

(1)



NATIONAL COMPUTER SECURITY CENTER

AD-A207 926

FINAL EVALUATION REPORT OF COMPUTER ASSOCIATES INTERNATIONAL

CA-ACF2/VM

RELEASE 3.1

9 September 1987

S DTIC
ELECTE
MAY 23 1989
Cb **H** **D**

Approved for Public Release:
Distribution Unlimited

89 5 23 014

FINAL EVALUATION REPORT

COMPUTER ASSOCIATES INTERNATIONAL

CA-ACF2/VM Release 3.1

**NATIONAL
COMPUTER SECURITY CENTER**

**9800 Savage Road
Fort George G. Meade, Maryland
20755-6000**

September 9, 1987

**CSC-EPL-87/007
Library No. S228,570**

Final Evaluation Report Computer Associates CA-ACF2/VM

CSC-EPL-87/007
Library No. S228,570

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



FOREWORD

This publication, the Final Evaluation Report of Computer Associates International's CA-ACF2/VM, is issued by the National Computer Security Center under the authority of and in accordance with DoD Directive 5215.1, "Computer Security Evaluation Center." This report documents the results of the formal evaluation of Computer Associates' CA-ACF2/VM operating system. The requirements stated in this report are taken from *DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA* dated December 1985.

Approved:

September 9, 1987

Eliot Sohmer
Chief, Computer Security Evaluations, Publications, and Support
National Computer Security Center

ACKNOWLEDGEMENTS

Team Members

Team members included the following individuals, who were provided by the following organizations:

**Mark D. Gabriele
Joseph Bulger
Rick Siebenaler**

**National Computer Security Center
Fort George G. Meade, MD 20755-6000**

R. Leonard Brown, Ph.D

**The Aerospace Corporation
El Segundo, CA 90245**

Further Acknowledgements

Technical support was also provided by Bruce Crabill, Unisys Corporation.

Final Evaluation Report Computer Associates CA-ACF2/VM

	Foreward	iii
	Acknowledgements	iv
	Executive Summary	vii
Section 1	Introduction	1
	Evaluation Process Overview	1
	Document Organization	2
Section 2	System Overview	3
	Hardware Architecture	3
	Software Architecture	7
	The TCB	9
	TCB Protected Resources	10
	TCB Protection Mechanism	13
	Trusted Processes	16
Section 3	Evaluation as a C2 system	25
	Discretionary Access Control	25
	Additional Requirement (B3)	26
	Object Reuse	26
	Identification and Authentication	27
	Audit	30
	System Architecture	32
	System Integrity	33
	Security Testing	34
	Security Features User's Guide	36
	Trusted Facility Manual	37
	Test Documentation	38
	Design Documentation	40
Section 4	Evaluators' Comments	41
Appendix A	Evaluated Hardware Components	A-1
Appendix B	Evaluated Software Components	B-1
Appendix C	Bibliography	C-1

EXECUTIVE SUMMARY

The security protection provided by Computer Associates' Access Control Facility 2/Virtual Machine (CA-ACF2/VM) add-on package release 3.1 running with IBM's Virtual Machine/System Product (VM/SP) Release 4.0 or IBM's Virtual Machine/System Product High Performance Option (VM/SP HPO) Release 4.2 operating system has been examined by the National Computer Security Center (NCSC). The security features of CA-ACF2/VM were examined against the requirements specified by DoD Trusted Computer System Evaluation Criteria (the Criteria), dated December 1985, in order to establish a candidate rating.

The NCSC evaluation team has determined that the highest class at which CA-ACF2/VM could satisfy all the specified requirements of the Criteria is class C2. CA-ACF2/VM, using the specified hardware and software (see Appendices A and B), configured and operated in the most secure configuration as described in the Trusted Facility Manual, has been assigned a class C2 rating. Note that this evaluation does not include the operating systems on the individual user virtual machines or objects protected by those operating systems.

A system that has been rated as being a C2 system provides a Trusted Computing Base (TCB) that enforces a finely grained discretionary access control mechanism and ensures that individual users are accountable for their actions through login and auditing procedures.

Computer Associates' security computer systems;

10/1/87

INTRODUCTION

In November 1985, the National Computer Security Center (NCSC) began a developmental product evaluation of Computer Associates' Access Control Facility 2/Virtual Machine (CA-ACF2/VM) add-on package. The objective of this evaluation was to evaluate CA-ACF2/VM running with either IBM's Virtual Machine/System Product (VM/SP) or IBM's Virtual Machine/System Product High Performance Option (VM/SP HPO) against the Criteria, and to establish a candidate rating for the product. The version of CA-ACF2/VM included in this evaluation is Release 3.1, and the versions for VM/SP and VM/SP HPO are Release 4.0 and Release 4.2, respectively.

Material for this report was gathered by the NCSC CA-ACF2/VM team through system documentation and interaction with system developers.

Evaluation Process Overview

The Department of Defense Computer Security Center was established in January 1981 to encourage the widespread availability of trusted computer systems for use by facilities processing classified or other sensitive information. In August 1985 the name of the organization was changed to the National Computer Security Center. In order to assist in assessing the degree of trust one could place in a given computer system, the DoD Trusted Computer System Evaluation Criteria was written. The Criteria establishes specific requirements that a computer system must meet in order to achieve a predefined level of trustworthiness. The Criteria levels are arranged hierarchically into four major divisions of protection, each with certain security-relevant characteristics. These divisions are in turn subdivided into classes. To determine the division and class at which all requirements are met by a system, the system must be evaluated against the Criteria by an NCSC evaluation team.

The NCSC performs evaluations of computer products in varying stages of development from initial design to those that are commercially available. Product evaluations consist of a developmental phase and a formal phase. All evaluations begin with the developmental phase. The primary thrust of the developmental phase is an in-depth examination of a manufacturer's design either for a new trusted product or for security enhancements to an existing product. Since the developmental phase is based on design documentation and information supplied by the industry source, it involves no "hands on" use of the system. The developmental phase results in the production of an Initial Product Assessment Report (IPAR). The IPAR documents the evaluation team's understanding of

Final Evaluation Report - Computer Associates CA-ACF2/VM
Introduction

the system based on the information presented by the vendor. Because the IPAR contains proprietary information, distribution is restricted to the vendor and the NCSC.

Products entering the formal phase must be complete security systems. In addition, the release being evaluated must not undergo any additional development. The formal phase is an analysis of the hardware and software components of a system, all system documentation, and a mapping of the security features and assurances to the Criteria. The analysis performed during the formal phase requires "hands on" testing (i.e., functional testing and, if applicable, penetration testing). The formal phase results in the production of a final report and an Evaluated Products List entry. The final report is a summary of the evaluation and includes the EPL rating which indicates the final class at which the product successfully met all Criteria requirements in terms of both features and assurances. The final report and EPL entry are made public.

