z/VM Security and Integrity

Alan Altmark
IBM Corporation
Alan_Altmark@us.ibm.com

Cliff Laking
IBM United Kingdom Ltd
Cliff_Laking@uk.ibm.com

Table of Contents

Introduction

The purpose of this document is to provide information about the security and integrity characteristics of the z/VM® hypervisor.

When IBM @server® zSeries® hardware and virtualization technology are used to host virtual servers, the combination provides a highly attractive platform for the Internet Service Provider (ISP), Application Service Provider (ASP), Internet Data Center (IDC), or any other commercial or academic customer, from both a technical and a Total Cost of Ownership (TCO) point of view.

In assessing a proposal that incorporates z/VM, a prospective customer may wish to know how the isolation and integrity of virtual servers can be provided and what security measures are implemented for both access control and management of the environment.

The information herein draws on material from a variety of sources - IBM manuals, ITSO Redbooks™, Web based data, etc. It is not a complete survey of the security and integrity features in z/VM, but encompasses those that will be of most interest to customers contemplating running virtual servers in a z/VM environment, with particular emphasis on Linux®. It should be noted, however, that security of the z/VM system and security of the virtual servers such as Linux are completely separate issues and must be addressed independently. The subject of securing individual Linux servers is not covered in depth within this paper, but references that discuss this in detail can be found at the end of this document.

VM has been used for decades by IBM customers in pratically all industry sectors and by IBM as a trusted, reliable, security-rich and robust platform for multi-user computing and for hosting multiple virtual zSeries servers.

This paper is divided into four major sections. It examines first the issue of virtual machine integrity - that is, how virtual machines are isolated from each other and from the hypervisor functions of z/VM. Second, the security aspects of z/VM are discussed, including authentication and resource access controls. Third, it provides guidance on enhancing z/VM security and integrity. Finally, it briefly touches on Linux virtual sever security and its relationship to z/VM.

System Integrity

Computer users often confuse the terms security and integrity. While related, they are very different concepts.

Within z/VM, the term security is a reference to the authentication and authorization schemes used to identify users and to control access to resources. System integrity, on the other hand, allows the z/VM Control Program (CP) to operate without interference or harm, intentional or not, from the guest virtual machines, as well as protecting the guest virtual machines from interfering with each other.

IBM has been developing and improving the virtual server technology contained within z/VM for more than 35 years. As a direct result of this commitment, the chances of a new exposure being identified are extremely small. But in the event an integrity exposure is identified, IBM is committed to react quickly to address the problem. As a matter of policy, IBM does not disclose the details of integrity or security defects to the general public, though it may, at its discretion, disclose the information to specific individuals, institutions, or organizations that it determines have a legitimate need to know about them.

The System Integrity Statement for z/VM

The IBM commitment to resolve system integrity problems is not just "wishful thinking" by the authors. In fact, IBM publishes a System Integrity Statement for z/VM in z/VM Version 5 General Information, which is cited here:

> *IBM has implemented specific design and coding guidelines for maintaining system integrity in the development of z/VM. Procedures have also been established to make the application of these design and coding guidelines a formal part of the design and development process.*
>
> *However, because it is not possible to certify that any system has perfect integrity, IBM will accept APARs [problem reports] that describe exposures to the system integrity of z/VM or that describe problems encountered when a program running in a virtual machine not authorized by a mechanism under the customer's control introduces an exposure to the system integrity of z/VM...*
>
> *IBM will continue its efforts to enhance the integrity of z/VM and to respond promptly when exposures are identified.*

It is important to understand the elements of system operation that contribute to system integrity in the z/VM environment. The z/VM Version 5 General Information publication defines the specific limitations placed on virtual machines so that the integrity of the system is maintained at all times. In particular, a virtual machine without special privileges, authorizations, or access rights may not:

- *Obtain control with a privilege class or other authorization greater than those assigned to the virtual machine by the system administrator.*
- *Circumvent or disable the Control Program's access controls of protected system resources, whether by password or an External Security Manager (ESM). This includes, but is not limited to, commands, DIAGNOSE functions, minidisks, virtual unit record devices, tape volumes, remote network nodes, and terminals.*
- *Obtain control of the CPU in real supervisor state.*
- *Circumvent or disable the Control Program's isolation of the real memory used by virtual machines*
- *Circumvent or disable the disk extent access limitations placed on a virtual machine by the Control Program.*
- *Circumvent the system integrity of any other guest operating system (that itself claims to have system integrity) as the result of an operation by any Control Program command or facility. The term "guest operating system" refers to any operating system that is running in a virtual machine.*

Much of this protection is provided by specific hardware facilities discussed throughout this document.

Because the Control Program and the virtual machine configurations are under the control of the customer, the actual level of system integrity that a customer achieves will depend on how the z/VM environment is set up and maintained.

This is made clear by having customer responsibilities defined as follows, again from z/VM Version 5 General Information:

> *While protection of the customer's data remains the customer's responsibility, data security continues to be an area of vital importance to IBM. The IBM commitment to the system integrity of the z/VM environment, as described in this statement, represents a further significant step to help customers protect their data.*
>
> *Product documentation, subject to change, describes the actions that must be taken and the facilities that must be restricted to complement the system integrity support provided by z/VM. Such actions and restrictions may vary depending on the system, configuration, or environment. The customer is responsible for the selection, application, adequacy, and implementation of these actions and restrictions, and for appropriate application controls.*

IBM will accept integrity APARs and will fix any integrity problem so reported.

