MILS: Protecting our most vital systems

By Rance J. DeLong

Photo courtesy of U.S. Army

Military systems increasingly involve safety- and security-critical matters, requiring a new approach to designing high-threat, high-asset-value systems. The emerging Multiple Independent Levels of Security (MILS) paradigm provides a modular, flexible, and trustworthy foundation for national security and critical infrastructure.

Commercial products and approaches to security do not meet the requirements of contemporary network-centric "systems of systems" in high-threat, high-asset-value environments. The MILS paradigm[1] – based on a new breed of commercial security products including high-assurance separation kernels and middleware – applies venerable, established principles alongside recent advancements in microprocessors, computer security, avionics safety, software engineering, and formal methods.

Critical trade-offs

The unrelenting growth of embedded controls, information processing, and communications in military systems has caused a massive spike in demand for computer power in deployed systems. The challenge of meeting such demands within the feasible constraints of Space, Weight, and Power (SWaP) will never be fully resolved, but consolidating functions on powerful microprocessors helps ease SWaP constraints.

The C4ISR trend is gravitating toward increasing connectivity and increasing need for controlled sharing; coalitions are formed, redefined, and dissolved. Information must be shared and analyzed at speeds dictated by tactical constraints. Systems must protect valuable information assets and be robust against serious threats.

Systems increasingly involve safety- and security-critical considerations. Military vehicles, ships, and aircraft serve as weapons platforms and intelligence conduits, and onboard computer systems are becoming more and more integral to vehicle operation. Diverse requirements for different kinds of systems are being streamlined into combined requirements that must be met by a single system.

Safety and security background

Well-established yet distinct traditions for the construction of dependable safety- and security-critical systems require a degree of assurance that far surpasses what "best commercial practice" provides.

The safety of commercial airborne systems is subject to SAE Aerospace Recommended Practice as interpreted by RTCA DO-178B[2] requirements, and corresponding safety standards exist for military aircraft. The DO-178B Level A (the most stringent level) can be characterized as technically conservative because it applies conventional process and testing practices, albeit very thoroughly and conscientiously. DO-178B does provide an escape clause for alternative methods, such as formal methods, as long as they achieve the same objectives as the ones prescribed. This is rumored to be much more explicit in the awaited DO-178C, scheduled for release in late 2008.

The development history of high-assurance secure systems is long, if sparse, and can be characterized as technically progressive because it has applied the best available methods. Confidence in the trustworthiness of secure systems has typically been sought through formal methods. In fact, security and formal methods grew up together in the 1970s when much advancement in formal methods was motivated – and funded – by security projects.

Since the 1960s, security projects have recognized the need for security to be designed and implemented at the low-

est levels: the operating system and the hardware mechanisms that support it. Representatives of early secure operating system developments include ADEPT-50, the Multics security enhancements, the UCLA Data Secure Unix Kernel, the Kernelized Secure Operating System (KSOS), the Secure Communications Processor (SCOMP), the Provably Secure Operating System (PSOS), Multinet Gateway, BLACKER, the Boeing MLS LAN, and GEMSOS.

These systems often incorporated the security policy-enforcing mechanisms in a security kernel, typically a general purpose, heavyweight operating system. The policy enforced was a mandatory access control policy, usually a version of the Bell-LaPadula Model (BLP)[3], also known as Multi-Level Security (MLS). BLP attempts to formally describe the familiar practice of classifying information, assigning clearances to individuals, and granting or denying access on the basis of classification, clearance, and mode of access.

One limitation of these systems is that practical, operational considerations lead to the need for trusted processes that require special privileges granted by the security kernel in order to perform their functions. The security kernel taken with such non-kernel security-related software comprised the Trusted Computing Base (TCB).

Assurance background

The primary barrier to providing a convincing argument about the trustworthiness of a TCB is complexity. The security kernel and other TCB components typically comprise large, complex, monolithic objects. To perform rigorous and complete analysis of such objects using formal methods was beyond state-of-the-art 25 years ago and is arguably so even today. Rigorous and complete analysis or formal methods refer to using specifications written in languages that have formal semantics and an associated proof system so that analysis and proofs are automated (or human directed), objective, repeatable, and logically sound. Such methods can only be applied to well-structured objects of limited complexity.

Consequently, one of the tenets of MILS is to decompose a system into a collection of reusable components, each of which is small enough to be rigorously analyzed for correctness and/or security properties.

