

Operating systems – shouldering the security and safety burden

David N. Kleidermacher, Green Hills Software

A NEW APPROACH DIVIDES AND CONQUERS THE PROBLEM OF OPERATING SYSTEM SECURITY. GREEN HILLS INTEGRITY ADOPTS THE MILS (MULTIPLE INDEPENDENT LEVELS OF SECURITY) ARCHITECTURE WHICH STIPULATES A LAYERED APPROACH TO SECURITY

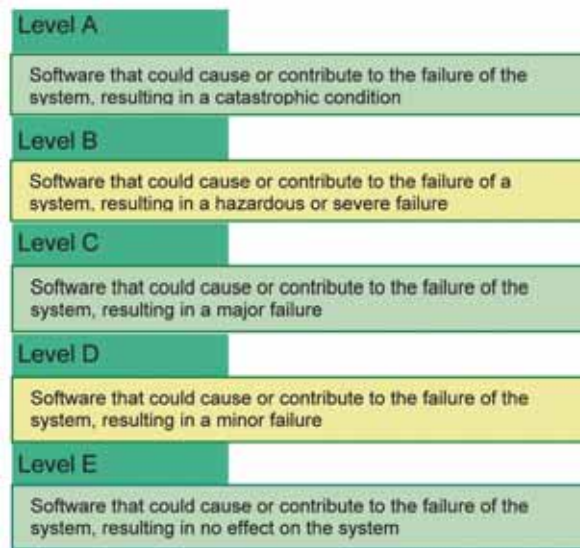


Figure 1. DO-178B software certification levels

Operating systems control the computers upon which our most security and safety critical systems depend. By adhering to a well established security or safety standard, such as DO-178B Level A, an operating system vendor can ensure that a single kernel can be used across a wide range of products, ranging from avionics and secure PDAs to industrial control and medical devices. The MILS architecture represents the future of safe and secure computing, with the separation kernel providing the foundation. The separation kernel partitions the system so that viruses or bugs are contained and software at varying levels of criticality (e.g. DO-178B Level A with Level C or classified and unclassified) can coexist on the same computer.

In both military and civil circles, computer security is an exceptionally hot topic. On the civil side an increasing number of safety-critical systems in automotive, financial, medical, infrastructure, and industrial control include networked computers. This blueprint makes for increased convenience and performance, but also opens new avenues for internal and external security threats. Network capability affects all aspects of daily life; from the workplace and office equipment to cars and even our homes through the use of internet-connected appliances that if sabotaged spell trouble for property and human life.

The operating system bears a tremendous burden in achieving safety and security. Because the operating system controls the resources (e.g. memory, CPU) of the computer, it has the power to prevent unauthorized use of these resources. Conversely, if the operating system fails to prevent or limit the damage resulting from unauthorized access, disaster can result. Operating system security is not a new field of research. Yet today, even though a few are on their way to achievement, there are no operating systems that have been successfully evaluated at the highest levels of assurance – EAL 5, 6, or 7 – the highest assurance levels of the Common Criteria, an internationally conceived and accepted security evaluation standard. The holy grail of high assurance for security is EAL 7 because it requires rigorous, formal design and mathematical proof that the security policies of the system are upheld. One of the reasons for the lack of secure operating systems is the historical approach taken to achieve security. Legacy security kernels attempted to provide a mass of services – protection and partitioning, mandatory access controls, secure file systems, and secure network services. As a result, these systems were simply too large and complicated to evaluate at high assurance levels.

Another serious problem is that civil and military organizations are employing operating

systems that were never designed for security in the first place. The Common Criteria states that EAL 4 (a low level of assurance) would be the highest level at which it is likely to be economically feasible to retrofit an existing product line. As a matter of fact, most SCADA systems (computer systems used to monitor and control a plant or equipment in industries such as water and waste control, energy, and oil refining) are running Windows. In 1998, a 12-year-old hacker broke into the computer system controlling the Roosevelt Dam and gained complete control of the dam's massive floodgates.