System Integrity Implementation by z/VM

The IBM eServer zSeries z/Architecture™ is at the core of the system's ability to maintain integrity. One crucial aspect of this is the ability to keep each virtual machine isolated from every other virtual machine. This isolation even extends to the Control Program because it is logically separate from all virtual machines in the system.

Interpretive Execution Facility

At the center of z/VM integrity is the Interpretive Execution Facility of the zSeries hardware. It is an element of Processor Resource/Systems Manager™ (PR/SM™) that permits a virtual machine instruction stream to be run on the processor using a single instruction, SIE. The SIE instruction is used by the machine's Logical Partitioning (LPAR) support to divide a zSeries processor complex into proven secure logical partitions.

When the Control Program dispatches a virtual machine, details about the virtual machine are provided to the hardware. The SIE instruction runs the virtual machine until the virtual machine's time slice has been consumed, or the virtual machine wants to perform an operation that either the hardware cannot virtualize or for which the Control Program must regain control. At that point, the virtual machine exits from SIE, control is returned to Control Program which simulates the instruction or performs the I/O (for example), and control is returned to the virtual machine. In this way, the full capabilities and speed of the CPU are available to the virtual machine and only those instructions that require assistance from or validation by the Control Program are intercepted.

This mechanism also enables the Control Program to limit the scope of many kinds of hardware or software failures. When the error can be isolated to a particular virtual machine, only that virtual machine fails and it can be reinitialized (rebooted) without affecting any testing or production work running in other virtual machines. The Control Program is designed so that failures that occur in virtual machines do not affect the Control Program or other virtual machines.

IBM eServer zSeries 900 PR/SM was evaluated at Common Criteria EAL5 on 27 February 2003. IBM eServer zSeries 800 and 900 GA3 PR /SM was evaluated at Common Criteria EAL5 and EAL4 on 06 June 2003. IBM eServer zSeries 990 PR/SM was evaluated at Common Criteria EAL5 in Germany, EAL4 world-wide, on 13 May 2004. IBM eServer zSeries z890 is designed to offer many of the features and functions of the z990 and is currently under evaluation for EAL5 certification. See **ibm.com**/security/standards/st_evaluations.shtml for the latest security evaluation information on IBM products.

Virtual Memory

Like most other platforms, zSeries provides an address translation capability, allowing an operating system to create virtual address spaces for memory isolation and management. For example, Linux running natively on a zSeries machine creates a separate address space for each process. Each address space has an associated set of region, segment and page tables which contain precise information on the real memory locations being used by the process. These tables are used by the address translation hardware to convert virtual memory addresses to real memory addresses. Because these tables are maintained by the operating system and are not accessible by the processes themselves, it is not possible for a process to read from or write to memory that is used by the operating system itself or another process.

zSeries takes this capability a step further by supporting two levels of address translation. An operating system running in a virtual machine under z/VM constructs its address-translation tables as usual to isolate and contain the memory for its processes. The entire memory of this virtual machine, although viewed by the guest operating system as real memory, is in fact virtual storage as well, defined by another set of translation tables managed by the z/VM Control Program. Even if an application running on a guest operating system were able to compromise the integrity of the guest, the damage would be limited to that one virtual machine, because of the separate layer of protection provided by zSeries hardware and z/VM.

A virtual machine may not access an address space owned by another virtual machine unless the address space owner allows the virtual machine to do so. The only exception to this rule is when the virtual machine uses a CP command to alter real memory. This command, CP STORE HOST, is, as you might expect, a privileged command assigned to privilege class C.

A different set of hardware facilities are used to isolate the memory used by z/VM preferred guests. These guests are not paged in or out, but reside in real memory at fixed storage locations called zones. Instead of using region, segment, and page tables to map virtual memory addresses to real memory addresses, the hardware simply relocates the address by applying a fixed offset. In addition, an address limit is applied (again, by the hardware) to any address so that a preferred guest cannot access memory outside of its zone. This zone relocation works together with special I/O assists to enable the Control Program to dispatch a preferred guest with minimal overhead.

A block of real memory, called a page frame, is allocated to a virtual machine when it first refers to a virtual page for which a real page has not yet been allocated. Allocation continues, as needed, up to the maximum virtual memory size permitted by the virtual machine's z/VM system directory entry.

In z/VM, many virtual machines share main memory simultaneously. Because main memory cannot hold every page of every virtual machine at the same time, pages of virtual memory not being actively referenced are transferred to expanded storage (a special classification of memory) or to disk. This is known as "paging out." The real memory that the page occupied then becomes available to another virtual machine. If that same page is needed again, the system transfers it from expanded storage or disk back into main memory in a process known as "paging in."

A powerful feature of z/VM is its ability to allow multiple virtual machines to share memory. These shared segments help the system manage its storage more efficiently by dramatically reducing the number of duplicate page frames.

Shared segments can be read-only or read-write blocks of storage containing code or data shared by many virtual machines. It is valuable, for example, for several Linux virtual servers to share read-only kernel code or application binaries and libraries in order to economize on utilization of real storage. This is an instance where several virtual machines' segment tables will point to the same page frames in real memory.

