KasperskyOS

Whitepaper

MOTIVATION

Many modern computer systems – including systems that form part of our critical infrastructure, the Internet of Things and Machine to Machine communication – run a daily gauntlet of numerous and varied cyber-threats. A lot of these systems have specific security objectives related to their features and special aspects of their use.

To achieve these objectives the system needs to implement an appropriate security policy – and that policy needs to be firmly enforced. Indeed, without that enforcement, security becomes a drain on resources as it attempts to plug multiple holes. Security experts agree that strict enforcement of proper security policies plays the key role in securing the system. But if the issue is just about policy why are systems still insecure?

Common-purpose OS are incapable of conforming to the precise security policies of each critical application because common-purpose solutions are flexible and versatile but not intended to be secure by nature. A special-purpose system is likely to implement its specified policy with some guarantees. But it is both difficult and expensive to assure full implementation of each specific set of security requirements – and the process needs to begin from the ground up in each individual system.

There is a gap in the market for products which implement diverse security policies for systems requiring security assurance. KasperskyOS aims to close this gap by providing a high-assurance secure platform which is capable of enforcing any given policy for different critical applications.

PURPOSE

KasperskyOS aims to protect software and data systems from the consequences of the intrusion of malicious code, viruses and hacker attacks. These can provoke harmful behavior in any part of the system, potentially resulting in loss or leakage of sensitive data, reduced performance and denial of service.

In addition it reduces the risk of harm caused by program bugs, unintentional mistakes or pre-meditated abuse.

CONCEPTS

Unified low-level interprocess communication. The microkernel of the operating system implements the low-level IPC as the only way for processes to interact.

Complete mediation in access control. Interprocess communications are controlled according to the desired security policy and this control cannot be bypassed by any means.

Typed Communications. Communications are categorized by type and properly handled by security runtime to fill the semantic gap between low-level IPC and high-level security policies.

No anonymous messaging. All processes and their types of permitted communications must be configured before execution, any other interaction is denied by default.

Separated security server. The enforcement of a specific policy is not a security kernel concern but is the sole task of an independent engine which returns a verdict on whether access is permitted or not.

FEATURES

Proprietary microkernel and independent security engine. KasperskyOS is based on a reliable microkernel that implements the only way of communicating. This lightweight microkernel can be implemented on various platforms. At the same time the loosely coupled security engine makes it possible to replace the in-house microkernel with another kernel if necessary.

Multi-level compatibility. While the system is kept mostly POSIX-compatible, the use of a native API further guarantees the secure behavior of applications. The developer can choose how to keep a proper balance between program code compatibility and security.

Security domain separation: KasperskyOS efficiently separates security domains – confined groups of applications with a restricted influence on each other. This does not preclude the possibility of interdomain communication, if explicitly allowed.

Mandatory identification and labeling. All applications in KasperskyOS are accompanied by their security configuration. Nobody can install an application without installing its relevant behavior configuration. Hardware and application-level resources (files, databases, network ports, etc.) are labelled with appropriate security attributes. It is impossible to access a resource that doesn't have a security label.

Diverse policies enforcement. An independent security engine can enforce the policy that best matches the identified security objectives. The security policy can also be individually configured for every application in the system.

Tamperproof configuration and services. The security configuration is stored in protected memory and can be accessed by trusted services only, which are also restricted in their communications.

ADVANTAGES

Initially secure system. KasperskyOS is designed with security in mind and remains secure during its whole lifecycle.

Modular design. A modular approach to system design minimizes the footprint of the trusted base and makes it possible to build each individual solution on a case-by-case basis.

Secure architecture of applications. Application design is based on a component model that makes secure development easy and elegant.

Easy-to-configure policies. IPC typing and simple configuration language help to easily define the rules of interprocess communication and access control.

Verifiability. Strict adherence to security concepts in system design and implementation makes it possible to verify the security of all solutions based on KasperskyOS.

SUPPORTED ARCHITECTURES

- **KasperskyOS for x86/x64**
  CPU: Pentium II or higher
  RAM: 8Mb or more
  Ethernet: Realtek RTL8139, Intel i82580
- **KasperskyOS for ARM**
  CPU: ARMv7 or higher
  RAM: 8Mb or more

© 2014 Kaspersky Lab ZAO. All rights reserved. Registered trademarks and service marks are the property of their respective owners.