KasperskyOS®

Secure OS for the Internet of Things

Most IoT appliances are based on common operating systems that are incapable of addressing specialized security requirements

Technical requirements

- POSIX API (~98% API) compatible
- Intel x86, x64 и ARM (v6, v7, v8)

Patents

The technologies that form the basis of KasperskyOS and Kaspersky Security System are covered by a set of 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.

Introduction

The Internet of things is a new paradigm that is changing the world before our very eyes. It could make our world safer, improve our health, save us time and money, reduce waste and add a new dimension to production control and life in general.

The IoT concept encompasses a huge variety of appliances, gadgets, technologies, software and communication protocols. This heterogeneous environment generates lots of security risks that could seriously hamper any aspect of our life related to the IoT.

Our aim is to make the most of the IoT's undoubted benefits, while minimizing the associated risks.

Most IoT appliances are based on common operating systems that are incapable of addressing specialized security requirements.

These systems tend to be over-featured, with functionality that isn't necessary for the connected device. At the same time, almost no attention is paid to patching the multiple vulnerabilities caused by poor design, bad implementation and improper use of operating systems in these devices.

This deep integration of interconnected devices embedded into our daily lives means security is of paramount importance. Because there are so many embedded devices, it is wasteful and impractical to apply add-on security controls to each IoT device. Security needs to be in-built, fitting the environment and supporting system functionality without any restrictions.

The key pillar of built-in security is a proper security policy. Traditional 'office use' security policies and policies for IoT devices are quite different. Instead of focusing on unauthorized access to information, data corruption or DDoS attacks, IoT policies should mitigate so-called thing-level attacks. These attacks exploit the exposure of the system to physical hazards, or result in physical consequences.

Due to the diversity of IoT applications, security policy enforcement mechanisms must be as adaptable as possible. Security researchers have already come up with specific security models for the IoT – thing-based access control (similar to role-based access control), capability-based approaches, etc. The way security policies are defined is also important: they should be clear and simple, but expressive enough to make rules without flaws and omissions.

At the same time, security mechanisms should not weaken existing safety measures, hamper system functionality or significantly reduce system, application or device performance.

Purpose

To address the issue of cyber security for IoT devices, while minimizing the time required to develop security features, we offer KasperskyOS, a secure operating system based on an architecture designed to ensure software is executed securely, including non-secure applications. In addition, KasperskyOS provides protection in the event of random software errors and improper user actions.

Advantages

Inherent security. KasperskyOS is secure by design and we intend to keep it that way by using the best practices of software development.

Versatile modular architecture. Building the system based on loosely coupled modules helps minimize the amount of trusted code and tailor each solution to specific needs.

Well-designed applications. The component-based approach to creating secure applications makes developing them relatively easy and convenient, helping reduce the amount of time needed to take new products to market.

Flexible security configuration. Well-designed configuration tools make it easy to create declarative rule definitions and combinations of rules to control interactions in the system.

Separation of application features from security functions. The security architecture is designed to separate security functions from application business logic, making both configuring security policies and developing applications easier.

Full-fledged security for attached devices. KasperskyOS is a reliable platform for embedded systems that have special cybersecurity requirements.

KasperskyOS is a secure operating system based on an architecture designed to ensure software is executed securely, including non-secure applications. In addition, KasperskyOS provides protection in the event of random software errors and improper user actions

Features

One of the most important KasperskyOS components is Kaspersky Security System (KSS) – a versatile security engine making it possible to define and check custom security policy for IoT applications.

Kaspersky Security System is based on the principle of isolating the security component from the information system's functional components. This ensures the system's secure operation regardless of the way its functional components are implemented, making it possible to build trusted systems using untrusted components. As a result, the security policy can be modified without changing any functional components. KSS supports the combining of different security models, including the ability to use basic and specialized policies at the same time.

KSS is about more than just malware protection; it also prevents common violations of security rules. The solution adds security without harming production safety. Kaspersky Security System is embedded in the firmware of IoT devices, computing security verdicts that are defined and configured by the manufacturer.

There are other additional security features that can be provided together with KasperskyOS for IoT:

Trusted Channel

This is a set of components that can be used to organize a secure communication channel between a device and a remote party.

The technology is based on the TLS protocol, a mature standard protocol providing security for communications. Multiple implementations are available (including open source) from various vendors.

However, it is often the case that TLS-based solutions incorporate numerous functions (e.g. Linux process) into one domain:

- TLS implementation
- Connection management
- Application-specific protocol processing (e.g. HTTP)
- And even more high level logic

It means all these functions must be considered as trusted: compromising any of them results in a whole system being compromised.

Trusted Channel's main objective is to minimize the size of trusted code by separating secure connection, authorization and remote request processing. In KasperskyOS, a secure connection is made with TLS in a separate domain (entity), as well as authorization of the connection. Neither TLS nor authorization performs any application-specific message processing.

In this architecture, network modules, connection management and any application-specific data processing (e.g. HTTP parsing) are treated as untrusted. The only trusted components are TLS and authorization.

Secure Storage

Secure Storage is a key-value database with a simple interface, suitable for storing important configuration parameters.

Every parameter in the database is associated with its own security attributes.

A security policy can be applied to get/set a particular parameter based on its security attributes. It is also possible to specify a security policy for the whole configuration update that ensures individual parameter updates are aligned with each other.

KSS uses secure storage to store security policy parameters. Storage can also be used by any application in a system and a security policy has fine-grained control over which application can use which parameters.

www.kaspersky.com

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