If a virtual machine has a legitimate reason for wanting to change a read-only saved segment, then that virtual machine must specifically request an exclusive copy of the saved segment and be authorized to do so in the z/VM directory. The unmodified code remains shared among the other virtual machines. z/VM exploits hardware memory protection mechanisms to ensure that a virtual machine cannot alter read-only saved segments.

In addition to read-only pages, a shared segment can include pages that are shared in read-write mode, meaning all virtual machines can write into the memory at the same time.

As z/VM is delivered, only a virtual machine with privilege class E can define and save a shared segment. Care should be exercised so that the ability to define and save shared segments is not given out indiscriminately.

Virtual Devices

The Control Program erects a barrier between the virtual machines and the devices to which the Control Program has access. A primary function of the Control Program is to mediate access to those real devices in different ways, depending on whether the device is intended to be shared between two or more virtual machines simultaneously, such as DASD, or whether the device is to be made available for the exclusive use of a single virtual machine, such as a tape drive.

When a virtual machine makes an I/O request, the request is intercepted by the Control Program so that virtual memory addresses in the I/O request can be translated to their corresponding real memory addresses. In addition, the Control Program examines the I/O request so that no harmful device maintenance requests or device subsystem functions are performed except by authorized virtual machines. Once the I/O operation has been validated, the Control Program starts the I/O operation on behalf of the virtual machine.

The virtual DASD devices used by virtual machines are called minidisks. They are created by partitioning a real DASD volume into cylinder ranges that appear as separate disk volumes to the virtual server. A minidisk can span a whole real disk volume. Or, as is sometimes needed, an entire DASD volume can be dedicated to a virtual machine for its exclusive use.

If the device is being shared, the Control Program will prefix the I/O request with additional device controls to further limit it. For example, I/O requests to minidisks are constrained by the insertion of a DEFINE EXTENT channel command into the real channel program, which forces the device to limit the I/O request to only a certain range of cylinders. In other cases a virtual machine will be given read-only access to a device, in which case the Control Program inserts commands into the I/O request that disable all write-type operations.

In this manner, the surrounding zSeries control units and devices themselves help z/VM to maintain user data integrity and privacy.

As mentioned earlier, data integrity is the responsibility of the customer. For example, when a minidisk is write-shared between two or more virtual machines, some application or guest operating system software is needed to manage access to the data. While the Control Program is able to cache the contents of minidisks in real or expanded storage to improve application response times, and to share this minidisk cache among several virtual machines, it does not mediate access to the minidisk data. It is up to the customer to enable the guest operating system or application to use the facilities inherent in the I/O architecture or available in the Control Program to provide full data integrity.

The planning required for shared data integrity among multiple virtual machines is no different than that which is required when devices are shared among physical machines. In fact, the same device functions are available to serialize access to devices shared among virtual machines as would be used to serialize access to real devices shared among real machines, including Reserve/Release and Assign/Unassign.

Failure to plan for and implement data integrity functions present in applications or the guest operating systems may result in data loss on a write-shared minidisk.

Control Program Commands and Functions

Virtual machines communicate with the Control Program in one of two ways:

1. A person or automation tool may issue Control Program (CP) commands from the virtual machine console.
2. The programs running in the virtual machine may themselves communicate with CP using the DIAGNOSE instruction. The parameters passed with the DIAGNOSE instruction provide all of the details CP requires to obtain input and return a response.

The CP command set and the various functions of the DIAGNOSE instruction are divided into functional groups called privilege classes. The set of general user commands and functions intended to be used by all virtual machines, such as the ability to IPL a guest operating system, link to permitted minidisks, display and alter the guest's virtual storage, create and delete virtual I/O devices, and reset the virtual machine, among others, are confined to the single privilege class G . By design, none of the class G commands can affect the Control Program or other virtual machines.

Additional privilege classes may be assigned by the system administrator, depending on the virtual machine's need and function, but additional privileges should be given to only trusted, secure virtual machines as some of the additional CP commands that will be made available are designed to alter the Control Program or real hardware resources such as CPUs or I/O devices, and may affect the security and integrity of the system as a whole.

If a virtual machine attempts to use a CP command or DIAGNOSE instruction that is outside its privilege class, the system ignores the command and an error condition is returned to the virtual machine.

IBM designed the structure of z/VM privilege classes with the organizational hierarchy of a typical computing installation in mind. If you find the IBM privilege class structure inappropriate to your installation's needs, you can modify it or completely replace it. It is possible to define up to 32 privilege classes that partly or completely override the privilege class structure that comes with your system. In this way you can make specific privileged commands available to a virtual server without giving access to all other CP commands and functions that are, by default, in the same privilege class.

System Security

To answer questions about z/VM security, we must take a closer look at the way z/VM handles the sharing, isolation, reconfiguration, and management of resources, as well as the specific authentication and authorization mechanisms that are available to customers.

A customer can augment the Control Program's native security capabilities by the addition of an External Security Manager (ESM) product such as the IBM Resource Access Control Facility for VM (RACF®). The ESM offloads many of the security functions to a separate subsystem, allowing the implementation of various access rules and groups, simplifying the administration of the users on the system. It also provides more granular authorization and auditing capabilities than are available without an ESM. A more complete description of the IBM RACF product is provided later in this document.