Document Organization

This report consists of four major sections and three appendices. Section 1 is this introduction. Section 2 provides an overview of the system hardware and software architecture. Section 3 provides a mapping between the requirements specified in the Criteria and the CA-ACF2/VM features that fulfill those requirements. Section 4 provides a brief summary of the findings of the formal evaluation. The appendices identify specific hardware and software components to which the evaluation applies, and provide a bibliography for this report.

SYSTEM OVERVIEW

Hardware Architecture

Computer Associates' CA-ACF2/VM is an add-on security package for IBM's Virtual Machine (VM) operating system. The VM operating system runs on IBM System/370 computers. Many different machines share this architecture (see Appendix A). This system overview will discuss the System/370 hardware and some of its main architectural features in order to provide a basis for understanding the CA-ACF2/VM system..

The System/370 series are two-state machines which support paged memory. They can operate in either problem state or supervisor state. When in supervisor state, the process may address any area of memory and use privileged instructions. Executing in supervisor state effectively gives the process access to all of the machine's resources. When operating under VM, no non-privileged user may execute in supervisor state.

Hardware-Supported Memory Protection Mechanisms

The System/370 series allows the user the potential to directly address 2^{24} (=16M) bytes of memory. It also allows for a virtual memory mechanism in order to expand and protect the available memory. This mechanism is called Dynamic Address Translation (DAT). DAT provides the system the ability to page. It also provides a virtual memory facility which is used by the VM Control Program as its primary means of self-protection. Each user's virtual address space is kept separate from those of other users, and that of the VM operating system, via this mechanism. The effect is that a process may be interrupted at an arbitrary moment, be saved (with its data) in auxiliary storage, and at a later time be returned to different main storage locations and continue execution. The address translation tables are maintained in storage which is controlled by VM and is unaddressable by unprivileged users.

Segment protection is another memory protection device. It is used to protect DisContiguous Shared Segments (DCSSs) from modification. A segment is a contiguous block of memory either 64K bytes or 1M bytes in size (this is determined by bit 11 of control register 0). If the segment protection bit in the segment table entry is turned on, then the segment of memory corresponding to that table entry is protected from attempts to store information. That is, a segment of memory may be made read-only by setting the segment protection bit of the segment table entry for that segment of memory. If an attempt is made to write data into a protected segment, a program interrupt for protection occurs, and the protected region of memory remains unchanged. Note: This mechanism is not available on some processor models; however, on these processors, each 2K page of any DCSS which is in use is checked for alteration each time the process is paged out of memory. If the page has been altered, an uncorrupted copy of that page is brought in from secondary memory.

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

The effect is the same; the difference is that testing each 2K page of any DCSS which a process has in use, every time that process ceases instruction execution, is much less efficient than the hardware segment protection mechanism.

The System/370 series computers support two memory protection mechanisms in addition to DAT and segment protection. These are low-address protection and storage-key protection. Neither of these features is used by the VM operating system, but they are included here for completeness.

Each page of memory in a System/370 machine may be protected by a storage key. Page size may be either 2K or 4K bytes; this is controlled by bits 8 and 9 of control register 0. Storage keys are 7-bit tag fields which are used to mark full or half-pages, respectively, and protect them from modification or reference by unauthorized processes. The ability to set storage keys is a privileged function; storage keys are not part of addressable memory. The operating system manages storage keys so that they are set correctly, and so that one user may not deliberately set the process storage key in order to gain access to another user's data. Storage keys allow some flexibility in protection. Each page may be marked as being read-only, read-write, or no access, depending upon the bit settings. When the storage key of the target page matches the key presented by the user, access to the page is granted according to the setting of the fetch-protection bit. Additionally, a reference bit and change bit are set if the page has been referenced or modified, respectively, since it was allocated or since the last time the Reset Reference Bit (RRB) instruction was issued for that page of storage.

The last of the hardware memory protection features in the System/370 series architecture is the low address protection mechanism. This provides protection against the destruction of information used by the CPU during interrupt processing by prohibiting the action of storage instructions which attempt to address locations 0 through 511 in real memory. The enforcement is based upon a toggle bit in control register zero. If an attempt is made to write to an address in the range of 0 through 511 inclusive and the low address protection bit is turned on, a program interrupt for protection occurs, and the protected region of memory remains unchanged. Note that such protection does not occur for access made by the CPU or a channel for such purposes as updating the interval timer, interrupt servicing, initial program loading (IPL), or I/O data transfer, as none of these mechanisms are subject to the low address protection checks. It does occur when a user program attempts to explicitly store data at any location in this range.

I/O Interface

The System/370 machines interface to all I/O devices through the use of a control unit. Different control units service different device types. The control unit may be housed with the I/O device, in a physically distinct unit, or in the CPU. In any case, from a programming point of view, most control unit functions merge with I/O device functions. In order to increase efficiency, I/O controllers do not communicate directly with the CPU; this interface is managed by devices called "channels."

A channel, like a control unit, maybe an independent unit or integrated with the CPU. In either case, all functions performed by the channel are identical. Differences may be found in the maximum data transfer rate depending upon the implementation, however. Three types of channels exist: selector, byte multiplexer, and block multiplexer.

The selector channel is the simplest in function. It must always transfer data in "burst" mode, which means that it communicates with one and only one device at a time. While one device is selected, all other devices connected to that channel must wait to communicate with the channel until they have been selected.

A byte multiplexer channel can operate in either burst mode or "byte multiplex" mode. In byte multiplex mode, due to the channel's time-slicing abilities, the channel is able to switch rapidly between I/O devices under its control. The byte multiplexer channel may have many "subchannels" which are each capable of acting as a channel to the I/O device to which they may be connected.

The block multiplexer channel may operate only in burst mode, but it can support a number of subchannels.

I/O may be initiated and controlled by two different instruction types: CPU instructions and Channel Command Words (CCWs). Instructions are decoded by the CPU and are part of CPU programs, just like any other CPU instruction (e.g., Add, Subtract, Load, Store, etc.). A CCW is decoded and executed by the channel or I/O device itself. These initiate I/O operations such as reading and writing. One or more CCWs arranged in sequence may form a channel program. Instructions and CCWs are used together to control I/O in the following fashion:

A CPU program initiates I/O operations by executing a Start I/O instruction or a Start I/O Fast Release instruction (SIO or SIOF) which identifies the channel to be used and causes the channel to fetch the Channel Address Word (CAW) from a fixed location in CPU memory. The CAW designates the location in storage from which the first CCW will be fetched. The CCW specifies the command to be executed and the storage area to be used (if any storage is required). The storage area specified by a CCW is a real address in main memory which is not subject to address translation. One CCW may "chain" to another CCW, and by this method channel programs are written. Channel

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

programs give direct instructions to I/O devices, and may cause an I/O device to do anything within its realm of capability. For example, a channel program may erase a disk volume, or read the entire contents of a tape volume into a designated buffer. Channel programs are therefore very relevant to system security. However, when the system is configured in a secure fashion, a non-privileged virtual machine is not capable of executing a real channel program. VM will translate the channel program presented by the user's virtual machine into a real channel program whose addressing capabilities are restricted to an area entirely within the boundaries of the user's virtual devices.