Because MILS is intended to meet both safety and security standards, it would be tempting to apply all of the processes recommended for security and safety standards. This, however, would result in an excessive burden and cost of process. Rather than applying the union of the processes, defining a single process that would satisfy the union of the two standards' objectives is recommended.

The National Computer Security Center (NCSC), formed at the NSA in 1981, published the Trusted Computer System Evaluation Criteria (TCSEC)[4] to provide a standard for the requirements for secure systems and the measurement of systems intended to meet those requirements. The standards represented in the TCSEC evolved through a series of renditions including the Federal Criteria in the United States and the Information Technology Security Evaluation Criteria in the United Kingdom and Europe. Appearing first in 1996, since 1998 the Common Criteria (CC)[5] has provided a broadly accepted international standard (ISO/IEC 15408). The TCSEC (a.k.a. Orange Book) designated systems according to D, C1, C2, B1, B2, B3, and A1. The CC designates systems according to Evaluation Assurance Levels (EAL) 1 through 7.

Precursors to MILS

Spurred by the NCSC and TCSEC, a host of computer vendors commercially produced trusted MLS operating systems from the mid-1980s through the 1990s. These systems are, for the most part, only medium assurance, that is, B1 according to the TCSEC or EAL 4 according to the CC. Ironically, since such systems can only be evaluated to medium assurance, they do not meet accreditation requirements to be deployed in the environments where they would actually be used to protect and separate classified data. Instead, EAL 5 through EAL 7 are required, depending upon the threat environment and the value of the assets.

Microkernels date back to the 1980s, originally serving as the basis for early experiments in factoring operating system functionality into a minimal kernel supporting highly modular services. They typically performed poorly compared to monolithic operating systems on the microprocessors of the day.

Virtual Machine Monitors (VMMs) date back to the 1972 IBM VM/370. Traditionally, a VMM creates a virtual environment indistinguishable from the bare hardware an operating system may run on without modification. A VMM is not a separation kernel, and vice versa. A VMM enforces a policy of isolation, while a separation kernel additionally enforces a policy of information flow control. A separation kernel could be constructed with VMM properties, provided appropriate hardware support is available.

Enter MILS: the separation kernel and MILS middleware

A separation kernel, first proposed by John Rushby[6], program director for formal methods and dependable systems at SRI International, works with the protection mechanisms provided by the underlying microprocessor hardware to enforce with a very high degree of assurance the primitive policies of isolation and information flow control – the prerequisite guarantees needed for the construction of software reference validation mechanisms that enforce higher-level policies such as MLS. The MILS paradigm depends explicitly on Saltzer's and Schroeder's[7] Principle of Least Privilege and Principle of Complete Mediation, enforced within the separation kernel and supported by the separation kernel for higher levels of the system design. The security requirements for a separation kernel are set forth in the Separation Kernel Protection Profile (SKPP)[8].

When the separation kernel was first conceived, microprocessor features and performance were not adequate to implement complex systems while paying the security tax for robust isolation provided by a separation kernel. As recently as 10 years ago, 10,000 partition switches per second would have left little, if any, of a processor's cycles available for applica-

tions. Today, a processor can perform 60,000 partition switches per second and still have more than 95 percent of its cycles available to applications. Quantitative improvement in processor speed has enabled MILS' qualitatively different approach to security, putting separation kernels squarely in the sweet spot of the perennial performance/security trade-off.

Most of the services provided by conventional operating systems are pushed out of the separation kernel into other high-assurance components referred to as MILS middleware.

The separation kernel and MILS middleware subsystems must be sufficiently simple to enable rigorous analysis of the properties of each. The MILS paradigm calls for each high-assurance subsystem to be decomposed into as many elements as necessary to facilitate that analysis by delineating the role and constraints on each element. Complex high-assurance systems can, in turn, be constructed from MILS components by building upon the functionality and security properties of those components. Figure 1 shows an MLS system constructed in the MILS style.

LynxSecure Separation Kernel

LynuxWorks, in collaboration with SRI International, is developing a high-assurance separation kernel and an integrated formal development approach for MILS systems. The project aims to provide a high-assurance integrated development environment that will enable experienced engineers, though not experts in formal methods, to use this secure separation kernel to develop high-assurance products and systems. The companies are also helping lead the MILS community through The Open Group's Real-Time Embedded Systems (RTES) Forum (www.opengroup.org/rtforum).

The LynxSecure Separation Kernel (Figure 2) fully implements the SKPP and will be certified at the highest levels of security and safety: CC EAL 7+ and DO-178B Level A. Using Intel Virtualization Technology as a platform for its first release, this kernel will create virtual machines able to run heavyweight operating systems such as Microsoft Windows as guest operating systems without modification. Hardware virtualization support, now appearing in commodity microprocessors, makes it possible to provide virtual machines with a minimum performance impact.