Virtual Machine Definition

In the simplest terms, z/VM takes the principles of partitioning – which at the hardware level are implemented by the PR/SM microcode – and enriches them through virtualization. The z/VM Control Program is able to virtualize hardware resources, either by sharing or partitioning real hardware resources, or by emulating their behavior. The definition of the virtual machines (also known as guest systems or virtual servers) and of the resources available to them, as well as the management of this environment, is also provided by the Control Program.

Virtual machine definitions are contained in the system directory. A virtual machine definition includes:

- *The virtual machine identifier (user ID)*
- *The virtual machine password (if an External Security Manager is not in use)*
- *POSIX user ID (UID) and group ID (gid)*
- *Assigned privilege classes*
- *Initial and maximum virtual storage sizes*
- *Real devices to which this virtual machine has exclusive access*
- *Simulated devices*
- *Virtual devices, such as minidisks*
- *Minidisk passwords*
- *Permissions to use special Control Program functions*

The system administrator is responsible for maintaining the system directory, whether manually by directly editing the directory files, or with the assistance of a directory management product such as the IBM Directory Maintenance for z/VM (DirMaint™) product. A directory management product is recommended as it allows virtual machine owners to change a limited set of attributes (such as passwords) without the assistance of the system administrator.

User Authentication

User login to a z/VM system is achieved by starting a terminal session with z/VM (local or telnet) and then providing a z/VM user ID and its associated password.

Local terminal sessions are by definition highly secure since the data does not travel over a network. Remote terminal (telnet) or file transfer (ftp) sessions which travel over internal or external IP networks can be made highly secure by configuring and using the z/VM Secure Sockets Layer (SSL) support. The processing required for SSL is delivered through an SSL server supplied with z/VM, supporting 128-bit encryption and decryption services. The z/VM SSL server is a Linux for zSeries application that must be installed separately.

Once the user has supplied the user ID and password, the Control Program validates the information. If the user ID and password are valid, the login is permitted and the terminal session is connected to the virtual machine's virtual console.

If you think of z/VM as a virtual computer room full of virtual servers, then think of a virtual machine user ID as a "virtual cage" around the server. No one can enter the cage unless they possess the key: the virtual machine password. This is very different from the discrete environment, where access to a machine room automatically gives access to all servers in that room.

Because the server console is protected by a z/VM password, you can more safely eliminate the protections normally given to a server's console. Automation techniques are greatly simplified if the automation tools do not have to, for example, enter the root password of a Linux server in order to shut it down or reboot it. While at first glance this may seem to reduce system security, it actually improves security by not requiring the root password to be known by the automation software. (The fewer people that know a secret, the safer that secret will be!)

A special capability available with z/VM is "Logon By." This function enables the system administrator to define a shared virtual machine. When the user enters the shared user ID, the user also provides his or her own user ID and password. In this way an audit trail is

maintained of who is actually logged into a shared user ID and the problems inherent in sharing passwords are avoided.

Remote access protocols such as rexec, ftp, and nfs, all require the client to authenticate using a z/VM user ID and password. At no time will z/VM trust the claims of an unauthenticated client. Once authenticated, the remote client has the same access rights as the user would have if he or she were logged into the system with a terminal session.

For network applications, z/VM provides a Kerberos server and the programming interfaces that permit programs to take advantage of Kerberos authentication and encryption facilities. It should be noted that the IBM-provided network application suite and the z/VM Control Program do not use Kerberos authentication.

While anonymous access to specific resources or to a virtual machine can be allowed by z/VM, such access must be explicitly enabled by the z/VM system administrator.

Authorization

Once logged into the z/VM system the virtual machine can access various types of resources within the z/VM system, including entire DASD volumes, minidisks, tape drives, network adapters, user files, system files, and so on. The security features of z/VM are designed so that a virtual machine can access only the resources specifically permitted to it.

Those permissions may be given by the system administrator so that when the virtual machine is started, it automatically receives access to a certain resource, even if that virtual machine would otherwise be unable to access that resource. Alternatively, permissions may be given dynamically by the system administrator or the owner of the resource.

Some resources are accessible based on privilege class, others require additional authorization.

The security facilities provided by z/VM can be enhanced according to any special or specific requirements for the customer's environment by the addition of an External Security Manager.

It should be noted that while privileged commands can be used to change a running z/VM system, the system administrator may choose to set system configuration options during system initialization. This is accomplished by updates to the Control Program system configuration file. Consequently, access to the system configuration file must be tightly controlled.

Intrusion Detection

As an element of z/VM intrusion detection capabilities, if a login is denied, the denial is tracked and a security journal entry is made when the number of denials exceeds an installation-defined maximum. When a second maximum is reached, logon to the user ID is disabled, an operator message is issued, and the terminal session is terminated.

Journaling is supported on z/VM. Virtual machine logons and linking to other virtual machine's minidisks are detected and recorded. Using the recorded information, you can identify attempts to log on to a virtual machine or to link to minidisks using invalid passwords.

To enhance the security and integrity of a z/VM system, there is a detailed list of points to consider and recommendations in Chapter 12 of z/VM Version 5 CP Planning and Administration. This manual also describes the z/VM facilities for detecting and foiling attempts to break system security and for detecting and preventing integrity exposures.

The TCP/IP component of z/VM will detect and report a variety of network intrusions., including SYN flooding, "Smurf" attacks and the "Ping o' Death".