Interrupt Handling

System/370 machines support six different types of interrupts. These are: external, machine check, I/O, program, restart, and supervisor call (SVC). Whenever an interrupt is encountered, the current Program Status Word (PSW) is saved to a specific location in memory, and a new PSW associated with that interrupt is loaded. Each of these interrupt types will be discussed from the standpoint of its relevance to system security. The VM operating system is capable of servicing each of the interrupts discussed, and "reflecting" relevant interrupts back to the individual virtual machines as necessary.

External interrupts are generated by various conditions, including emergency conditions, from outside the computer itself. External interrupts may reach the CPU only via a hardware connection to the machine or one of its peripherals. External interrupts of this type are generated by elements within the TCB boundary. Timers within the processor may also cause external interrupts. These timers are managed by the Control Program (CP) and are used either for CP's own purposes internally, or for the simulation of virtual machine timers. A non-privileged user may not make use of real external interrupts; however, virtual external interrupts may be generated on a virtual machine by executing the EXTERNAL CP command.

Machine check interrupts are generated by the machine itself to signal equipment malfunctions. These come from within the machine and are handled within the TCB boundary. An untrusted user has no way of generating them, and this type of interrupt cannot be handled by a virtual machine operating system.

I/O interrupts are generated by channels presenting data to the CPU. The CPU must have enabled I/O interrupts for any channel from which it expects to receive an interrupt. The ability for a channel to interrupt is determined by mask bits in the PSW and in control register 2. These bits may be set only by a process running in real supervisor state. I/O interrupt handlers reside in the TCB. When a real I/O interrupt occurs, it may be reflected to the virtual machine which initiated the I/O, so that the virtual machine may continue to process the data.

Program interrupts have many different causes. However, program interrupts do not cause the machine to enter an insecure state. Program interrupts occur when a user executes a program which attempts to do something which the machine does not allow, such as execute a privileged instruction, address beyond the legal boundaries for that process, or cause an arithmetic error such as an overflow or divide by zero exception. When a program interrupt occurs, control is returned to the virtual machine operating system, with the same type of program interrupt reflected to the virtual machine operating system.

Restart interrupts are keyed in from the front panel of the physical machine. They instruct the machine to resume processing at a location which is stored at real memory location 0. Since a restart interrupt must be keyed in from the physical machine's console, which is assumed to be physically secure, this is not considered a threat to system security. Restart instructions may be simulated for a virtual machine by use of the CP command SYSTEM RESTART.

The most powerful form of interrupt is the supervisor call. SVC interrupts are non-maskable and are available to processes running in either virtual supervisor state or virtual problem state. An SVC instruction is followed by an argument, which specifies the service that the process is requesting from the virtual supervisor. The VM operating system will not service a real SVC instruction. Instead, a user may define an SVC table which may be used only by that user's virtual machine. Consequently, a non-privileged user virtual machine may not access the real SVC functions available to CP. The one exception is that of SVC 76, which allows a virtual machine to pass I/O error information to CP for recording in CP's error logging area.

Software Architecture

Computer Associates' CA-ACF2/VM release 3.1 is an add-on security package for IBM's Virtual Machine/System Product and Virtual Machine/System Product High Performance Option (generically referred to in this report as VM) operating systems. Since the VM/SP and VM/SP HPO operating systems are so similar, they will be discussed together for the remainder of the report. VM operating systems are virtual machine operating systems which present each user with a virtual IBM System/370 computer.

It is important to note that the VM systems offer the ability for a designated virtual machine to have some pages of their storage directly assigned to specific hardware pages. This option is referred to as Virtual=Real. It allows for a system administrator to select one frequently used virtual machine, choose a page range which he or she feels would be more efficiently kept in real memory at all times, and avoid having those pages swapped out of real memory. This avoids the overhead incurred in swapping commonly used pages in and out of memory, and increases system performance. Address translation is not used for these pages, with the exception of page zero. Page zero of the Virtual=Real virtual machine can never be positioned at real page zero. Options which are frequently used along with Virtual=Real are the Preferred Machine Assist (PMA), the CP SET

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

NOTRANS ON command, and the CP SET STBYPASS VR command. PMA, which is only available in VM/SP HPO, allows a Virtual=Real virtual machine to execute privileged instructions in real supervisor mode, rather than have CP simulate their execution. The SET NOTRANS ON command allows the Virtual=Real virtual machine to execute real channel programs without address translation. The SET STBYPASS VR command allows the Virtual=Real virtual machine to bypass the "shadow" page tables which CP maintains for each virtual machine which is operating in extended control mode. In order to operate the system in a C2 configuration, these options must be set as described in the system's Trusted Facility Manual (TFM). The exact specifications for each virtual machine as it is created for the user are kept in the VM directory. In the VM operating system, everything at the user interface is virtual. Virtual disks, card readers, printers, etc., are all provided by the operating system, or more specifically by the Control Program.

The Control Program

The Control Program (CP) is the VM supervisor underlying the individual VMs. CP presents each user with a bare virtual System/370 machine which is then capable of running any IBM System/370 operating system. The most commonly used operating system is CMS, the Conversational Monitor System. CMS is used in conjunction with CP to provide a workable user interface. Although CA-ACF2/VM requires the use of some CMS routines in its C2 configuration, a user may be running any other operating system on his or her virtual machine and still be protected by CA-ACF2/VM. CA-ACF2/VM will still function correctly because it owns its own copy of CMS. If a user attempts to modify or corrupt CMS, the user is given a copy of the corrupted CMS, while CA-ACF2/VM still uses the "clean" copy of CMS.

CP always executes in the real machine's supervisor state. Any time a virtual machine is executing, the instructions are being executed in real problem state. Should a virtual machine attempt to execute a privileged instruction, CP will determine whether the process was executing in virtual problem state or virtual supervisor state. If the process was in virtual problem state, CP returns a privileged instruction exception to the virtual machine; if the user was in virtual supervisor state, CP simulates execution of the instruction and returns control to the virtual machine.

VM Directory

The VM directory maintains a list of all userids known to the VM system and their VM privileges; it also defines all of the system's non-removable Direct Access Storage Device (DASD) space, how it is allocated (minidisk, temporary disk, system storage), and the virtual machine to which the DASD space is assigned. In short, the VM directory controls system resource allocation for all users on the VM system. The VM directory requires frequent updating for any desired change in system resource allocation. The VM directory exists in two forms; the first is a human-readable,

source version, and the second is a machine-readable, control block form which is used by the VM system for all online directory-related actions. In order to change the VM directory, a change is made to the source version; this change is then processed through the CP module DMKDIR, and brought online.

CA-ACF2/VM Intercepts

The CA-ACF2/VM package interfaces to VM by installing intercepts in the CP nucleus. This has the effect of front-ending certain CP routines with CA-ACF2/VM routines. Thus, when a security-relevant CP routine is called, CA-ACF2/VM is invoked. CA-ACF2/VM then validates the action, takes whatever action is necessary according to its rule sets, and returns control to the point at which the CA-ACF2/VM routine was invoked. The CA-ACF2/VM package operates in this manner for all resources it protects.

