RC 23624 (W0506-041), 6 June 2005 (Revision 1 19 October 2005)
Computer Science

IBM Research Report

Multi-Level Security Requirements for Hypervisors

Paul A. Karger
IBM Research Division
Thomas J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598, USA

IBM

Research Division

Almaden – Austin – Beijing – Delhi – Haifa – T.J. Watson – Tokyo – Zurich

Limited Distribution Notice: This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties). Some reports are available at http://www.research.ibm.com/resources/paper_search.html. Copies may requested from IBM T.J. Watson Research Center, 16-220, P.O. Box 218, Yorktown Heights, NY 10598 or send email to reports@us.ibm.com.

This paper has been accepted by the 21st Annual Computer Security Applications Conference, to be held 5-9 December 2005 in Tucson, AZ. It will appear in the conference proceedings, published by the IEEE, and downloadable from http://www.acsac.org

Multi-Level Security Requirements for Hypervisors

Paul A. Karger
IBM Thomas J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598, USA
karger@watson.ibm.com

Abstract

Using hypervisors or virtual machine monitors for security has become very popular in recent years, and a number of proposals have been made for supporting multi-level security on secure hypervisors, including PR/SM, NetTop, sHype, and others. This paper looks at the requirements that users of MLS systems will have and discusses their implications on the design of multi-level secure hypervisors. It contrasts the new directions for secure hypervisors with the earlier efforts of KVM/370 and Digital's A1-secure VMM kernel.

1 Purpose of this paper

There have been a number of recent efforts to develop multi-level security (MLS) for hypervisors or virtual machine monitors (VMMs), such as NetTop [39], sHype [43], and a proposed combination of Xen [16] and sHype [32]. There has been a lot of confusion about what the requirements are to adequately support multi-level security (MLS) in a hypervisor. The hypervisor is being used to separate multiple instances of untrusted operating systems, running at different security levels. The purpose of this paper is to clarify what end-users of MLS expect to be able to do[1], and what technical issues impact those requirements at Common Criteria levels EAL4 and above.[2] This paper presents no major new ideas or innovations. The goal is to assist developers of hypervisors to decide which of these ideas and features are important to make multi-level security useful to the end-users. A hypervisor with fewer features is less expensive to build and is easier to evaluate under the Common Criteria. However, if the hypervisor is too restrictive, then the customers will be unable to implement the MLS applications that they want to run. This paper identifies a set of features that are needed to make the hypervisor useful, yet are still simple enough to assure its security.

2 End-User Expectations

2.1 What does Multi-Level Secure Mean?

MLS systems can mean many things to many people. What this paper will describe are the requirements and implications of a multi-level secure mode of operation as was defined many years ago in DoD Directive 5200.28 [11] and in the implementing manual [13].[3] A system that runs in *multi-level secure mode* has information at a variety of classification levels[4], but not all users are cleared for all information. By contrast, most classified systems in the DoD today run in *system-high mode* and have information at a variety of classification levels, but all users are cleared for the most sensitive information in the system. The system may be a single machine or an entire network. For example, the DoD's SIPR network stores information marked from Unclassified through Secret, but all users are required to have at least a Secret clearance. There is also a *controlled mode* of operation in which all users are cleared to some level, but not necessarily the highest level of information. The first successfully deployed controlled mode

[1] The end-user requirements are derived from the author's personal experience designing, deploying, and supporting a variety of MLS systems within the DoD and in designing high security hypervisors.

[2] Trusted Information Systems, Inc. developed a proposed interpretation of the Orange Book for Virtual Machine Monitors [10] that attempted to clarify some of these issues, but it did not address the networking issues on which this paper particularly focuses.

[3] A more modern version of these definitions can be found in [6]

[4] This paper will speak of security levels in most cases to make the language simpler. However, using only hierarchic security levels is an over-simplification of the model. The DoD security model is actually a lattice-structure with both levels and categories. A point in the lattice is usually called an access class, and any pair of access classes may be comparable (<, =, or >) or they may be disjoint and totally incomparable. See [17] for details.

system was the Multics system at the Air Force Data Services Center in the Pentagon that processed Top Secret information, but allowed users who were only cleared for Secret. Under the old Orange Book evaluation system [5] and as recommended in the Yellow Books [4, 12], system-high systems were typically evaluated B1 or below. Controlled mode systems were typically evaluated at B2, and true multi-level mode required B3 or higher. Translating to the Common Criteria [7-9], B1 and below are roughly EAL4 and below, B2 is roughly EAL5, and B3 and higher are roughly EAL6 and higher.

This paper has focused on the use of MLS for the defense applications, but they are by no means limited to defense applications. MLS can be extremely useful in commercial applications. An example use of MLS in a frequent-flyer smart card application is shown in [28, 29]. IBM has developed a new extended mandatory access control model, designed to provide multi-organizational MLS in a meaningful way to the entire Internet. This is described in [24] and in section 3 of [46]. The development of multi-organizational MLS for commercial use also has payoffs for the military. Traditional military MLS models have been single-organization models. Everyone in the Department of Defense follows the same security rules. However, this traditional single-organization model has problems when multi-national coalition forces must work together. Each country's military has its own security policies, and those policies do not easily map into a single policy. By contrast, IBM's multi-organizational MLS, designed to handle many different businesses on a single world-wide Internet, is much better suited to modeling the many different security policies of multi-national coalition forces.

2.2 What do Users Want to do with MLS Systems?

The most basic requirement is that the MLS system keeps highly classified information from leaking to people who are not properly cleared. This requirement is met by a system that implements the Bell and LaPadula security model [17]. However, this requirement can also be met by simply keeping data of different classifications on different computer systems and restricting access to those systems by clearance levels. Most systems in the DoD do exactly that and run in a system-high mode.

