KasperskyOS

Cyber Immune operating system for industries with high information security requirements

KasperskyOS brings Kaspersky's expertise in information security and best practices for developing inherently secure specialized operating systems to the general-purpose OS market

KasperskyOS allows you to create an IT system with the highest level of security guarantees. Most types of cyberattacks on such a system cannot affect its core functions.

Kaspersky has long been working to solve the problem of how to create a trusted IT system from untrusted components. The results are implemented in the Kaspersky Cyber Immunity® architectural approach and KasperskyOS operating system, designed to develop products for industries with increased security, reliability and predictability requirements.

The aim of KasperskyOS is to protect IT systems from malicious code and exploitation of vulnerabilities. These threats can lead to the loss or leakage of sensitive data, performance degradation, or a denial of service. In addition, KasperskyOS reduces the risks associated with code errors and accidental or deliberate damage.

Architecture features

KasperskyOS is based on a combination of different security approaches.

Due to its distinctive architecture, KasperskyOS creates an environment in which it is safe to run untrusted and potentially vulnerable programs.

For most operating systems, security is achieved by sharing rights and controlling access to system resources. To this KasperskyOS adds the ability to configure and guarantee compliance with the security properties required for each specific task.

Microkernel. Minimal amount of code lines necessary to make kernel mechanisms work, providing strict control over the OS code quality.

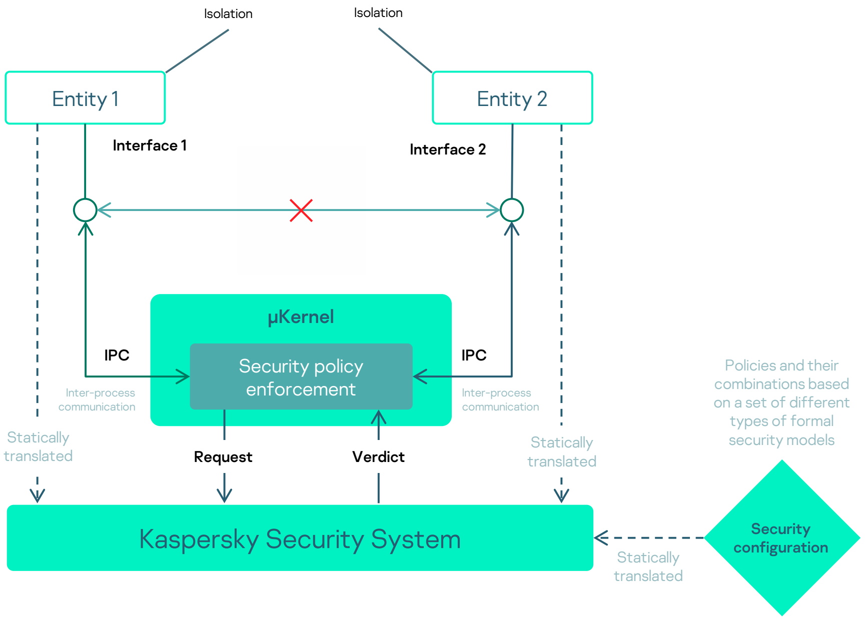
Guaranteed isolation. The system guarantees isolation of security domains and separation of security features from functional components.

Unified inter-process communication (IPC) mechanism. The microkernel provides a single IPC mechanism.

Strictly defined typed interfaces. Interfaces must be statically defined for each application or driver.

Kaspersky Security System. The KSS subsystem checks the validity of all IPC messages against the interface definitions and controls communication between different parts of the system, preventing exploitation of vulnerabilities by attackers.

Static security configuration. All processes and their permitted types of communication are preconfigured and checked before functioning.


Isolation
Isolation
Entity 1
Entity 2
Interface 1
Interface 2
μKernel
IPC
IPC
Security policy enforcement
Inter-process communication
Inter-process communication
Statically translated
Statically translated
Request
Verdict
Policies and their combinations based on a set of different types of formal security models
Kaspersky Security System
Statically translated
Security configuration

The main security principles of KasperskyOS

System Requirements

CPU requirements:
Memory Management Unit (MMU);
IOMMU (SDMA for ARM) is highly recommended for reliable isolation of hardware resources.