CA-ACF2/VM Databases

The CA-ACF2/VM database system is the heart of all of the control mechanisms. There are three separate record databases: the Rules database, where the CA-ACF2/VM access rules reside (these access rules resemble traditional access control lists, but are much more flexible, and allow for determining access based upon time of day, physical location of user, etc.); the Logonid database, which contains the CA-ACF2/VM user LOGONID records; and the Infostorage database, which contains scopelist and command limiting records (see TCB Protection Mechanisms).

The TCB

The trusted computing base for the CA-ACF2/VM system consists of the following:

- CA-ACF2/VM release 3.1, with:
- DIRMAINT release 2.0, Program Update Tape (PUT) 8704,
- VM Batch Subsystem, Release 1, Modification 5,

and either:

- VM/SP release 4.0, PUT 8704,

or:

- VM/SP HPO release 4.2, PUT 8704.

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

TCB Protected Resources

There are five sets of TCB protected resources that will be covered here: subjects, protected objects, interprocess communications, the DIAGNOSE codes, and CP commands.

Subjects

There is only one type of subject supported by the VM operating system: the virtual machine. All user processes and all trusted processes are virtual machine-based. CP creates virtual machines by generating a process control block called the VMBLOCK for each virtual machine, then filling the VMBLOCK with various user attributes (including such things as logonid, virtual registers, pointers to the virtual machine's virtual storage, pointers to the virtual machine's device control blocks, etc.). Once a process has had a VMBLOCK assigned to it, it is capable of acting as a virtual machine on the VM system.

Protected Objects

The protected objects of the CA-ACF2/VM system include the following: minidisks, temporary disks, tape volumes, DASD volumes, server virtual machines, spool files, VMSAVE areas and a virtual machine's virtual memory (InterProcess Communication (IPC) buffers are also protected objects, but will be discussed separately). When the CA-ACF2/VM system is configured in a trusted manner, all objects in the system are protected. One of the crucial object protection mechanisms is object reuse. The object reuse protection mechanism guarantees that a storage object allocated to a user contains no data for which that user is not authorized. This occurs by ensuring that all storage objects are cleared of residual data before they are reallocated. Following is a description of each of the protected objects, as well as a description of the object reuse protection mechanism for each object.

A minidisk is comprised of one or more consecutive cylinders on a real storage device. A minidisk can be defined to be a virtual disk drive for use by a virtual machine. A given virtual machine may be defined to be the owner of zero or more minidisks and may also link to minidisks which it does not own. Linking to a minidisk may allow the virtual machine to access the minidisk as if it were the owner. The CA-ACF2/VM package provides each user the ability to write rules describing how his minidisks may be accessed by other users. Allowable access modes are read only, read/write, or no access. DIRMAINT is the object reuse protection mechanism for minidisks, and is invoked when a minidisk is released to the system. DIRMAINT destroys all data on the minidisk and formats it for use by the next user. A detailed description of the operation of DIRMAINT can be found in the Trusted Processes section of this report.

Temporary minidisks (T-disks) are minidisks that are allocated for a single session only and may not be shared with other users. They expire at the end of a logon session or when a user specifically

Final Evaluation Report Computer Associates CA-ACF2/VM
System Overview

deallocates them. Upon expiration, CP destroys the temporary minidisk. This destruction involves the clearing of data from the minidisk and removal of any minidisk format information.

Tape volumes exist on magnetic tapes, which are physically mounted by a system operator at the request of a user. When the user is finished with the tape volume, the operator must manually clear the volume. To clear the volume, the operator must either erase the data from the tape or demagnetize the tape to remove its data. The operator must also remove the rule entries which allow the user access to the volume. This manual clearing procedure is fully described in the Trusted Facility Manual (TFM).

Direct Access Storage Device (DASD) volumes exist on magnetic disk or magnetic drum. Like tape volumes, attachable DASD volumes require a manual mount and dismount by the system operator. To clear DASD volumes, the operator uses the CP format/allocate program, IPL FMT. Access rules which pertain to the specific DASD volume must also be deleted upon deallocation of that volume. As is the case with magnetic tape volumes, this procedure is described in the system's TFM.

Spool files are used to transfer data between a virtual machine and a virtual device. A user may spool his output to a virtual printer, for example. A sharing of resources occurs when a user spools data from a virtual output device on his or her machine to a virtual input device attached to another user's machine. Spool files are defined in increments of 4K virtual memory pages, with the size of the file being indicated by an end-of-file pointer. Multiple 4K pages can be allocated to a given spool file to extend its size. The virtual memory for spool files is owned by CP and resides on a system-owned DASD. When a spool file is deallocated from a user, the storage area that was allocated to it is freed for allocation to new spool files. When a new spool file is allocated CP overwrites as many 4K memory pages as necessary, clearing the page of residual data, and sets the size of the last page with an end-of-file pointer.

When the system abnormally terminates a virtual machine, the contents of the virtual machine may be dumped into a VMSSAVE area. The VMSSAVE area may be loaded at a later time to restore the terminated session. A user releases and clears a VMSSAVE area by issuing the SET VMSSAVE OFF or LOGOFF commands. Only when an area has been cleared and released in this manner is it available for other users.

Each virtual machine's memory is also considered to be a storage object. Object reuse for virtual storage is handled by the VM paging mechanism. Fixed-length pages of 4K extent are swapped in and out of memory. When a page is brought in, the previous page is overwritten and cleared. A user may modify the amount of virtual storage allocated to him- or herself, up to a limit defined by the system administrator. When the size of a virtual machine's virtual storage is changed, a virtual system reset occurs, and all of that virtual machine's virtual storage is cleared.

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

InterProcess Communication

VM allows five types of InterProcess Communication (IPC), which are: the DIAL command, Inter-User Communications Vehicle (IUCV), Virtual Machine Communication Facility (VMCF), virtual Channel-To-Channel Adapters (CTCA's), and Discontiguous Shared Segments (referred to as shared virtual storage, or DCSS). The DIAL command is a method of connecting directly to another user's virtual machine; VMCF and IUCV both work by defining buffers and passing those buffers between virtual machines; CTCA's establish virtual connections between virtual machines which, after their definition, may be used in any fashion desired; and DCSS's set up common areas of main memory which are then used by several cooperating virtual machines as common data areas.

The purpose of the DIAL command is to allow a user to configure a server virtual machine - e.g., one user may decide to share his or her virtual machine with members of his or her project group. The server virtual machine would be logged in, set up, and left running by the owner. Anyone who wished to use the server virtual machine would DIAL a connection to that machine. The DIALer is prompted for ID and password, and a CA-ACF2/VM access rule is checked to ensure that the user is authorized to make the connection. Once the connection has been made, any action which is taken by the server machine is logged against the owner of the server machine. The system administrator must specify that a given virtual machine may be accessed via the DIAL command, and who may execute the DIAL command. The object reuse requirement is not applicable on server virtual machines since the server machine is not a storage object.

