

Multilevel security in tightly coupled military systems: Virtualization as a path to MLS

By Diana L. Hecht, PhD, and Warren A. Rosen, PhD

The authors describe the similarities between virtualization and Multi-Level Security (MLS) systems and provide an example of how both can be supported in a COTS network protocol by placing security labels in network packet headers and specialized processing structures built into switches.

Security is a major cost driver in military systems and is of particular concern when using commercial network protocols in the military environment. MLS systems may be employed to reduce the amount of application software that must be secured and certified at the highest level, avoiding the expense and complexity associated with maintaining the entire system at the highest level of security. The major problem with MLS systems is that the limited demand for them (primarily military and government agencies) does not compensate for the high costs of development and certification. Virtualization, on the other hand, is rapidly becoming a standard technology capable of supporting security while also providing scalability, flexibility, increased utilization, lower cost, and availability for today's IT infrastructure.

It should be noted that MLS differs from Multiple Independent Levels of Security (MILS). MILS is a layered architecture consisting of a layer directly above the hardware known as the *separation kernel*. The next layer, directly above the kernel is known as *middleware* and runs in user mode and performs the tasks traditionally handled by the OS (memory allocation, I/O drivers, and so on). It also provides services to extend the scope of the separation kernel allowing for intersystem communication. The uppermost layer is where the user applications run.

We describe how virtualization can be used as a path to a multi-level secure system and how network security can be provided for such a system in a military environment. We first explain the similarities between MLS and virtualized systems and then, as an example, show how virtualization and security can be supported within the RapidIO protocol. We provide an extension to the existing RapidIO protocol through the overloading of existing fields in a particular packet header format in order to provide a security label designation.

Features of MLS and virtualization

Security is an increasing concern in military and other mission-critical computing environments. The major security risk that exists in current approaches to general purpose computing arises

from sharing resources among the various processes/applications running in the computer. Typical military systems support a number of applications running concurrently that create or use information of different security levels or classifications. In such cases, it is extremely important to separate information/resources with different security levels and to prevent any leakage of information among the various application processes that are running at different security clearance levels (top secret, sensitive, unclassified, and so forth).

MLS systems and virtualized systems are similar in nature in terms of their goal of providing strong isolation of their supported components (application/memory partitions in the case of MLS systems and virtual machines in the case of virtualized systems). An MLS system is one that has system resources at more than one security level and is able to prevent users from accessing resources for which they lack authorization. Applications are compartmentalized so that they can be prevented from interacting with one another. Security labels or tags are assigned to resources (memory partition, application data) when they are created or allocated to indicate the security level required for access. The operating system uses the security label to allow only authorized processes to access the data or other resources.

Virtualization provides a layer of abstraction between the physical hardware and the operating system and/or user applications. It also allows multiple virtual machines, possibly with different operating systems, to run in isolation, side-by-side on the same physical machine. A Virtual Machine Manager (VMM) or hypervisor is a small kernel that resides just above the hardware level and manages the virtual machines and their access to system resources.

MLS systems and virtualized systems differ in the degree of isolation and containment they provide (Table 1). MLS systems can consolidate data of different sensitivities onto a single computer or system and regulate access to the data by ensuring that data of a certain sensitivity level can only be accessed by authorized users. In order to enforce appropriate user access to data, processes and their assigned resources are compartmentalized so that they can be prevented from interacting with one another. The entire system works under a single operating system that provides mechanisms for keeping processes of differing security levels from interacting or interfering with each other. Virtualized systems provide a framework under which a number of virtual

Comparison of MLS and virtualization

	MLS	Virtualization
Similarities		
Goal	Allows multiple users with different security clearance to share physical machine but prevents unauthorized access to data or system resources	Allows optimum utilization of physical machine by switching between virtual machines and managing shared resources
Isolation	Provided by MLS operating system	Provided by virtual machine monitor
Differences		
Unit of isolation	Partitions within a machine	Between virtual machines
I/O separation	Makes use of security label	Makes use of Device-ID
Market	Primarily used in military systems	Becoming widespread in IT markets

Table 1

machines (each with their own operating system and address space) can coexist on the same physical machine.