Supported architectures: x86, x86_64, ARMv5, ARMv7, ARMv8 & MIPS32.

Tested hardware platforms:

- Intel Generic & Atom CPUs,
- NXP i.MX6 (Solo, Duo & Quad),
- NXP i.MX27, TI Sitara AM335x,
- TI Sitara AM43xx, HiSilicon Kirin620,
- MIPS24k.

The minimum amount of RAM depends on the solution. The recommended amount of RAM is 128 MB.

Patents:
US 7386885 B1, US 7730535 B1,
US 8370918 B1, EP 2575318 A1,
US 8522008 B2, US 20130333018 A1,
US 8381282 B1, EP 2575317 A1, US 8370922 B1,
EP 2575319 A1, US 9015797 B1,
DE 202014104595 U1.

Advantages

Proprietary microkernel

KasperskyOS is not a modification of an existing operating system; it is based on a proprietary microkernel that only allows a specific type of communication.

Multi-level compatibility

Because the system is for the most part POSIX-compatible, using a native API ensures secure application behavior. The developer can choose how to maintain a balance between program code compatibility and security.

Mandatory identification and labeling

All KasperskyOS applications have their own secure configuration, without which they cannot be installed. Hardware and application-level resources (files, databases, network ports, etc.) are labeled with the appropriate security attributes. It is also impossible to access a resource that does not have a security label.

Modular architecture

The modular approach to system architecture minimizes the size of the trusted code base and enables each individual solution to be built on a case-by-case basis.

Secure application architecture

The application architecture is based on a component model that makes solution development easier and more efficient.

Easy-to-configure policies

A simple configuration language makes it easy to set up rules for interprocess communication and access control.

Reduced attack surface

Splitting applications into security domains and full control over interprocess communications enables the safe use of potentially vulnerable and/or untrusted applications.

Verifiability

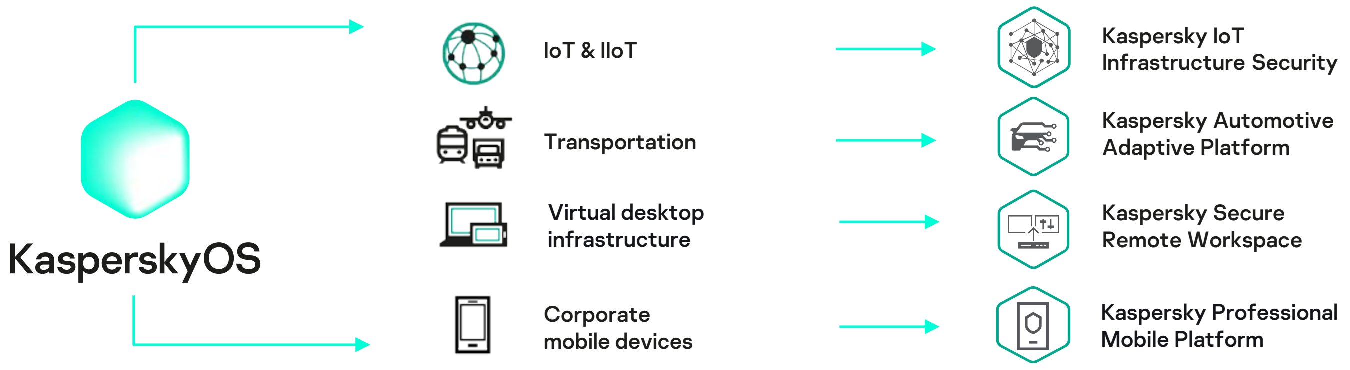
Strict adherence to security principles in the design and implementation of the system allows you to verify the security of all solutions based on KasperskyOS.

Secure by design system

KasperskyOS was designed and developed to be inherently secure. It remains safe throughout its entire life cycle.

Areas of application

KasperskyOS is used in industries where IT systems are subject to higher cybersecurity, reliability and predictability requirements.



KasperskyOS

IoT & IIoT → Kaspersky IoT Infrastructure Security

Transportation → Kaspersky Automotive Adaptive Platform

Virtual desktop infrastructure → Kaspersky Secure Remote Workspace

Corporate mobile devices → Kaspersky Professional Mobile Platform

KasperskyOS

Find out more os.kaspersky.com

www.kaspersky.com