The VMCF is another method by which virtual machines may communicate. Both the sending and receiving virtual machines must execute the VMCF AUTHORIZE command before communications are initiated, and a CA-ACF2/VM generalized resource rule is checked to insure that the connection is allowed. Both virtual machines use a buffer to perform the VMCF function. These buffers are allocated within the address space of each virtual machine, and are cleared in the same manner as the virtual memory of the virtual machine.

The IUCV operates after both the sending and receiving users have logged in. Each must issue an instruction to allow IUCV communication. The CA-ACF2/VM package has the ability to audit or restrict the establishment of this communication path by a CA-ACF2/VM generalized resource rule. For communication to occur, the sender must define a reply buffer in his address space. The sender then sends an interrupt to the target process. If the target wishes to respond, that response is issued directly back into the buffer which was set aside in the sender's address space. Since the buffer exists within the address space of a virtual machine, the IUCV buffer will be cleared in the same manner as the virtual memory of the virtual machine.

Virtual CTCA's are defined by the CP DEFINE CTCA command, and connected via the CP COUPLE command. This command is used to connect a virtual channel-to-channel device to another user's device of the same type. Data is directly passed between virtual machines, without using a buffering mechanism. Use of these commands may be restricted by CA-ACF2/VM

command limiting. Discontiguous Shared Segments (DCSS) are of two basic types: protected and unprotected. Protected DCSS's are generally used to contain reentrant code and other data which may be shared by many users but should not be altered by unprivileged users. Unprotected DCSS's may be written to, as well as read from, by unprivileged users. These may be used as common areas for data sharing among groups of users. Only a system administrator may install a DCSS, and any user may be prevented from accessing DCSS's by the use of CA-ACF2/VM command and DIAGNOSE limiting rules.

DIAGNOSE

The term Diagnose is used in two manners on the VM system. Diagnose codes are used to simulate privileged commands and supervisor calls for a virtual machine. The DIAGNOSE command is used by a virtual machine to signal special requests to CP. The DIAGNOSE command uses Diagnose codes to perform its interaction with CP. Diagnose codes provide many different functions, such as examining system hardware and modifying segments. Since many of these functions are security relevant, it is necessary to limit and/or audit their usage. The CA-ACF2/VM system uses DIAGNOSE limiting rules to allow the system administrator to control the usage of DIAGNOSE codes.

CP Commands

CP commands allow users to reconfigure their virtual machines, control devices attached to their virtual machines, perform input and output spooling functions, and simulate many other functions of a real computer console. Due to the extensive capabilities of CP commands, it is necessary to restrict the usage of certain commands in a C2 environment. This is done using the CA-ACF2/VM command limiting feature. Command limiting rules allow a system administrator to write rules which allow, log, or prevent execution of any CP command, based upon the command, the logonid of the user executing the command, and the specific parameters provided with the command. The CA-ACF2/VM command limiting mechanism is a tool which allows for a high degree of customization based upon the security needs of each user of the CA-ACF2/VM system.

TCB Protection Mechanisms

VM Privilege Classes

VM supports "privilege classes" in order to enforce the principle of least privilege. These classes are used to restrict certain CP commands to certain privilege classes. VM is distributed with seven privilege classes, A through G, of which each is restricted in that no one class has the ability to execute all CP commands. These privilege classes are additive; that is, a class A user may perform only functions which would be expected of a system operator; a class G user would be restricted to

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

those commands expected of a general user of the system; a user with classes A and G could execute commands of an operator and a general user. Certain CP commands are class ANY (e.g., LOGON), which allows any user to execute them. Release 4.0 of VM/SP and release 4.2 of VM/SP HPO allow the system administrator to define 25 additional privilege classes, beyond the seven (A-G) which have been provided. The CA-ACF2/VM package strengthens this mechanism by incorporating its own set of privileges.

CA-ACF2/VM Privileges

The CA-ACF2/VM package defines its own privileges, completely distinct from the privilege classes provided by VM. These CA-ACF2/VM privileges permit security-relevant operations involving CA-ACF2/VM and its databases; however, they do not allow a virtual machine to perform any functions which would not be allowed by its VM privilege class. The CA-ACF2/VM privileges are as follows:

SECURITY: A security user may inspect and delete audit data, and may modify, create or inspect access rules, and any of the CA-ACF2/VM privileges of a user. However, a security user may not add or delete logonids to or from the system.

ACCOUNT: An account manager may add new users to the system, but may not set up any special security attributes for them. In this way, CA-ACF2/VM separates the powers of an accounts administrator from those of a security administrator.

LEADER and **CONSULTANT**: Leaders and Consultants have the ability to display most fields in the user's LOGONID record. The only noticeable differences between the two are that a Leader has the ability to change the user's phone number field and the user's suspend status. The suspend status may deny the user access to his or her account, so this is a privilege which may inconvenience a user if it is abused.

AUDIT: Audit privileged users have the ability to inspect (but not delete) audit data, and to display LOGONID records and access rules. Audit users do not have the ability to modify any components of the CA-ACF2/VM system.

Finally, there are general class users. Such users may be able to set their own passwords and a few other minor attributes. This is the class to which most CA-ACF2/VM users belong. These CA-ACF2/VM privileges may be further restricted by the use of scope modifiers.

Scope Facility

Scope controls can be defined using the SCPLIST field of the LOGONID record to limit the authority of privileged users. the SCPLIST field identifies the name of a scope list which is stored in the CA-ACF2/VM Infostorage database. Scope lists contain entries that limit the user's ability

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

to change, add, list, or delete CA-ACF2/VM records and rules. A CA-ACF2/VM privileged user that is assigned a scope list is considered restricted.

At logon, CA-ACF2/VM checks the SCPLIST field for the name of any scope record. If a SCPLIST field is found, the user is logged on under the restrictions of the scope list record. If no SCPLIST is present, then the user is unrestricted. Scopes can only be defined by an unrestricted user with the SECURITY privilege.

CA-ACF2/VM Rules

There are three types of rules in CA-ACF2/VM: command and DIAGNOSE limiting, access, and generalized resource. Command and DIAGNOSE limiting rules control availability of Diagnose codes and CP commands. Access rules determine which users can access minidisks and Direct Access Storage Device (DASD) volumes. Generalized resource rules are used to determine if a user may use the IUCV, VMCF, and DIAL facilities, as well as the valid locations and times that a user may logon to the system.

The Diagnose Limiting facility of CA-ACF2/VM allows the system to validate a user's authority to issue a diagnose instruction with a particular diagnose code. This validation occurs after normal CP user class validation for diagnose instructions. A user with the SECURITY privilege can issue a diagnose instruction with any diagnose code regardless of the diagnose limiting rules in effect. Such an event will be logged. Only a user with the SECURITY privilege can write diagnose limiting rules.

CP command limiting rules allow CA-ACF2/VM to validate a user's authority to issue a particular CP command. This validation occurs after normal CP user class validation. A user with the SECURITY privilege can execute CP commands regardless of any command limiting rules in effect, with a logging of the event. Only a user with the SECURITY privilege can write command limiting rules.

