Kaspersky Secure Hypervisor: Zero tolerance to insecurity

Security is the key feature in the adoption and success of embedded systems in a world where professional hackers and APTs is the harsh everyday reality. Threats vary, from attacks on exposed interfaces to undocumented or aggressive behavior of peripheral equipment (e.g. PCI, USB). After a successful attack, an intruder can go further, e.g. by exploiting operating system vulnerabilities, potentially threatening critical processes. Despite the vulnerabilities and large attack surface of general-purpose systems, vendors prefer popular platforms because of the widespread availability of software.

Virtualization technology gives an opportunity to harden system security, while retaining the ability to reuse an existing code base.

Introduction

Kaspersky Secure Hypervisor is a virtualization technology developed by the Future Technologies department at Kaspersky Lab. It is a type-2 hypervisor that runs on the KasperskyOS microkernel and utilizes its security capabilities, turning KasperskyOS into a hypervisor solution.

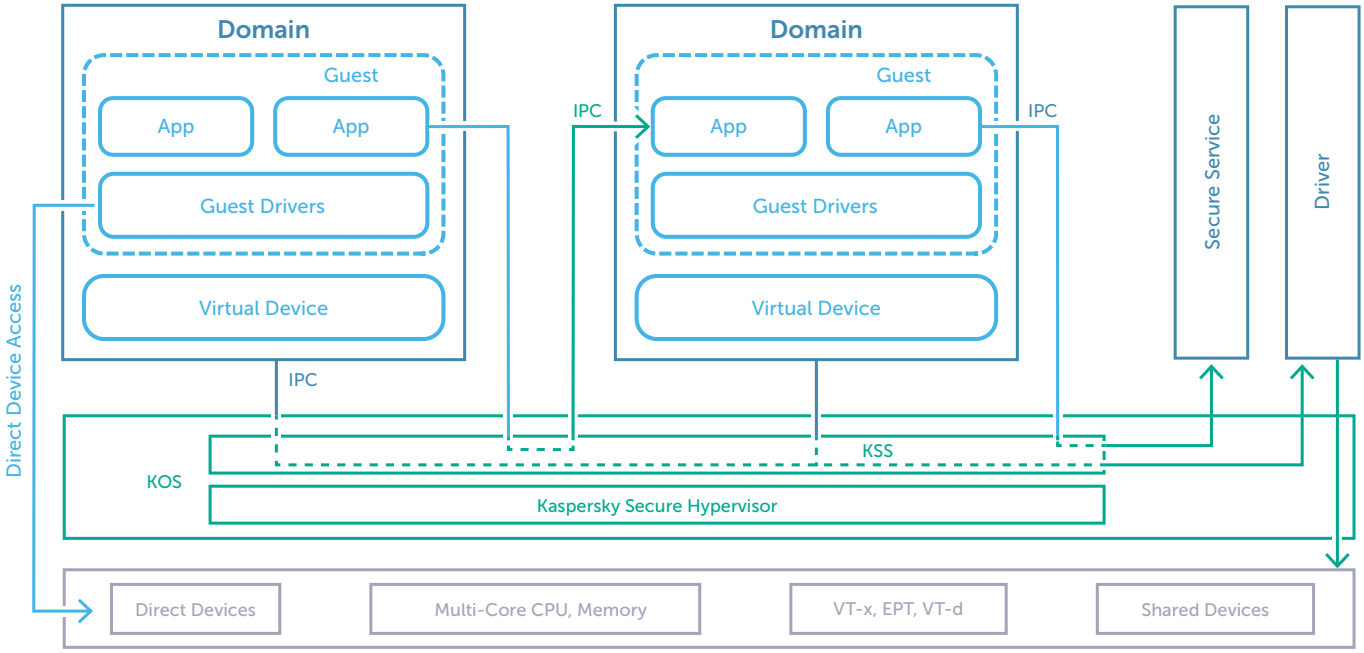
The goal of the solution is to make it possible to run multiple virtual machines (guest operating systems) on a single physical machine, distributing physical resources among guest systems. Normally, virtualization is supported by modern CPU capabilities that give guest OSs performance close to that on a dedicated physical machine.

The main benefit of a virtualization solution is the separation of possibly untrusted guest operating systems from each other and from critical services collocated on the same physical machine, reducing the attack surface and minimizing the possible impact of vulnerabilities exploitation. The hypervisor itself is strongly protected from actions of guest OSs in such a way that malicious activities of a guest system cannot damage the critical services and the hypervisor itself. We see Kaspersky Secure Hypervisor as a trusted framework for IoT, automotive, healthcare, and industrial automation that can fulfill Kaspersky Lab's mission of creating a safe and secure world Kaspersky Secure Hypervisor contains two software components: one running in a privileged kernel mode, and the other running in a user mode. The privileged kernel component is responsible for the management of resources (e.g. memory, CPU) and provides access to I/O devices. User-mode components of Kaspersky Secure Hypervisor include common host user-mode device drivers and a special guest driver, called **KL secdev**, that provides communication channels between domains or between a domain and the hypervisor itself.

