Institute for
System Architecture
Operating
Systems Group

TECHNISCHE
UNIVERSITÄT
DRESDEN

English version

µSINA
Secure Microkernel-based System Architecture

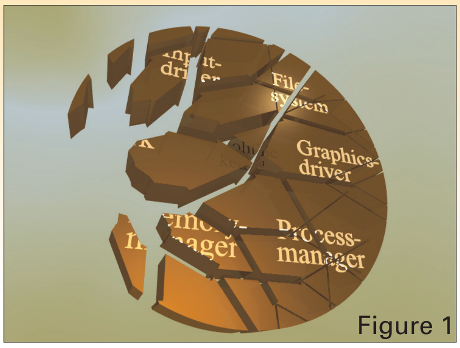

Figure 1

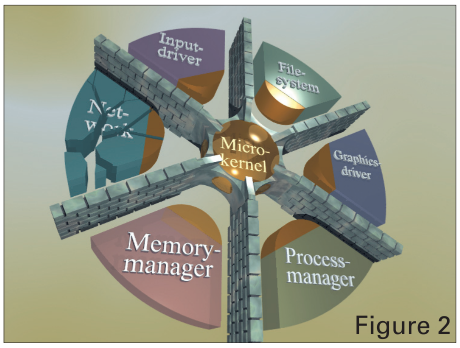

Figure 2

Motivation

Modern operating systems must provide a wide variety of functionality. This includes complex software components such as network stacks, file systems, process management as well as driver support for a broad range of devices and architectures.

On the other hand, with today's increasing reliance on IT infrastructures, modern operating systems must meet very high demands for secure operation. This becomes increasingly difficult if not impossible as long as operating systems are based on monolithic kernels. Monolithic kernels – the classical approach used by the majority of todays commodity operating systems – integrate all basic components of the operating system into a complex piece of software. The entire monolithic kernel is executed in hardware privileged mode that enables it to directly access and manipulate all hardware and software components of the system. Users of such systems need to trust the entire monolithic kernel. Confidence into the correct and secure operation of such large and complex monolithic structures is hardly justified as experience shows. For example, a typically configured kernel of the Linux operating system has about 500,000 lines of code. It is impossible to fully avoid bugs and security leaks at a system of such scale. Bug-prone program code or a successful penetrator can corrupt the operation of the whole system and thus, can impose fatal consequences (Figure 1).

The modern alternative to build systems that require high levels of confidence is based on microkernels. Microkernels allow to build the high confidence parts of systems as a collection of a small number of small components with small interfaces. They use virtual address spaces to encapsulate the components to the effect that a bug-prone component is locally confined (Figure 2). The minimal-complexity microkernel comprises the only part of the system that has hardware-kernel-mode privileges thus reducing the overall complexity of program code running in privileged mode by more than an order of magnitude. Our microkernel L4/Fiasco is implemented in only 15,000 lines of code.

A common application of microkernels as pioneered by Technische Universität Dresden is to run one or more entire, legacy monolithic kernels as user-level components on top of the microkernel besides critical, for example high-security or real-time parts of a system. In such a scenario, the legacy system, for example TU Dresden's "L4Linux", a Linux variant running on top of L4, is completely encapsulated and has no hardware privileges. And still, L4Linux is fully binary compatible for existing Linux applications.

This principle, to run legacy operating system code in encapsulated components besides critical components, can also be employed to allow critical components to reuse untrusted functionality of the legacy components. µSINA uses this approach to provide an implementation of a VPN gateway with a trusted computing base of minimal complexity running on the L4/Fiasco microkernel. In addition, it reuses as much legacy code in encapsulated components as possible for functionality that need to be part of the high confidence core of a VPN gateway.

A proof-of-concept VPN gateway

The pervasive nature and the relatively high availability of the internet leads to a wide application of virtual private networks (VPN). A VPN connects trusted islands over untrusted links and especially over the internet. This enables distant network nodes (for example portable devices) to securely communicate with a company's server via a virtual network. In VPNs, all traffic transported over the internet is encrypted via crytographic methods as described in the IPSec standard. This ensures the protection of sensible information against unauthorized inspection and manipulation. A VPN gateway implements the needed security mechanisms and acts as the interface between a private network and the public internet.