CA-ACF2/VM access rules must exist in order for a user to read or change another user's minidisk. An access rule contains information about a minidisk, indicates who may access it, and under what conditions the access is allowed. A user with SECURITY privilege may also write access rules for another user's minidisks; this event will be logged.

The CA-ACF2/VM generalized resource facility can be used to control user access to logical system resources or any other resource the installation wants to control. Some of the default system resources that are controlled by generalized resource rules include: IUCV, VMCF, DIAL, logon source locations and shift times. A user with the SECURITY privilege writes rules to control system resources.

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

Character strings in CA-ACF2/VM rules may be replaced with a character string mask. Masking allows generalized rules to be written for groups of users. Two special symbols derive the character substitution process: asterisk and dash. Asterisks are positional masking characters. They are used to indicate that any character, including a blank, may be substituted on a one-to-one basis for the asterisk or asterisks. The asterisk at the end of a string matches a null character as well as any other single character. A dash is treated as a masking character when it is on the end of a UID string or device address (e.g., SMITH-, V19-). If a dash is embedded between characters (e.g., SMITH-J), it is considered to be literally a dash, and not a mask. Use of the dash in the last position of a field mask will match any number of trailing characters in the string.

Trusted Processes

The CA-ACF2/VM system has six virtual machines that are essential to the operation of the system. These are the CA-ACF2/VM service virtual machine, the VM Batch Subsystem service virtual machines, the directory maintenance service virtual machines, AUTOLOG1, and the system maintenance and operations virtual machines. These virtual machines have access to special TCB resources, such as the CA-ACF2/VM databases, allowing them to take actions not usually allowed to normal virtual machines. These virtual machines are considered TCB components and are trusted. In this section, these virtual machines' functions will be discussed in detail.

CA-ACF2/VM Service Machine

The CA-ACF2/VM service machine (normally given the logonid ACF2VM) is a virtual machine that is automatically logged on when the first user logs on to the system after an Initial Program Load (IPL). The service machine remains active for as long as the VM system is running. The service machine has four functions: it supports and maintains the CA-ACF2/VM databases, it processes audit data recording, it performs identification and authentication, and it mediates access requests. The support and maintenance of the CA-ACF2/VM databases is discussed within the following sections covering the other three service machine functions.

Audit Function

The CA-ACF2/VM service machine maintains an audit trail of activity on the system. The audit feature of CA-ACF2/VM is dependent upon journaling performed by the CP intercept routines. The audit mechanism automatically journals invalid authorization attempts, modifications to the service machine databases, and any use of privilege. The audit function has the ability to selectively audit accesses to objects, and the execution of commands.

Selective audit occurs by including a rule in the rules database that specifies that the object access or command execution be audited. Access rules for objects may be written by the owner of the

object, and by anyone with SECURITY privilege acting within their scope. The potential exists for an owner of an object to write a rule that could supersede a rule written by a SECURITY privileged user. This situation may be controlled by activating the NOSTORE or TRACE option. NOSTORE and TRACE are fields of a LOGONID record of a user. The NOSTORE option, when selected, will prohibit a user from modifying access rules. The TRACE option will not prevent the user from modifying access rules, but will cause all security-relevant actions by that user to be logged.

These five events are automatically logged:

- . Any attempted action in violation of the command or DIAGNOSE limiting rules.
- . Invalid access attempt on a minidisk or direct access storage devices (DASD).
- . Any use of the ACFSERV command, which allows users with AUDIT, ACCOUNT, or SECURITY privilege to examine or modify the CA-ACF2/VM databases.
- . Actions performed by DIRMAINT.
- . Any other invalid authorization attempt. This includes logon attempts using a non-existent logonid, as well as logins outside shift restrictions and use of invalid passwords.

These three events can be selectively logged:

- . Any use of a command or DIAGNOSE for which a limiting rule has been entered into the information storage database, specifying that the event be logged.
- . Any access attempt on a minidisk, tape volume, or DASD device for which a rule has been entered into the information storage database, specifying that the event be logged.
- . Any use of a generalized resource, such as DIAL, IUCV, or VMCF for which a limiting rule has been entered into the information storage database, specifying that the event be logged.

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

Audit records are stored in System Management Facility (SMF) format. The resulting records may be processed by SMF file utilities, or by the audit reduction tools provided with CA-ACF2/VM. These report generation tools are comprehensive, and the reports may be analyzed online or in hardcopy. The audit log may be accessed only by a person with the AUDIT or SECURITY privilege, and destroyed only by a person with SECURITY privilege.

The CA-ACF2/VM system ensures that audit data is not lost due to an overflow of audit records to the service machine's minidisk. The default condition causes the contents of this minidisk to be spooled to the MAINT virtual machine, and possibly others as specified by a user with SECURITY privilege. The SMF files remain in spool space until processed by a user with SECURITY or AUDIT privilege. When overflow of the spool space is imminent, appropriate warnings are sent to all users on the system, and manual intervention is required. When all available space for audit recording has been exhausted, the TCB will cease to perform any security-relevant action.

Identification and Authentication

The CA-ACF2/VM service machine performs identification and authentication for the system. The service machine maintains the Logonid database, which contains a record for each virtual machine on the system. The record contains the information needed to form the User IDentification string (UID), which is used to determine whether an attempted system access should be allowed and/or audited.

A unique logonid with a maximum length of eight characters is assigned to each user by the system administrator. The characters in the logonid, or some subset of its characters, may be appended to additional identifying characters to form a conceptual UID of at most 24 characters. The format of the UID is site selectable, and is automatically enforced when a new user is added to the Logonid database.

Authentication is accomplished by requiring a password to be entered at login. The password is restricted by an installation dependent minimum and an eight character maximum (imposed by VM). The password is encrypted upon entry and this encrypted string is compared with the encrypted password stored in the LOGONID record. The minimum password length cannot be set to zero.

The CA-ACF2/VM package also provides a method whereby several users may share a specified virtual machine and still retain individual accountability. This is called the group logon facility. A system administrator identifies a given virtual machine to be shared among a number of users, who are placed on a list for that virtual machine in the Infostorage database. When a user attempts to logon to the group virtual machine, he is prompted for his personal logonid and password. Once these are provided, the user is logged on with the logonid of the group virtual machine. All access checking is performed according to the rules written for the group virtual machine, and audit records show that the group virtual machine acted on the behalf of the logged on user.

The Logonid database can be modified only by privileged users, who must use the ACF command to create or modify an entry. Users with the ACCOUNT, SECURITY, or LEADER privileges may modify certain fields, and these users may have a restriction placed on the scope of logonid values that each may modify.