Kaspersky Secure Hypervisor uses hardware virtualization features (Intel VT-x) to create virtual environments (**domains**), which share common CPU and memory. If present, Intel VT-d hardware virtualization features can be used to pass through PCI devices (such as a video adapter, network adapter, hard disk controller, USB) to guest OSs. This technique improves the performance of these devices but makes sharing impossible. See the list of emulated and passed-through devices below.

If sharing of devices is required, we use a **user-space device emulation** technique. The idea is to run a KasperskyOS user-space driver with direct access to a device that needs to be shared. The driver implements **device emulation**, i.e. it provides an interface for guest OSs that can be used to access

Architecture



a device; with guest OSs not knowing that the device is virtual. A security benefit of device emulation is that the hypervisor can intercept all the transactions between a guest OS and the device (e.g. network card, SATA controller), and implement additional security measures (e.g. traffic filtering, encryption), while a guest OS cannot bypass these measures. Device emulation can also be used when hardware virtualization features (such as Intel VT-d) are not available.

If a solution requires a guest OS to communicate with another guest OS or with native KasperskyOS applications, the guest OS can use **KL secdev**, a special hypervisor-aware guest driver (**paravirtualized** driver) included in the Kaspersky Secure Hypervisor delivery set. All communications are mediated by a flexible access control system, namely Kaspersky Security System (KSS) – another technology developed by the Future Technologies department at Kaspersky Lab.

Use Cases

In-Vehicle Infotainment

In this architecture, two domains are used: one for mission – critical software (e.g. vehicle control and network software) and another for non-critical infotainment software (media, connectivity, voice features, user interface). Separate virtual environments ensure the stability of mission-critical software, whatever actions a user takes. With the device emulation technique, shared hardware (such as network card, GSM or WiFi modules) can be used by domains, reducing the hardware costs. This architecture also makes it possible to keep the infotainment software up to date without affecting critical components during software updates.

Industrial Security Systems

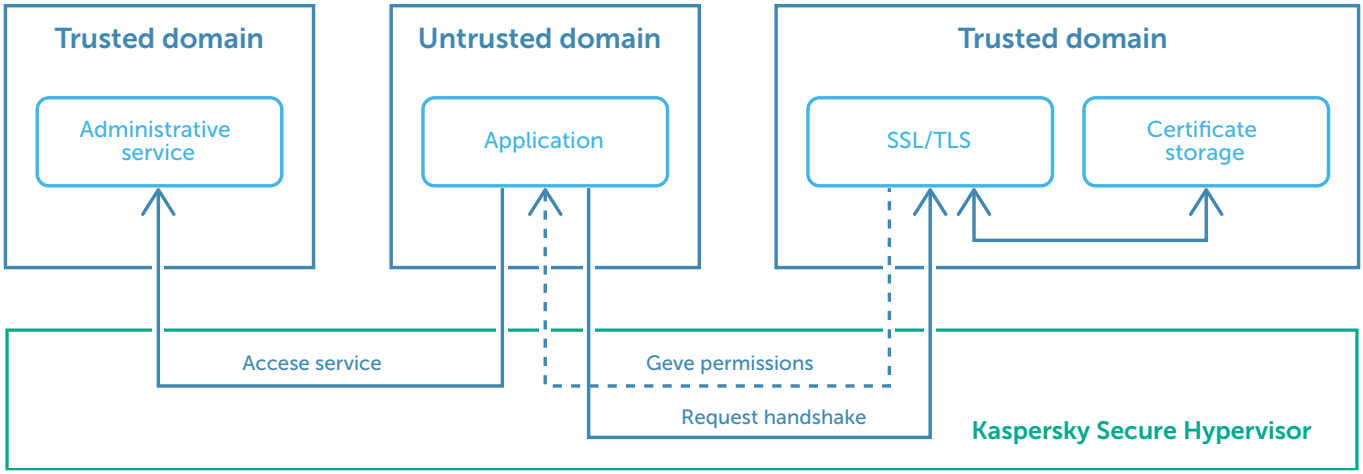
In this architecture, two or more domains are used to separate industrial software, communication software, databases, and user interface.

Mobile Devices

A hypervisor solution can create separate domains for corporate data, personal data, and critical applications (e.g. broadband software stack, security services, storage for certificates and credit cards).