The LynxSecure Separation Kernel also includes a high-assurance runtime interface, a lightweight guest operating system with a simple, formally specified and verified API that facilitates the construction of high-assurance applications.

Unprecedented vendor cooperation

One surprising outcome of the MILS initiative is that it has encouraged cooperation and collaboration from competitors that usually go out of their way to avoid each other. Because MILS components don't come from a single source, each part is provided by a company specializing in that particular technology. This requires an extraordinary level of cooperation among competitors to achieve products that can not only interoperate but do so securely.

As a testament to this cooperation, four MILS component vendors and a major system integrator presented an integrated live demonstration of MILS capability and interoperability at the meeting of The Open Group in Washington, D.C. in April 2006.

Within The Open Group's RTES Forum, the MILS community is developing a coherent set of community standards in the form of protection profiles based on the Common Criteria (CC). There, MILS stakeholders are working in concert to achieve a common vision of the MILS architecture and to streamline the process of developing the many needed protection profiles and other standards.

Future of MILS

The MILS effort is breaking new ground in the area of high-assurance security and functional composition. Each part of a high-assurance system must meet a precise set of constraints for the whole to meet system-level security requirements and functionality. The RTES Forum is working toward a unifying MILS integration framework.

The military's Global Information Grid (GIG) envisioned by strategic planners levies challenging requirements for information assurance: a dynamic, highly connected environment that demands unprecedented robustness, flexibility, and agility. The degree of security robustness required in a dynamic, highly connected network architecture cannot be overestimated. MILS promises to provide this kind of assurance and flexibility at every level, from sensor and control processors to communications devices to workstations and servers.