The system offers two possible values for each of two password options. This presents a problem in that not all of the combinations of the options are desirable. One option, PSWDALT, indicates whether or not users are allowed to change their own passwords. The other, PSWDFRC, is used to indicate whether or not users will be forced to change their passwords when they log on if the current password was set by a user with SECURITY privilege. This results in four possible cases:

- 1) a user may alter his password and is forced to alter it after first use;
- 2) a user may not alter his password but is forced to alter after first use;
- 3) a user may alter his password but is not forced to do so after first use;
- 4) a user may not alter his password and is not forced to do so after first use.

Case 1 is the default value and is the recommendation of the TFM; Case 2 is caught by ACF2 at system configuration time, and prevented. ACF2 informs the system administrator and refuses to assemble the system when configured in this manner. Cases 3 and 4 are recommended against in the TFM.

Access Control

The CA-ACF2/VM service machine maintains the Rules database and mediates accesses to objects. The CA-ACF2/VM discretionary access control mechanism allows users to grant or deny access to objects. It uses access, command limiting, and generalized resource rules to mediate accesses between subjects (virtual machines) and named objects (minidisks, tape volumes, DASD, server virtual machines, spool files, and interprocess communication buffers).

There are different syntax formats for the three types of rules. The following provides the full syntax of the rules followed by the descriptions of all the parameters of the rules:

Access rule entry:

```
dsn      VOL(vol)      UID(uid-mask)      SOURCE(source)      SHIFT(shift)
UNTIL(date)/FOR(days)  READ(A/L/P)        WRITE(A/L/P)        EXEC(A/L/P)
DATA(data) NEXTKEY(next-key)
```

Final Evaluation Report Computer Associates CA-ACF2/VM
System Overview

Command limiting rule:

operand-mask UID(uid-mask) UNTIL(date)/FOR(days) DATA(userdata)
ALLOW/LOG/PREVENT

Generalized resource rule:

UID(uid-mask) UNTIL(date)/FOR(days) DATA(userdata)
SERVICE(READ,UPDATE,ADD,DELETE) ALLOW/LOG/PREVENT

The descriptions of the parameters for the rules are as follows:

dsn - The data set name keyword is one of the following, depending on whether the rule pertains to minidisks or DASD:

Minidisks: the virtual device address preceded by a 'V' and followed by VOLUME, such as V190.VOLUME.

DASD devices: "R" followed by the real address of the device, such as R190 for a disk drive.

operand-mask - This parameter specifies the operands of the CP commands defining the access environment.

VOL (vol) - This parameter is only valid for DASD device access rules. VOL specifies the volume serial number of the DASD volume that must be mounted on the device in order to match this rule. If omitted, any volume is considered matched.

UID (uid) - A pattern specifying the set of users to which this rule should apply. If omitted, the entry applies to all users of the system.

SOURCE (source) - A job input source, source group name, or terminal for which this rule should apply. If omitted, any input source will be valid.

SHIFT (shift) - The allowable days, dates, and times that this rule entry is in effect. If this parameter is not specified, then any access indicated by the rule will be appropriately allowed, logged, or prevented for all days, dates and times.

UNTIL (date) - A date which will be the last date on which this rule will be considered valid.

Final Evaluation Report Computer Associates CA-ACF2/VM
System Overview

FOR (days) - Number of days (between 0 and 365) that this rule will be considered valid, starting from the day the rule set was compiled.

READ (A/L/P) - This is used to specify the read access permission to be applied if there is a successful match of the dsn, vol, uid, and source parameters. "A" indicates that access should be allowed. "L" indicates that access should be allowed and logged, and "P" indicates that access should be prevented. The default is "P". Read access implies execute access.

WRITE (A/L/P) - The same as read except that it applies to write access. Write access implies read and execute access.

EXEC (A/L/P) - The same as read except that it applies to execute only access (EXECUTE only applies to CMS files which are not part of the C2 evaluation).

DATA (text) - A character string of up to 64 characters that may be meaningful in local implementations of CA-ACF2/VM.

NEXTKEY (NextKey) - The ruleid of the next rule set that should be checked for this access. If access to this dataset is denied based on the rule set environment and access permissions in the original rule, CA-ACF2/VM will then proceed to the rule specified in the NEXTKEY operand for further checking.

ALLOW/LOG/PREVENT - ALLOW specifies that the execution will be allowed; LOG specifies that the execution will be allowed, but logged; PREVENT specifies that the execution will be prevented. PREVENT is the default, if a parameter is not specified.

SERVICE (READ,UPDATE,ADD,DELETE) - This parameter specifies the type of access associated with the request.

The CA-ACF2/VM package can control access to minidisks and DASD through the use of access or command limiting rules. Users may control the sharing of their objects with individual users or groups of users by entering rules into the Rules data base. A user with SECURITY privilege may also enter access rules on behalf of another user. The SECURITY privileged user can control the user's actions by using NOSTORE or TRACE, or by writing command limiting rules on the LINK (to minidisks) and ATTACH (to DASD) commands. Access control on tape volumes and spool files is mediated by command limiting rules on the ACFMOUNT and CP spooling commands, respectively. Generalized resource rules may be written to protect server virtual machines and IPC buffers.

Final Evaluation Report Computer Associates CA-ACF2/VM System Overview

Access Validation Procedure

In a VM system, data ownership is established via MDISK statements in a user's VM directory entry. In CA-ACF2/VM, the PREFIX, which is an eight character field in the LOGONID record, specifies ownership. A user's PREFIX coincides with the PREFIXes of minidisks defined to the system via the MDISK statements in the user's VM directory. By default, the PREFIX is set to the logonid during the creation of a user's LOGONID record.

When a user requests access to a minidisk, via the LINK command or a LINK statement in the VM directory, the PREFIX is referenced to determine ownership. If the user owns the minidisk, and the RULEVLD option is not selected, then the access is allowed and no further validation occurs. The access rule is checked if the user is not the owner, or if the RULEVLD option is selected; the rule for the object will state whether the user should have access to the object. Finally, if not allowed by the rule, then the user's privilege is referenced, and if the user has the SECURITY privilege, access is allowed.

The VM Batch Monitor

The VM Batch Subsystem is a facility that allows users to execute and monitor time-consuming or non-interactive jobs in other virtual machines. Batch machines run jobs as what appear to be background processes. To use this facility, the user submits a job to the batch mechanism for execution. The batch machine adopts the job requestor's identity and access authorizations, and executes the job on the user's behalf. Since these machines can assume any user's authorizations, they must be trusted.

The VM Batch Subsystem is controlled by the VM Batch Monitor. The VM Batch Monitor is a supervisory virtual machine that dispatches and monitors other virtual machines to process the batch jobs. The VM Batch Monitor accepts batch jobs spooled from a user's virtual card punch to the VM Batch Monitor virtual card reader. At batch machine startup, the submitter's logonid and password are passed to CA-ACF2/VM for validation.

Once the VM Batch Monitor has been spooled a job for execution, it will log on one of the batch virtual machines. As this virtual machine is logged on and Initial Program Loaded (IPL'd) upon demand, there is no chance of residual data being made available to the user. This virtual machine will be given the virtual card deck as its command input, and will proceed to execute those commands. The VM Batch Monitor will grant the batch machine the exact same set of privileges that the user would be allowed.