Secure Storage For Certificates And Keys

In this architecture, certificate storage and encryption services are kept in a separate trusted domain. Guest OS applications run in another domain, and get access to encryption services by Kaspersky Secure Hypervisor communication channels. With proper authentication, the trusted components can give additional privileged permissions to guest OS applications (e.g. permissions to administrative services). Even if guest OS applications have

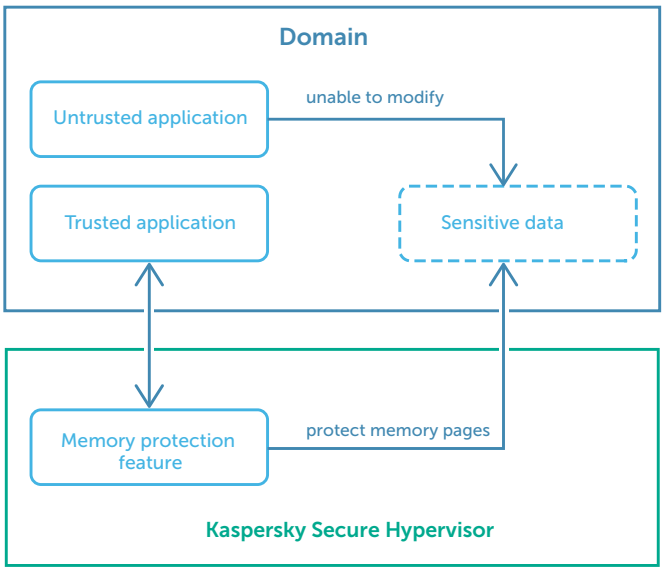


been hacked, they cannot get access to keys or escalate their privilege due to security policy enforcement and domain separation guaranteed by the hypervisor.

Network Analysis And Filtering

In this architecture, all network communications between guest OS applications and the external world are filtered transparently (i.e. with no modification to guest OS applications and with guest OS applications unaware of filtering). If needed, a traffic inspection can be implemented, remaining invisible to potential attacks from guest OS applications. Even if guest OS applications have been compromised, they cannot bypass filtering and send data to a remote party.

Protection Of Sensitive Guest OS Data



Kaspersky Secure Hypervisor is capable of protecting a guest OS's sensitive data from modifications via a memory protection mechanism. Memory protection is achieved by setting appropriate permissions to guest physical pages. In a typical scenario, a guest OS calls Kaspersky Secure Hypervisor to protect the guest OS's sensitive data before running an untrusted application. Examples of protectable data include guest OS kernel code, guest security services, and configurations.

Features

1. **Proprietary solution from Kaspersky Lab**
   Kaspersky Secure Hypervisor is a proprietary solution fully supported by Kaspersky Lab. The development process is based on best practices with systematic testing and verification.

2. **Strong isolation and mediated communications**
   Kaspersky Secure Hypervisor utilizes KasperskyOS features, providing the means to run multiple guest operating systems and KasperskyOS native applications on a single machine. Isolation between domains is guaranteed by the kernel component of Kaspersky Secure Hypervisor. All communications between domains and between a domain and the kernel are mediated by Kaspersky Security System according to a predefined security policy. Even if an attacker performs a virtual machine escape, further actions are limited by a security policy.

3. **Flexible access control**
   Kaspersky Security System is grounded on an attribute-based model and supports a wide range of policies (e.g. object capabilities, flow control, Type Enforcement and multilevel security). Its flexible and extendable nature makes it possible to develop custom domain-specific policies relevant to an application area.

4. **Resource management for guest OSs**
   Kaspersky Secure Hypervisor restricts the amount of resources (such as memory, or physical device access) available to guest systems to protect the whole environment from possible resource exhaustion attacks coming from guest OSs. Potentially dangerous external devices can be restricted, so that erroneous or malicious hardware is unable to access the memory of guest operating systems or the hypervisor.

Supported Os And Hardware

- **Host OS.** Kaspersky Secure Hypervisor is a type-2 hypervisor provided with KasperskyOS as a host system.
- **Platforms.** Intel x86 or x64 with support for VT-x and (optionally) VT-d technology. Support for ARM is in progress.
- **Guest OSs.** Unmodified Linux-based distributions such as Ubuntu and CentOS can be used as guest operating systems, on both x86 and x64 variants. Support for other guest environments (primarily, Windows) is in progress.
- **Emulated devices.** x86 legacy (PIC, PIT), PCI bus, NE2000, IDE/SATA controller, UART (COM port).
- **Tested pass-through devices.** USB controller, SATA controller, PCI Ethernet, Radeon/nVIDIA video cards, legacy IDE controller.

5. Ability to integrate with secure boot system

Kaspersky Secure Hypervisor includes features to guarantee integrity of the hypervisor and guest OSs.

6. Small trusted computing base

When used with KasperskyOS as a host OS, Kaspersky Secure Hypervisor benefits from the KasperskyOS microkernel design, providing a small verifiable **trusted computing base (TCB)***. All device drivers are run in a host user mode, further reducing the risk of a hypervisor being damaged or hijacked.

---

* Trusted computing base (TCB) is the set of all components (hardware, firmware, and software) critical to overall security of a solution. Small TCB facilitates exhaustive testing and verification.

www.kaspersky.com

Find out more at **os.kaspersky.com**
All about Internet security: **www.securelist.com**