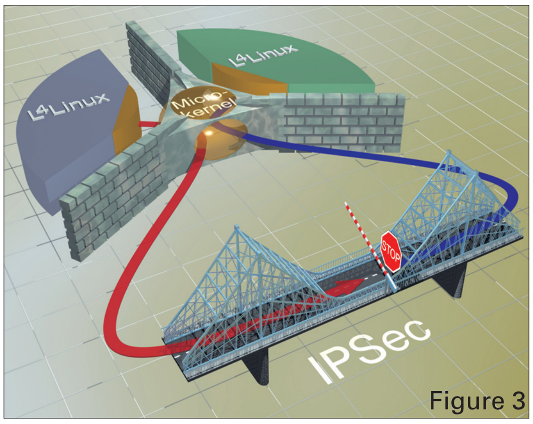

Figure 3

In monolithic operating systems, the IPSec implementation is integrated in the kernel and closely interwoven to other components of the kernel such as the network subsystem. However, the security relevant functions of a VPN implementation has a very small complexity compared to the rest of the monolithic kernel. Thus, bugs in the kernel code or successful penetration of the overly complex monolithic kernel can compromise the small set of security relevant functions.

Our implementation of a VPN gateway extracts the core IPSec functionality from the Linux kernel and executes it as a separate component, that we call a Viaduct (Figure 3). The Viaduct monitors the communication of two encapsulated instances of L4Linux that contain all other functionality needed for VPN gateway but that does not need to be trusted. The untrusted functionality includes network drivers and TCP/IP. Each instance of L4Linux transfers network packets only via a dedicated physical network connection (NIC), one connected to the private, the other to public network. The Viaduct controls the flow of information and thus, enforces the security policy of the system.

Through extracting this security-relevant component from the kernel we reduce the complexity of the trusted computing base of our VPN gateway by an order of magnitude in comparison to monolithic Linux. Furthermore, we enforce a clear separation between the privileged minimal-complexity microkernel, the IPSec implementation and the other components of the system. Intercomponent communication must use clearly specified interfaces and can be established only via mechanisms provided and monitored by the microkernel.

Microkernel-based secure system architecture

The VPN gateway as described so far is only one example that illustrates the way how microkernel technology can be used to partition secure systems into separated and protected components. Thus the effects of programming errors and security flaws of each component are locally restricted and do not comprise the whole system. The succesful implementation of our VPN gateway represents just a case study for further applications of microkernel technology in secure systems, for example firewalls, intrusion detection & response systems, routers and mobile end points.

In december 1999 the German Information Security Agency (BSI) commissioned the secunet Security Networks AG to develop SINA (Secure Inter-Network Architecture). SINA is an IT-security platform for processing, storing and verifying of confidential and ulteriorly sensitive data. Meanwhile, more than 2,000 SINA components are in use with a steadily increasing tendency. The implemented basic components of our microkernel-based secure network architecture lay the foundation of the next-generation SINA platform. The component-based approach as taken by µSINA enables further protected entities to be easily integrated into the system without any impact on the already established components. Therefore, µSINA enables openness and expandability of SINA in regard to future extensions, which makes this an ideal solution for almost all branches.

Co-operation partner

secunet Security Networks AG • Dr. Michael Sobirey
Ammonstraße 74 • 01067 Dresden
Tel: +49 (351) 43959-0
E-Mail: info@secunet.com • http://www.secunet.de

Bundesministerium
für Wirtschaft
und Arbeit

funded by Bundesministerium
für Wirtschaft und Arbeit

Contact

Technische Universität Dresden
Institute for System Architecture
Prof. Hermann Härtig
Hans-Grundig-Str. 25 • 01307 Dresden
Fon +49 351 463-39438
Fax +49 351 463-38284
E-Mail: mikrosina@os.inf.tu-dresden.de
http://os.inf.tu-dresden.de/mikrosina/