Virtual Processor Security

The z/VM Control Program defines and assigns virtual processors to the virtual machine. These virtual processors are matched to the physical or logical (if z/VM is running in a logical partition) processors available to the Control Program. If the operating system running in the virtual machine is capable of using multiple processors, it will dispatch its workload on its virtual processors as if it were running in a dedicated hardware environment. This capability can be extremely useful to test a guest operating system in a multiprocessor mode, even on a uniprocessor system.

The Control Program handles dispatching the virtual processors on the available real processors. A real processor can either be dedicated to a single virtual machine or shared among multiple virtual machines. Bear in mind that the Control Program only handles the processors it controls, so if z/VM is running in an LPAR, the logical processors may in fact be shared with other LPARs.

So, we have virtual machines dispatching their work on one or more virtual processors, which are mapped to one or more logical processors, which may be mapped yet again to one or more physical processors.

But don't worry. There is no significant security risk if the virtual, logical, or physical processor configuration is changed, or if work is dispatched on different physical processors.

The state of a processor is preserved for one virtual machine and restored for another by the z/VM Control Program just as PR/SM does for LPARs. Therefore, no information can be passed from one virtual machine to another via residual data in processor registers.

Data in Memory Protection

Each virtual machine has its own virtual address space, which it sees as main memory. The physical residency of the guest system's memory pages in real storage is managed by the Control Program's paging mechanism. As previously described, pages that have not been referenced may be paged out and the page frame made available for use by another virtual machine.

When a virtual machine touches a page that is no longer in real storage, a page fault occurs and the Control Program will bring the missing virtual page back into real storage. Before the Control Program allocates a new page frame, that page is cleared of any residual data that may have been left behind by another virtual machine or the Control Program itself, preventing a virtual machine from having unauthorized access to memory-resident data.

The Control Program also allows the sharing of virtual pages by a number of virtual machines. The system administrator enables this by defining, populating, and then saving a named shared memory segment. Data in read-only pages of the segment may not be changed by any virtual machine. On the other hand, data in read-write pages of the segment may be changed by any virtual machine which maps that shared segment into its address space. Because of the potential to expose sensitive data or to interfere with the correct operation of the virtual machines using read-write pages as a lock, for example, a shared segment may be defined to be Restricted, meaning that only virtual machines with explicit authorization may load the shared segment.

Disk and Tape Storage Protection

Unlike a discrete system environment, z/VM does not require that individual virtual machines be given access to entire DASD volumes. Instead, z/VM partitions DASD volumes into minidisks to be owned and accessed by individual virtual machines. If you want to delete a minidisk, returning the disk cylinders to the "pool" of available space, you should consider using a product such as IBM DirMaint to automatically clear DASD space whenever a minidisk is deleted. This applies when you release an entire minidisk or when you reduce the size of a minidisk, and prevents the next virtual machine to which the space is allocated from seeing any residual data. It is the same concept used by installations when they decommission a DASD volume. The data is typically erased in order to prevent the next owner of the DASD volume from accessing sensitive or confidential data.

z/VM implements temporary minidisks which last only until they are detached or the virtual machine logs off. You may wish to tailor your system to automatically clear each temporary minidisk (T-disk) before it is reassigned. Simply add the Enable Clear_Tdisk operand on the Features system configuration file statement.

Likewise, when magnetic tapes are in use and being recycled you may need to implement procedures so that each tape is cleared of data before it is assigned to a different user. (Never degauss modern tape cartridges - doing so destroys the tape formatting and will render the tape unusable!)

FCP LUN Access Control

z/VM Version 4 Release 4 and z/VM Version 5 support attachment of Fiber Channel attached SCSI disks both for use by Linux guests and, in Version 5, by z/VM itself.

zSeries FCP LUN Access Control provides added security for SCSI devices residing on a Storage Area Network (SAN), since a zSeries host can contain multiple operating system images in different logical partitions or z/VM guests. For a logical partition that runs z/VM, you can specify separate permissions for the z/VM system itself (when running z/VM utilizing SCSI disks for system operations) and for each Linux guest. In addition, if you run a second-level z/VM system, you can specify separate permissions for that system and each of its guests.

Without this function, an FCP channel may be shared among multiple Linux guests, but each guest can access all storage controller ports and storage devices (logical units) accessible by the channel. The zSeries FCP LUN Access Control is designed to prevent unauthorized sharing of devices and to allow multiple Linux guests to share a single channel. It enables the system administrator to define access rights to individual storage ports and devices for each Linux image.

zSeries FCP LUN Access Control complements the zoning and LUN masking schemes that currently exist in other open storage environments. Both types of access control can be utilized, if desired. In addition, zSeries FCP LUN Access Control allows concurrent access by multiple Linux guests to devices (logical units) in read-only mode.

Virtual Input and Output

Virtual machines communicate with the outside world through virtual devices. The mapping of virtual to real devices and resources is handled transparently by the Control Program.

Minidisks can be shared or non-shared. If authorized, one virtual machine can link to a minidisk belonging to another virtual machine to access the data on it. Links can be either read-only or read-write. Read-only sharing of minidisks by virtual machines is often used to share file system data.

It is also possible to define and share virtual minidisks (VDISKs) which are mapped into real storage by the z/VM Control Program, instead of residing on real DASD volumes. The principles of using shared minidisks, shared minidisk cache and shared virtual minidisks among multiple virtual machines are an important advantage when running multiple guest operating systems under z/VM.