I/O (network interface) for MLS/Virtualization

For the purposes of this article, we are interested in a method for providing isolation of I/O data (primarily traffic on the interconnection network of a distributed system). Both virtualized and MLS systems currently provide support for isolation of I/O data. Virtualized I/O mechanisms are beginning to emerge from vendors such as AMD and Intel. AMD's solution is to offer an I/O Memory Management Unit (IOMMU) that provides management of DMA accesses involving both access permission verification and physical-to-virtual address translation (based on which virtual machine the I/O device is assigned to). A Device-ID is used to access the data structures necessary to verify access rights to the data and to access the relevant address translation page tables. Intel's solution is based on a similar approach using the VT-d architecture and the Device-ID to access relevant data structures. Traditionally, MLS systems support isolation of network data via labeled networks. Security labels are placed in the network packets, and a monitor or guard mechanism is used in the network interfaces or switches to manage/restrict packet travel through the network and delivery to the destination.

One of the main differences between the domain protection provided in the I/O virtualization architecture and the protection required by the MLS implementation is that current virtualized systems control access to data based on the I/O device requesting access (using a Device-ID). Meanwhile, MLS systems control access to data based on the memory partition or process within a node that makes the request (using a security label). Since different partitions may have unique security levels, it is not sufficient for access to a partition in the destination node to be granted solely based on the device/node ID. The decision of

whether to grant access to a partition must be based both on the device/node ID and the security level of the individual partition within the node that generated the request.

RapidIO and MLS/Virtualization

To show how virtualization can be used to provide multilevel security in a COTS network, we used the RapidIO protocol as an example. RapidIO is becoming increasingly popular in military systems due to its high throughput, low latency, small footprint, low power, robustness, and support for a wide variety of features such as encapsulation.

A promising approach to adding security within the RapidIO protocol is to place security labels in the Class of Service field of the Type-9 packet header. This provides switches with the information necessary to determine whether a packet should be routed through to the output port/link indicated by the destination address. The Type-9 packet format is the Data Streaming transaction format presented in the RapidIO Interconnect Specification Part 10: Data Streaming Logical Specification, which provides support for segmentation and reassembly, encapsulation, and Class of Service designation. Rydal's proposed MLS extensions encapsulate RapidIO Logical I/O layer packets inside a Type-9 packet, thereby providing support for security in existing systems using basic I/O transactions described in the RapidIO Interconnect Specification Part 1: Input/Output Logical Specification (RapidIO.org).

When a process resident on a particular node requests a read/write access to a memory address located on another node, the RapidIO packet is created, and the security label assigned to the partition from which the request originated is placed in the Class of Service field in the packet header. The input port of the switch connected to an endpoint node receives the RapidIO packet and uses the destination ID to determine the security levels that are allowed at the destination node. The major data structure used for this task is the Node Partition Table, shown in Figure 1. The Node Partition Table is indexed by the destination ID, and each entry in the table is a 32-bit mask indicating all of the partition security levels that may access that node. This table is

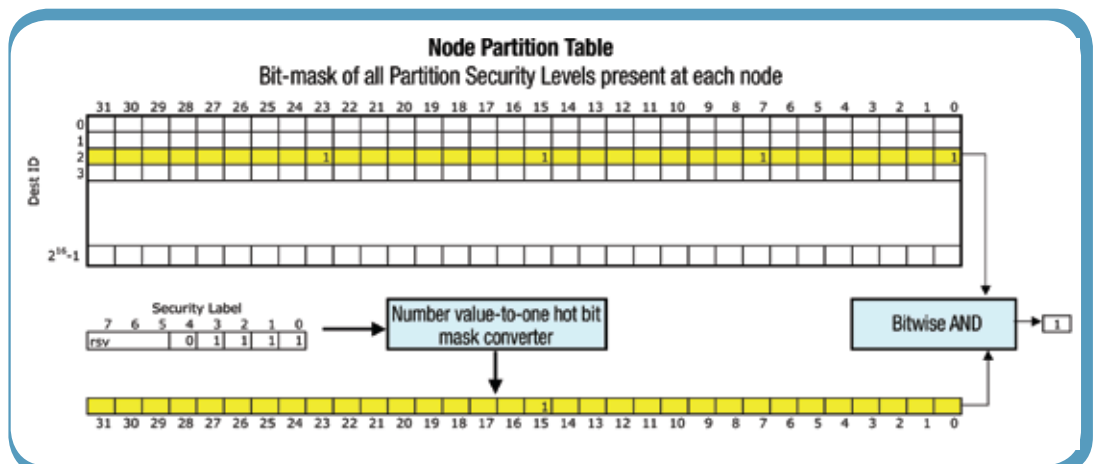


Figure 1

used by the switch processing logic to cross-check an incoming request of a particular security level with the allowable security levels for accesses to the destination node. If a read/write request is issued by a partition in the source node that does not have the authority to access the requested partition in the destination node, the request is not allowed to proceed through the switch but is instead handled as a security violation error condition.