MILS-style MLS System Stack

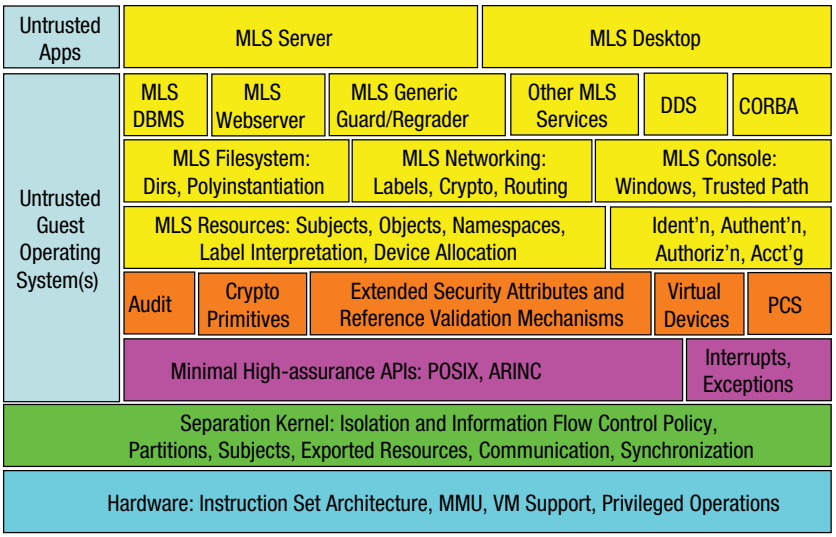

Figure 1

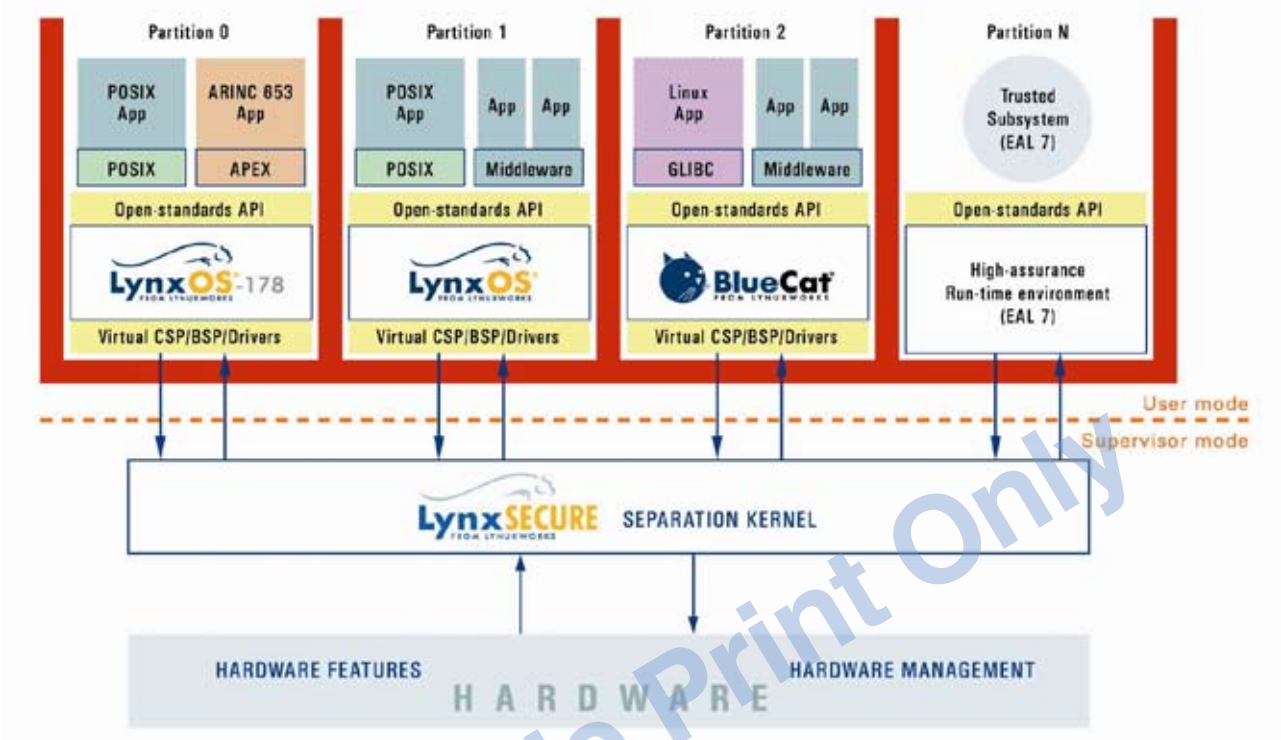

Figure 2

MILS shows potential for meeting the requirements of high-assurance security with new, high-assurance commercial products. MILS components exhibit the robustness needed for key security-enforcing GIG mechanisms.

Some significant defense programs have already committed to MILS, and many others are prime candidates. MILS component vendors are broadcasting the message that they are committed and that MILS technology is coming.

Acknowledgments

The author acknowledges colleagues Ed Mooring of LynuxWorks and John Rushby of SRI International for providing the critical mass of experience and know-how that has been applied to this work on the LynxSecure Separation Kernel and the methodology and tools for high-assurance development.

References

1. W. M. Vanfleet, R. W. Beckwith, B. Calloni, J. A. Luke, C. Taylor, and G. Uchenick. MILS: architecture for high assurance embedded computing. CrossTalk, 18:12–16, August 2005.
2. Requirements and Technical Concepts for Aviation, Washington, DC. DO-178B: Software Considerations in Airborne Systems and Equipment Certification, December 1992.
3. D. E. Bell and L. J. LaPadula. Secure computer system: Unified exposition and Multics interpretation. Technical Report ESD-TR-75-306, Mitre Corporation, Bedford, MA, March 1975.
4. Department of Defense. Department of Defense Trusted Computer System Evaluation Criteria, December 1985. DOD 5200.28-STD.
5. Common Criteria for Information Technology Security Evaluation, September 2006. Version 3.1, CCMB-2006-09-001, 002, 003.
6. John Rushby. The design and verification of secure systems. In Eighth ACM Symposium on Operating System Principles, pages 12–21, December 1981.
7. J. H. Saltzer and M. D. Schroeder. The protection of information in computer systems. In Proceedings of the IEEE, volume 63, pages 1278–1308, September 1975.
8. Information Assurance Directorate, National Security Agency. U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness, October 2006. Version 1.1.

Rance J. DeLong *is staff scientist for security and assurance at LynuxWorks, Inc., and an adjunct lecturer at the Center for Advanced Study and Practice of Information Assurance at Santa Clara University. He has 28 years of security product development experience including the Kernelized Secure Operating System, the Provably Secure Operating System, and Sun Microsystems' Trusted Solaris. Rance has a BS in Physics/Mathematics, a BA in Philosophy from Moravian College, and has completed extensive postgraduate work at Lehigh and Stanford universities.*

LynuxWorks, Inc.
855 Embedded Way
San Jose, CA 95138
408-979-3900
www.lynuxworks.com