If a device is dedicated to a virtual machine, the z/VM operating system does not interfere in the use of this device by the guest operating system except when it is necessary to exclude requests that may alter an entire peripheral subsystem (such as a DASD control unit cache, for example).

From a z/VM perspective, physically shared but logically distinct devices (e.g. minidisks) are, for all intents and purposes, separate. One virtual machine cannot access another virtual machine's data (by seeking beyond the end of a defined minidisk, for example).

Virtual Networking

Communication between virtual machines is provided by various simulated devices or by facilities that are unique to the z/VM operating system. Available communications paths include z/VM Guest LANs, the z/VM Virtual Switch, IUCV, and virtual Channel-to-Channel connections. Each of these options provides a highly secure communication path which is not detectable or in any way "sniffable" by other virtual machines. That is, no other virtual machine may eavesdrop on the data moving between virtual machines. Of course, these virtual network connections are only as secure as the connected operating systems using them.

A typical way to connect from a virtual network to the outside world is to use one or more virtual routers. These are virtual machines which have both virtual network connections and real network connections, routing traffic as needed between the two.

Using z/VM Virtual Switch capability provides an alternative way to connect from a virtual network to the outside world without requiring virtual routers.

Virtual networks should be planned with the same care and attention to security as would be taken for a real, physical network. Networks, virtual or real, must be designed and implemented so that no unauthorized access to data or resources is possible. For system

administration tasks, a separate network with secure access is recommended. The ability to define multiple virtual routers gives the ability to completely isolate traffic moving in and out of the zSeries server.

Highly secure communication between LPARs can be easily handled by using zSeries HiperSockets™ connections.

Guest LANs

A z/VM Guest LAN, provides multipoint any-to-any virtual shared media connections between guests. A virtual machine accesses a Guest LAN using a virtual Network Interface Card (NIC), which emulates either a zSeries HiperSockets adapter or, an IBM OSA-Express adapter in QDIO mode. As many Guest LANs as are needed may be defined and used simultaneously; all are distinct with no cross-talk between them unless that traffic is routed from one LAN to another by a virtual router .

In order to prevent unauthorized connection to a Guest LAN, the creator of the LAN can define it to be restricted, permitting only specific virtual machines to connect to it. Further, only a user with privilege class B is allowed to create a Guest LAN that is owned by the Control Program and survives even after the virtual machine that created it logs off. It is these persistent Guest LANs that would most often have restricted membership.

Connections to a Guest LAN are established dynamically via the CP COUPLE command or by a SPECIAL statement in the virtual machine's system directory entry.

Virtual Switch

Virtual Switch, introduced in z/VM Version 4 Release 4, was designed to improve connectivity to a physical LAN for hosts coupled to a Guest LAN. It eliminated the need for a router by providing connectivity to a physical LAN through an Open Systems Adapter-Express (OSA-Express) with built-in failover capability. z/VM 5.1 enhances the authorization capabilities for z/VM Guest LANs and Virtual Switches by using the Resource Access Control Facility (RACF) or any equivalent external security manager (ESM) that supports this new authorization function. It is designed to provide ESM-centralized control of authorizations. RACF can be used to protect Guest LANs and virtual switches using profiles in the VMLAN class.

VLAN

The IEEE 802.3 standard defines a Virtual LAN or VLAN. A VLAN allows a physical network to be divided administratively into separate logical networks. In effect, these logical networks operate as if they are physically independent of each other. This is not to be

confused with z/VM Guest LAN, although Guest LAN and Virtual Switch can participate in a VLAN.

VLAN-capable devices manage the separation of traffic among VLANs. The z/VM Virtual Switch is such a device. When operating as a VLAN-aware switch, it is designed to conform to the requirements of IEEE 802.1Q, the standard which defines the operation of VLAN-capable switches and bridges. As a result, the Virtual Switch manages the assignment of z/VM users to specific VLANs and ensures that the guest will receive only the packets belonging to VLANs for which the user has been authorized. If desired, z/VM 5.1 allows VLAN authorizations to be placed under control of the external security manager.

Virtual Channel-To-Channel

The virtual Channel-To-Channel (VCTC) device emulates a real CTC adapter (IBM 3088). Each virtual machine defines a VCTC and then uses the CP COUPLE command to connect the two endpoints. I/O operations to the device are intercepted by the Control Program, which moves the data between a pair of virtual machines.

A VCTC may be defined dynamically via the CP DEFINE CTCA command, or it may be defined in the virtual machine's system directory entry. If defined in the system directory, the partner user ID may be specified in order to restrict who may connect to the virtual machine.

IUCV

The Inter-User Communications Vehicle (IUCV) provides a high-speed pipe for communications between virtual machines. Unlike simulated I/O devices, the IUCV connections can be established between pairs of virtual machines on the same z/VM system or on different z/VM systems. IUCV provides the cross-memory communication of virtual CTC devices, but without the overhead required to simulate an I/O device.

Authorization to establish IUCV connections is defined using the IUCV statement in a virtual machine's system directory entry. A particular virtual machine may be authorized to establish IUCV connections to any virtual machine, or to only specific virtual machines. To ease system administration, a server can be allowed to accept IUCV connections from any user, eliminating the need to provide explicit authorization for each client. In many cases, such servers provide their own client authorization functions.