If source and destination nodes are trusted, the source node can be relied upon to correctly generate the security label from the memory address for the read/write operation and the destination node can be relied upon to correctly examine the data request, memory address, Source-ID, and security level to verify that the access should be granted. If the OS of the destination node is not trusted, the switch must be able to determine the partition to which the memory address belongs and compare it to the security label in the switch before allowing the packet to pass through the link to the node's input port. Figure 2 illustrates the two cases.

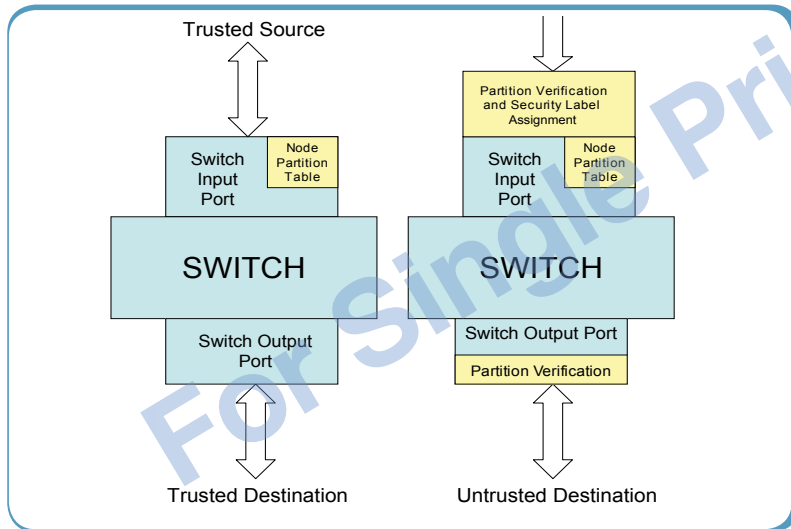


Figure 2

To make use of the RapidIO-MLS extensions in one of the virtualized systems described earlier, the I/O virtualization architecture implementation (IOMMU) must receive as the Device-ID a descriptor that can be used to indicate the security level of the partition (rather than the device) that initiated the request. Support for MLS on current virtualized systems can be provided by passing the security label to the IOMMU in place of a Device-ID. Providing a security label in the Type-9 packet header allows RapidIO the ability to support both MLS security and virtualization, thereby allowing a much larger number of applications to take advantage of the powerful features provided by RapidIO.

RapidIO provides path to support both MLS and virtualization

We have described the similarities and differences between MLS and virtualized systems and shown how virtualization and security can be supported within the RapidIO protocol. By building on the similarities between MLS and virtualization, a network can be designed to support systems based on an MLS operating system or virtualized systems. We have presented an example of

one approach to this within the RapidIO network protocol. The changes to the protocol packet structure are minimal, providing a great deal of flexibility for any combination of traditional, secure, or virtualized system structure.✦



Dr. Diana L. Hecht is senior research engineer at Rydal Research and Development, Inc. Currently she works in the area of advanced network and signal-processing technology. Diana is involved in a number of research and development efforts aimed at high-performance signal processing for military applications and led the design team that developed Rydal's FPGA-based RapidIO switch. She has also participated in research involving network switch design sponsored by the Office of Naval Research through a subcontract from Rydal Research. Diana holds a PhD in Computer Engineering from Drexel University.



Dr. Warren A. Rosen, president, founded Rydal Research and Development, Inc. in 1998 for the purpose of carrying out research and development of advanced networking and signal-processing technologies. Prior to that, he worked at the Naval Air Warfare Center, Aircraft Division in Warminster, Pennsylvania, where he established an optical communications laboratory for development and characterization of optical components, systems, and protocols for high-performance avionics data networks. He has conducted research sponsored by the National Security Agency, National Science Foundation, the National Oceanic and Atmospheric Administration, DARPA, the Office of Naval Research, and the Missile Defense Agency. He holds four U.S. patents in computer networking and signal processing. He earned his PhD in Physics from Temple University.

Rydal Research and Development, Inc.

1523 Noble Road
 Rydal, PA 19046
 215-886-5678
 diana@rydalresearch.com
 warren@rydalresearch.com
 www.rydalresearch.com