Operating Systems: Shouldering the Security and Safety Burden

In both military and civil circles, computer security is an exceptionally hot topic. A properly partitioned and small operating system can serve as a foundation for the highest levels of security.

by David Kleidermacher
Green Hills Software

On the civilian 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 automobiles and even into our homes through the use of Internet-connected appliances, which if sabotaged spell trouble for property and human life.

On the military side, the DoD believes future combat success depends on building the Global Information Grid (GIG) that will network together assets (soldiers, aircraft, ships, tanks, etc.) so that information can be efficiently and accurately distributed across theaters of operation. The security ramifications of this plan are staggering, especially when you consider the Internet. A single affected node on the GIG could spread like a cancer, placing missions and lives at risk.

Both internal and external threats, malicious and non-malicious are of concern. Aside from external infiltrations, internal attacks, such as a backdoor, can be targeted to inflict the worst possible damage at the worst possible time.

It is well known that intricately placed subversions often escape source code inspection. Case in point: In the ’80s the U.S. created a plan to sabotage a pipeline. The incident was recounted in the former Secretary of the Air Force, Thomas Reed’s book, “At the Abyss: An Insider’s History of the Cold War.” The account tells how the CIA inserted a Trojan horse into control software sold to the Soviets for the trans-Siberian gas pipeline. The sabotage eventually resulted in a three-kiloton explosion. We should expect our enemies to attempt similar infiltrations. Demanding high assurance operating systems is the only way to combat these security risks.

The Role of the Operating System

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 levels called the highest Evaluated Assurance Levels—EAL-5, 6 or 7—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 kitchen sink 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.

Reprinted from RTC September 2004

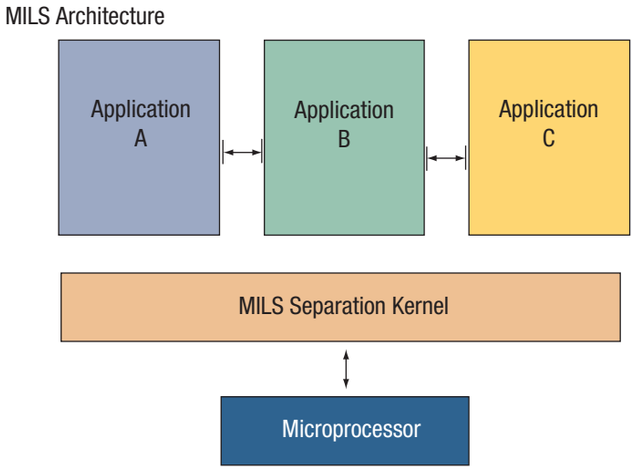

Figure 1 The MILS separation kernel restricts information flow between partitioned applications, isolates data so it cannot be read by other applications and limits the damage caused by viruses or bugs.

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 state that EAL-4 (a low level of assurance) "is the highest level at which it is likely to be economically feasible to retrofit an existing product line." It would be a bad idea to trust our critical systems to insecure operating systems, right? As a matter of fact, most of the nation's 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 have taken a new approach that divides and conquers the problem of operating system security, adopting the Multiple Independent Levels of Security (MILS) 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:* 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, Tamperproof and Verifiable.

The requirement that the policy enforcement be Verifiable 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 5,000 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 (Figure 1).

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 an underlying 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 of these applications to be evaluated at the highest assurance level.

More Security and Safety Standards

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.

A National Security Agency (NSA) Type 1 product, for example, makes use of military-grade cryptography and is used to secure classified U.S. government information. Such products must be certified by the National Security Agency. Type 2 products are endorsed by the NSA and deal only with unclassified information. Type 1 certification is not a published standard. The NSA applies internal methods to evaluate the device; it's certified when the NSA says it's certified.