Recently companies such as Green Hills Software, with its INTEGRITY real-time operating system, have taken a new approach that divides and conquers the problem of operating system security. INTEGRITY adopts the MILS (multiple independent levels of security) architecture which stipulates a layered approach to security. At the foundation is the MILS separation kernel, a small, real-time microkernel that implements the following functional security policies: information flow – information cannot flow between partitioned applications; data isolation – the data within partitioned applications cannot be read or modified by other applications; damage limitation – if a bug or virus damages a partitioned application, this damage cannot spread to other applications; periods processing

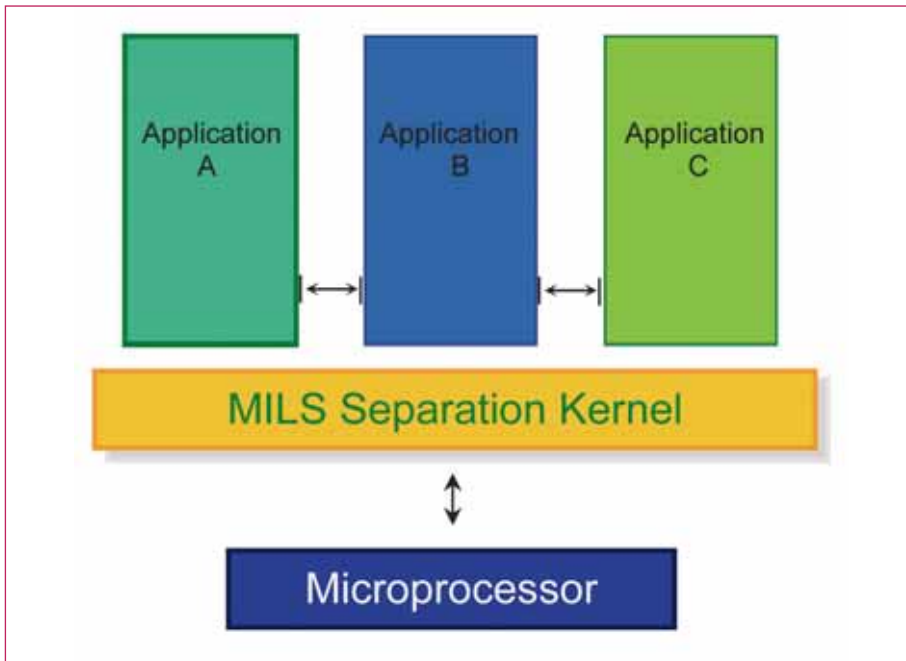


Figure 2. MILS architecture

– when switching from execution of one partitioned application to another, no latent information (such as data on the stack or in registers) from the old partition can be read by the new partition. In other words, the kernel must purge/scrub any resources of information before they can be reused.

The separation kernel realizes these policies by using the microprocessor’s memory protection hardware to prevent unauthorized access between partitions and by implementing resource allocation mechanisms that prevent one partition’s operation from affecting another (e.g. by exhausting a resource such as memory or CPU time). The MILS architecture also specifies enforcement of these policies such that they are: non-bypassable; always invoked; tamper proof as well as evaluatable. The requirement that the policy enforcement be evaluatable is absolutely critical and is the reason why the separation kernel enforces this focused set of policies and does not provide higher level security policies such as mandatory access control for files or network security. Since a high assurance Common Criteria evaluation requires a formal model and proof, a system of more than approximately 5000 lines of code becomes too difficult and expensive to evaluate. The MILS security policies can be implemented with a microkernel that is small enough to be evaluated at the highest assurance level.

Under the MILS concept, higher level secure software, such as a secure communications mechanism, web server or file system, can be layered on top of the separation microkernel. The MILS security policies are recursive: a MILS file system, using the fact that the un-

derlying separation kernel enforces its partitioning security policies, can be used to ensure file system data isolation, information flow, and damage limitation properties. In addition, multi-level security (MLS) can be built on top of the MILS components. The MILS components that make up an actual system can be selected by system designers as needed. If the system does not require a secure web server, then there is no need to go through the pain of evaluating one. MILS components can be independently evaluated at the highest assurance level and can come from multiple vendors.

Another major advantage of the separation kernel is that it allows software at varying levels of criticality to run on a single microprocessor. For example, an application containing classified data and algorithms can occupy one partition while another partition is connected to the unclassified internet. The MILS security policies, if assured at the highest level, make this possible. This can lead to enormous cost savings in product development because complicated multi-function applications can run on a single powerful microprocessor without requiring all these applications to be evaluated at the highest assurance level. An operating system that can meet the highest assurance levels of Common Criteria is a candidate for other demanding safety and security evaluations across multiple industries and requirements. For example, the same Green Hills INTEGRITY operating system under Common Criteria evaluation has been certified in DO-178B Level A avionics systems. A Level A system is one whose failure could be catastrophic and, consequently, the assurance requirements for Level A products are extremely demanding. ■