The biggest problem with system-high mode is that sharing information across security levels is very hard. Users at high levels of security want to be able to read low-level information, even though they do not want to contaminate that low-level information with high-level secrets. Keeping multiple copies of the low level information on different machines running at different system-high levels is not acceptable. First, you need to have significantly larger amounts of storage in such a case, and keeping the data synchronized can be very difficult. If you update the low-level data on a low-level machine, that update must be replicated onto all the other copies. Such replication is particularly difficult, because machines running at different system-high levels must NOT be networked together. The DoD frequently has to resort to *sneakernet* to apply these types of updates.

Users also want to downgrade information from higher security levels to lower security levels. The simplest form of this is the statutory downgrading required after the passage of specific numbers of years. Since statutory downgrading only happens after multiple decades have passed, there is little need to make it happen in real time, although there is a need for efficiently downgrading large numbers of files from archival storage.

However, there is another form of downgrading that does need to be done quickly and in real time. A user at a high security level may determine that a particular piece of information needs to be made available to someone at a lower security clearance. For example, an intelligence analyst may determine from a spy's report that the enemy is going to attack at dawn. The defenders who need to know about the upcoming attack, but those defenders should not know who is the spy. The analyst must sanitize the information, removing any indicator of who the spy is, but leaving the information that the enemy will attack at dawn.[5] The analyst needs to be able to isolate the information to be downgraded, ensure that the particular information cannot be modified until the downgrade operation has completed, and then release that information to the recipient on a timely basis.

3 Implications of the Bell and LaPadula Security Model

The Bell and LaPadula security model [17] imposes a number of constraints on possible implementations of MLS systems. In particular, Bell and LaPadula require

[5] Sanitization without leaving indicators is often very tricky, but for this paper, we assume that the analyst can easily determine which information is safe to downgrade.

that each process in a single system (or each system-high machine in a network) be identified at a particular security level. That process is allowed to read lower-classified information, but it is not allowed to write files that are marked at a lower classification level. This is to prevent Trojan horses from releasing arbitrary information. Note that this is a basic requirement of the model at evaluation levels EAL4 and above. It is not to be confused with covert channel issues [33, 36] that only come into play at B2 or EAL5 and above.

The result of this no-write-down requirement is that network connections between system-high systems are only generally useful if the systems are at precisely the same system-high level. Most network protocols require two-way communications (if only for packet acknowledgements), and acknowledgements cannot be permitted from high to low. This requirement is made clear in the Trusted Network Interpretation (TNI) [14] of the Orange Book [5].[6] It is possible to build truly one-way networks. Such networks were first proposed in chapter 7 of [25] and in [26]. Rushby and Randell [42] proposed a complete implementation of such a system, based on the Newcastle Connection, developed at University of Newcastle. There have been several commercial products evaluated in Australia[7] to implement one-way networks of one kind or another. These products from BAE Systems, Compucat, and Tenix Defence Systems all provide very limited communications capabilities.[8]

Why are these one-way networks so limited? Most network protocols use two-way communications to implement both flow control and error control. If you cannot have two-way communications, then there needs to be a trusted intermediary that accept and error check all messages sent by the sender, even if the receiver is refusing all input. This means that the intermediary may need huge amounts of buffer memory to hold hours or days worth of traffic. In addition, many protocols that run on top of TCP need two-way communications. For example, the FTP protocol [40] cannot run over a one-way network. The above-mentioned products use their own proprietary protocol to transfer files from low to high.

4 Hypervisor Implications

There are two classes of hypervisors that must be considered when examining the technical implications of MLS for hypervisors. The two classes are pure isolation hypervisors and sharing hypervisors.

4.1 Pure Isolation Hypervisors

A pure isolation hypervisor simply divides a machine into partitions, and permits no sharing of resources between the partitions (other than CPU time and primary memory). Implementing a pure isolation hypervisor is very easy, because the only security policy to be enforced is isolation. IBM's EAL5-evaluated PR/SM system [2] for the z/Series mainframes is a good example of a pure isolation hypervisor. There is essentially no sharing between partitions in PR/SM. PR/SM does have features for certain very limited forms of sharing (such as channel to channel connections, etc.), but under the EAL5 evaluation certificate, such sharing is absolutely forbidden. If a customer site turned on such sharing, they would no longer be running an evaluated configuration.

The partitions of a pure isolation hypervisor are essentially just like a collection of system-high separate computers. Each partition has its own disks and network connections, and if one partition is unclassified and the other is secret, then there cannot even be a network connection between them.

A valid question is, "Who would want a pure isolation hypervisor? You can get the same results by running several separate machines." In the case of a z/Series mainframe, there is a good reason. Mainframes are so expensive that the ability to partition one system into several isolated systems will save the customer lots of money, even if no sharing is permitted.

[6] The TNI [14] explicitly calls for strictly one-way networking at level B2 in section 3.2.1.3.4. However, in the B1 sections of the TNI, section 3.1.1.3.1 requires accurate labels on information transferred between network trusted computing base (NTCB) partitions, and section 3.1.1.4 requires that subjects and objects used for communication with other components are under control of the NTCS partition. The phrase "under control" is critical here, because the distinction between overt communications channels that must be secure at B1 and covert communications channels that need not be secure until B2 is whether or not they are "under control" of the TCB. Since the subjects and objects for communication are under control of the NTCB, the issues of one-way communications and packet acknowledgements are NOT covert channel issues. This is an inconsistency in the TNI and not an unexpected one. The TNI has been criticized in a number of ways for inconsistencies like this in [44].

[7] http://www.dsd.gov.au/infosec/evaluation_services/epl/dap.html

[8] The BAE Systems product evaluation report [3] indicates that it may have covert channel issues that are discussed in classified supplementary reports. The covert channel situation seems better on the other two products.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.