The DO-178B standard, published by the Radio Technical Commission for Aeronautics (RTCA) defines guidelines for the development of aviation computer systems. DO-178B focuses on ensuring safety through a robust software development lifecycle. In contrast to the Common Criteria, which allow for evaluation of individual components of a system (such as the separation kernel), DO-178B certification applies to the whole "box." For example, it may certify a flight management system, which may include an operating system. It does not, however, certify the operating system by itself. DO-178B specifies five levels of criticality, Level A through Level E. A Level A system is one whose failure could be catastrophic. Consequently, the assurance requirements for Level A products are extremely demanding. Assurance requirements include a rigorous form of structural coverage analysis called Modified Condition/Decision coverage (MCDC) for every line of source code. DO-178B Level A assurance requirements overlap significantly with high assurance (EAL-6 or 7) Common Criteria requirements. In fact, the same Green Hills INTEGRITY operating system under Common Criteria evaluation has been certified in DO-178B Level A avionics systems (Figure 2).

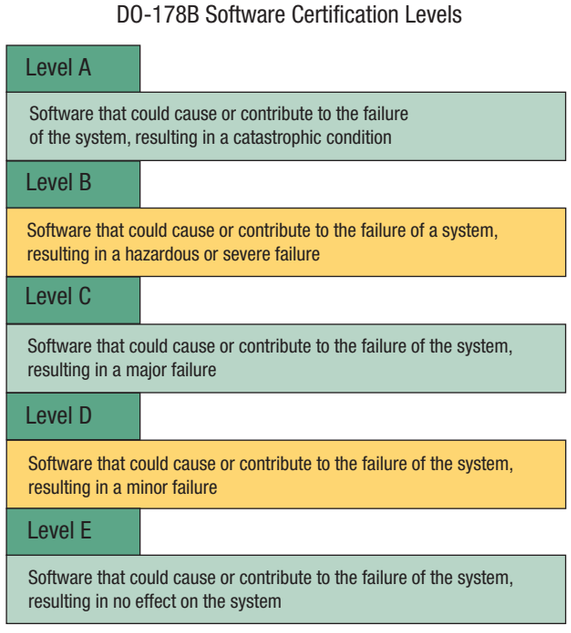

Figure 2 DO-178B defines five levels of criticality for software. These can map approximately to levels of required security.

The IEC 61508 standard, published by the International Electrotechnical Commission (IEC) also provides guidelines for the development of safety-critical systems. IEC 61508 defines four safety integrity levels—SIL 1 through SIL 4, with SIL 4 being the most demanding. IEC 61508 defines its safety levels by the amount of safety risk reduction required. Since the risk reduction factor is difficult to ascertain for software, IEC 61508, like DO-178B, focuses on a well developed design and development process to ensure software quality. IEC 61508 is used heavily in the industrial control and automation industries. Like Common Criteria, IEC 61508 allows for certification of a stand-alone component such as the operating system.

Unlike the previous security and safety standards that are designed to be widely applicable and generalized, Federal Information Processing Standards (FIPS) Publication 140-2 enumerates security requirements specifically for cryptographic modules. For a separation kernel, this standard would likely only apply to the secure delivery of the software (use of cryptographic protection to ensure that the bits created by the vendor are the same bits received by the end customer). In order to achieve high assurance (e.g., Common Criteria EAL-6 or 7, NSA Type 1 approval), higher level services such as secure communications and file storage systems, because they use encryption to protect sensitive data, would require compliance with portions of FIPS PUB 140-2. FIPS PUB 140-2 specifies four qualitatively increasing security levels, 1 through 4 as required for the secure design and implementation of cryptographic components.

The Food and Drug Administration specifies three device classes, I through III, with Class III referring to devices that support or sustain human life. Unlike the FAA, which requires the specific plans and processes of DO-178B for flight critical software, the FDA only provides general guidance to device manufacturers for software. The FDA Center for Devices and Radiological Health provides guidance document 1, "General Principles of Software Validation", which espouses many of the same principles, such as rigorous planning, traceability, configuration management and testing, as other safety and security standards. However, no specific techniques or methods are required. Rather, FDA regulators place the burden on the device manufacturer to demonstrate that software meets device-specific safety requirements.

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, 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. ◢

Green Hills Software
Santa Barbara, CA.
(805) 965-6044.
[www.ghs.com].

www.ghs.com ▲ 805-965-6044