source + support

The HASPOC Hypervisor will be made available as Open Source under the terms and conditions of Apache License 2.0 (available at: http://www.apache.org/licenses/LICENSE-2.0.html).

The HASPOC Secure Boot will be made available as Open Source under the terms and conditions of the GNU General Public License v.2. (available at: http://www.gnu.org/licenses/gpl-2.0.html).

The code and documentation can be found on https://bitbucket.org/sicssec/

If you plan to deploy the HASPOC platform on your systems and possibly want to adapt it to your needs, we can help you with advice and pointers, collaborate in a new project, or just share experiences in a seminar.

Adaptations for a specific hardware platform

Since this type of low-level software is closely tied to a specific hardware, care has been taken to minimize porting effort for new target platforms. The secure boot has been divided into two parts, one of which is not hardware dependent. The hypervisor has a framework for target-specific initialization and drivers and allows many target properties such as CPU topology to be defined as a configuration file.

Application/guest development and launch

The hypervisor provides a tool for creating a firmware bundle which contains guest firmware and target configuration. The guest firmware is user defined and can contain multiple OS images, secondary boot loaders, file systems and so on.

The secure boot provides tools to cryptographically sign the bundle and the hypervisor binary. During boot, the secure boot will verify the integrity of the two.

Guests run fully virtualized, but adaptations might be needed to improve performance or support some peripherals and services. For example, the inter-guest communication service requires the use of IGC kernel drivers.

The HASPOC project

HASPOC (High Assurance Security Products On COTS platforms) is a project in Vinnova's (Swedish Governmental Agency for Innovation Systems) Challenge Driven Innovation program, see Vinnova.se, and was partially funded by Vinnova.

Partners in the project are the Security Lab at SICS Swedish ICT AB, the Security Research Area at Ericsson Research, Sectra Communications, Tutus Data, T2 Data, atsec, and the Department of Theoretical Computer Science at KTH.

For further information and our introductory video see the project homepage https://haspoc.sics.se/.

For additional information, please contact:
Rolf Blom, SICS Swedish ICT, rolfb@sics.se

SWEDISH ICT SICS

PART OF RI SE