Enhancing the Security and Integrity of z/VM

The following two IBM software products supported by z/VM can be used to enhance the security and integrity of your system:

- *IBM Directory Maintenance for z/VM and VM/ESA® (DirMaint)*
- *IBM Resource Access Control Facility (RACF)*

IBM Directory Maintenance for z/VM (DirMaint)

DirMaint provides a safe, efficient, and interactive way to maintain the z/VM system directory. Through its command line or full-screen interface you can quickly and easily add, modify, or delete users from the system directory.

DirMaint includes these important features:

- *Distributed virtual machine management. DirMaint is designed on the assumption that there are multiple system administrators and is designed so that two administrators may not change the same directory entry at the same time.*
- *Automatic minidisk allocation. Instead of requiring you to pore over minidisk map reports to find available "slots" for new minidisks, DirMaint will automatically locate gaps in any number of DASD pools that you define and assign new minidisks in those gaps. This avoids the accidental definition of overlapping minidisks.*
- *Automatic minidisk erasure. When a minidisk is deleted DirMaint will asynchronously erase all data content on the minidisk before returning it to the pool of available DASD so that no residual data remains.*
- *Support for end users. A general user has the ability to make limited changes to their own system directory entry.*
- *Auditing of all transactions.*
- *Automatic backup of the system directory.*

In any z/VM installation where large numbers of virtual servers are being deployed, DirMaint is recommended. DirMaint is packaged as a priced feature of z/VM.

RACF

The Resource Access Control Facility (RACF) is an External Security Manager. It provides comprehensive security capabilities that extend the standard security implemented by the base z/VM product. RACF controls user access to the VM system, checks authorization for use of both system and virtual machine resources, and audits the use of those resources. RACF is packaged as a priced feature of z/VM. It is pre-installed on the system installation media and can be enabled for an additional charge.

RACF helps an installation implement its security policy by identifying and authenticating virtual machine access, controlling each virtual machine's access to sensitive data, and logging and reporting events that are relevant to the system's security.

RACF verifies virtual machine logon passwords (which are stored using a one-way strong-encryption algorithm) and checks access to z/VM resources such as minidisks, data in spool files, network nodes, shared segments, Guest LANs, Virtual Switch, and some system commands. You can use RACF commands to audit security-relevant events. Events you can audit include:

- *Any CP command or DIAGNOSE code (including privileged commands and DIAGNOSE codes).*
- *The creation, opening, and deletion of spool files.*
- *The dumping and loading of spool files through the SPXTAPE and SPTAPE commands.*
- *IUCV CONNECT and SEVER operations and certain VMCF functions.*
- *APPC/VM CONNECT and SEVER operations.*
- *The creation and deletion of logical devices.*

When running a Linux guest, such auditing may provide additional insight into the activities of the Linux guest. For example, an open source package is available for Linux on zSeries that provides an interface to some CP functions. One of the components is the hcp command, which uses the DIAGNOSE 8 interface to issue CP commands on behalf of the guest virtual machine running Linux. If desired, RACF can be used to track the execution of specific CP or DIAGNOSE commands.

z/VM provides the ability for a user who has not yet authenticated themselves to the system to do two things: send messages to users who are logged on, and access (using the CP DIAL command) virtual 3270 devices, other than the virtual console, created by a virtual machine**.**

If your security policy prohibits such anonymous access to VM terminal sessions, RACF provides facilities that can disable these functions.

z/VM Security Statement of Direction

At the time of writing IBM is in evaluation for Common Criteria (ISO/IEC 15408) certification of z/VM V5.1 with the RACF for z/VM optional feature, against the Labeled Security Protection Profile (LSPP) and the Controlled Access Protection Profile (CAPP), both at Evaluated Assurance Level (EAL) 3+.

The Common Criteria is an internationally recognized International Standards Organization (ISO) standard used by governments and other organizations to assess the security and assurance of technology products. Under the Common Criteria, products are evaluated according to strict standards for various features, such as security functionality and the handling of security vulnerabilities.

Cryptography on zSeries

zSeries servers offer specialized processors for cryptographic operations. In an ISP or ASP environment, cryptographic procedures are frequently used for highly secure TCP/IP connections between the server and a user somewhere in the Internet. Applications for firewalls, Web serving, and mail serving also have the requirement to protect data.

IBM leads the way with the cryptographic coprocessor features available on zSeries servers. The CMOS Cryptographic Coprocessor Facility (CCF) and the PCI Cryptographic Coprocessor (PCICC) have earned Federal Information Processing Standard (FIPS) 140-1 Level 4 certification, the highest certification for commercial security awarded by the United States Government. The CMOS Cryptographic Processor is supported for use by guests.

The PCICC enhanced the encryption capabilities of zSeries and S/390 servers by providing additional scalability and programmability. The PCI Cryptographic Accelerator (PCICA) feature on zSeries servers provides high levels of asymmetric encryption algorithm performance.

The PCI-X Cryptographic Coprocessor (PCIXCC) is a replacement for both the PCICC and the CMOS Cryptographic Coprocessor Facility (CCF). The PCIXCC supports highly secure cryptographic functions, use of secure encrypted key values and user-defined extensions. The PCIXCC feature is designed for FIPS 140-2 Level 4 Certification.

Crypto Express2 replaces PCIXCC and PCICA functionality through dual integrated cryptographic coprocessors. z/VM 5.1 provides z/OS® and Linux guest support for the Crypto Express2 feature. The Crypto Express2 has also been designed to meet the FIPS 140-2 level 4 criteria.

The PCICA and PCICC are supported for use by Linux on zSeries guests. The PCIXCC and Crypto Express2 are supported for use by Linux on zSeries and z/OS guests.

Newer IBM zSeries servers incorporate a standard CP Assist for Cryptographic Function (CPACF) feature that provides hardware acceleration for DES, TDES, MAC, and SHA-1 cryptographic services. Cryptographic keys must be protected by your application system, as required.

To use a cryptographic coprocessor, the guest operating system has to be able to recognize application requests for cryptography, pass the requests to the cryptographic hardware, and return the result to the application. Linux on zSeries and S/390 can use the cryptographic hardware for those asymmetric algorithms used by SSL, which results in significant performance improvements for SSL transactions.

The cryptographic coprocessors may be shared and used by any number of Linux guests. Each operation is discrete and independent of those that precede or follow it. z/VM manages the queue of requests so that a guest can see only its own request, as with a shared processor or a shared channel. The Control Program blocks any attempt by a virtual machine to access or change the master keys.

The virtualization of PCICC, PCICA, PCIXCC and Crypto Express2 adapters does not change z/VM support for the IBM CMOS Cryptographic Coprocessor Facility, which is intended for use with OS/390® or z/OS guests.

A z/VM system supports the use of all installed cryptographic options simultaneously by different guests on a z/VM system.

Best z/VM Security and Integrity Practices

Below is a list of suggestions which we believe can enhance the security and integrity of your z/VM system. Of course, the decision to implement these suggestions rests entirely with the customer.

- *After installing a new z/VM system, remember to change the default logon and minidisk passwords for all users in the system directory.*
- *Don't give virtual machines more authority than they require.*
- *Review all special privileges on a regular basis. If special privileges are no longer required, remove them.*
- *Use an External Security Manager such as IBM RACF/VM.*
- *Use a z/VM directory management product such as IBM DirMaint.*
- *Implement a password management policy, including password expiration and change requirements. An ESM together with a directory management product can provide the needed policy enforcement tools.*
- *IUCV ALLOW and IUCV ANY statements in the VM directory. They are powerful permissions that permit unrestricted use of IUCV. Explicit IUCV authorizations may be more appropriate.*

- *Instead of granting users special privileges, consider using an intermediate server to process the request (sent as a message, for example) and issue the command on behalf of the user. This provides centralized control and allows permissions to be given on a more finely-grained basis. The Programmable Operator (PROP), included with z/VM, provides the necessary elements needed to build a system automation solution tailored to your needs.*
- *Review journaling and audit reports on a regular basis. Look for problems*

While implementing the action items in this list will make your z/VM more secure, it is not an exhaustive list of all possible actions you can, or should, take. Such a list can only be compiled by careful examination of the environment, the users, and the applications, and assessing them against your company's information technology security policy.

Further guidelines are documents in the z/VM CP Planning and Administration manual.

Linux Security

The way an individual Linux server should be protected is highly dependent upon the server's purpose. You need to consider what kind of access to the server is required, what exposures have to be taken into account, what kind of security attacks can be expected, and which tools to use to maintain the security of the system.

It is outside the scope of this document to describe all the techniques and tools available to implement security policies on Linux itself. In this section we point to other sources of information and recommendations for keeping your Linux servers highly secure.

Summaries of basic Linux security measures can be found in the two ITSO Redbooks Linux for S/390 and Linux for zSeries and S/390: Distributions.

The Linux on zSeries Security White Paper gives a comprehensive overview on a complete and cost-effective security infrastructure for the Linux environment, describing the products and service offerings of IBM and its partners. This is available at:
ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130488.pdf

Another paper, Addressing Security Issues in Linux, presents a broad overview of the various security issues in Linux installations. This very useful paper discusses the most common tools and utilities for increasing the level of security, and refers you to various other URLs for further information. It is available on the Web at:
ibm.com/developerworks/linux/library/l-sec/

The Security Implications of Open Source Software discusses the prejudices against open source software that often inhibit the use of such applications in production environments. The paper asserts that an open source operating system need not be insecure and that the

availability of the source code, in fact, provides for a more secure system. The extra security is claimed to be the result of many persons running and testing the code, fixing bugs or security exposures with extraordinary speed. This paper may be found at:
ibm.com/developerworks/linux/library/l-oss.html

A list of current Linux Security Certifications can be found at:
ibm.com/security/standards/st_evaluations.shtml

But remember that your virtual Linux servers are only as secure as you make your z/VM hypervisor. Take the time to study and understand the way z/VM shares resources and manages security.

References

Linux for S/390	SG24-4987
Linux on IBM zSeries and S/390: ISP/ASP Solutions	SG24-6299
Linux on IBM eServer zSeries and S/390: Best Security Practices	SG24-7023
Linux on zSeries Security White Paper	GM13-0488
z/VM Version 5 General Information	GC24-6095
z/VM Version 5 CP Planning and Administration	SC24-6083
RACF V1 R10 Security Administrator's Guide	SC28-1340
DirMaint Version 1 Release 5 General Information	GC20-1836

IBM Corporation
Marketing Communications, Server Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
05/05

GM13-0145-01