Auditing of the VM Batch Monitor virtual machines is performed almost identically to auditing for interactive users by CA-ACF2/VM. The difference is that the audit record produced by CA-ACF2/VM shows the auditable action taken, the batch machine name, and the name of the person on whose behalf the batch machine was operating.

Directory Maintenance

The VM Directory is maintained by the use of the IBM Directory Maintenance Program Product (DIRMAINT). When this product has been installed, the VM directory is maintained by a system administrator through the use of two trusted processes, which are usually installed under the names DIRMAINT and DATAMOVE (although the actual name is site-selectable). The DIRMAINT virtual machine runs in a disconnected state, and accepts commands from authorized users via an interprocess communication channel. DIRMAINT, in turn, submits requests for actions such as clearing of deleted minidisks, minidisk size changes, etc., to the DATAMOVE virtual machine. DATAMOVE is essentially a slave to DIRMAINT, and its only purpose is to process time-consuming bulk data operations, such as minidisk copying and minidisk clearing.

The DIRMAINT program product monitors allocation of all of the non-removable DASD in use by the system. Because of this, DIRMAINT is capable of detecting and preventing possible security exposures. Any privileged function of DIRMAINT may only be executed by a trusted user.

AUTOLOG1

AUTOLOG1 is a virtual machine that plays an essential role in the CA-ACF2/VM system: it logs on the server virtual machines, and consequently causes CA-ACF2/VM to be initialized. The server virtual machines that are security relevant include DIRMAINT, DATAMOVE, and VM Batch Monitor. AUTOLOG1 itself is automatically logged on each time the VM system is IPL'd. The CA-ACF2/VM service machine is logged on when the first user, besides the operator and AUTOLOG1, attempts some action which must be mediated by CA-ACF2/VM. At this point, the VM system is initialized sufficiently for the CA-ACF2/VM service machine to begin processing.

Maintenance and Operations

The virtual machines described above run without human intervention. However, there are two other virtual machines which operate as normal console logins that have specific privileges which require that they be included in the TCB. These virtual machines are therefore trusted subjects and any user with the authority to logon to them must be trusted. These virtual machines are MAINT, which performs system maintenance, and OPERATNS, which is the operations virtual machine, where system dumps are sent. MAINT normally has privileges which allow it read/write access to all data on the system; these privileges should only be used under emergency circumstances, such as system crashes. OPERATNS normally has no special privileges, but is specified as the receiver of all system dumps. Since sensitive information may be found in a system dump, it is imperative that the OPERATNS virtual machine be considered a trusted subject.

EVALUATION AS A C2 SYSTEM

Discretionary Access Control

Requirement

The TCB shall define and control access between named users and named objects (e.g., files and programs) in the ADP system. The enforcement mechanism (e.g., self/group/public controls, access control lists) shall allow users to specify and control sharing of those objects by named individuals, or defined groups of individuals, or by both, and shall provide controls to limit propagation of access rights. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of including or excluding access to the granularity of a single user. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users.

Applicable Features

CA-ACF2/VM defines and controls access between subjects (virtual machines) and named objects (minidisks, tape volumes, attachable direct access storage devices (DASD), server virtual machines, spool files, and interprocess communication buffers) through the use of access, command limiting, and generalized resource rules.

All the objects in the system can be protected by writing one of the three types of rules. The following are the objects and the rules that can be used to control access: real devices, such as printers, tape drives, and attachable DASD may be controlled by command limiting rules; spool files may be controlled by command limiting rules; tape volumes and minidisks may be protected by access rules; logical system resources such as server virtual machines and the IPC buffers may be protected by generalized resource rules.

Access rules may be written by the owner of the object, or by a user with SECURITY privilege. Command limiting and generalized resource rules can only be written by a user with the SECURITY privilege. Any of these types of rule is capable of allowing or denying access based upon individual user identity, or group identity. In addition, access rules may allow or deny access based upon time of day, day of week, or the terminal from which the request originated. They may also be set to expire on a specified date. Protection is provided by default in that a rule must be written to permit access.

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluation as a C2 system

The CA-ACF2/VM system provides discretionary access controls for devices (such as printers) through the use of CA-ACF2/VM command limiting rules; this feature may be very useful in some environments.

Although CA-ACF2/VM documentation describes the protection of CMS files, the C2 evaluation applies only to objects of no finer granularity than minidisks.

Conclusion

CA-ACF2/VM satisfies the C2 Discretionary Access Control requirement.

Additional Requirement (B3)

The following changes are made in this requirement at the B3 level:

CHANGE: The enforcement mechanism (e.g., access control lists) shall allow users to specify and control sharing of those objects. These access controls shall be capable of specifying, for each named object, a list of named individuals and a list of groups of named individuals with their respective modes of access to that object.

ADD: Furthermore, for each such named object, it shall be possible to specify a list of named individuals and a list of groups of named individuals for which no access to the object is to be given.

Conclusion

CA-ACF2/VM satisfies¹ the B3 Discretionary Access Control requirement. Computer Associates' CA-ACF2/VM access rules are capable of supporting functional access control to this level of granularity.

Object Reuse

Requirement

All authorizations to the information contained within a storage object shall be revoked prior to initial assignment, allocation, or reallocation to a subject from the TCB's pool of unused storage

¹ Although CA-ACF2/VM satisfies this requirement at the B3 level, it does not satisfy the assurance requirements above its rated level.

objects. No information, including encrypted representations of information, produced by a prior subject's actions is to be available to any subject that obtains access to an object that has been released back to the system.

Applicable Features

The object reuse requirement applies to those objects which have a storage capability. In the CA-ACF2/VM system, storage can occur within main memory and on secondary storage devices.

Each user in the CA-ACF2/VM system accesses main memory through a virtual storage mechanism. Fixed-length pages of 4K extent are swapped in and out of memory. When a page is swapped into memory, the previous page is overwritten. If a user modifies the amount of virtual storage allocated to his or her virtual machine, a virtual system reset takes place, during which all pages allocated to the user are zeroed.

As discussed in the system overview, secondary storage devices satisfy the object reuse requirement through a variety of methods. Minidisks, temporary minidisks, and VMSAVE buffers are cleared upon deallocation. Spool files are cleared upon allocation, and an end-of-file pointer is assigned designating the size of the file. Direct Access Storage Device volumes and tape volumes must be cleared manually by the operator, upon deallocation.

Conclusion

CA-ACF2/VM satisfies the C2 Object Reuse requirement.

Identification and Authentication

Requirement

The TCB shall require users to identify themselves to it before beginning to perform any other actions that the TCB is expected to mediate. Furthermore, the TCB shall use a protected mechanism (e.g., passwords) to authenticate the user's identity. The TCB shall protect authentication data so that it cannot be accessed by any unauthorized user. The TCB shall be able to enforce individual accountability by providing the capability to uniquely identify each individual ADP system user. The TCB shall also provide the capability of associating this identity with all auditable actions taken by that individual.

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluation as a C2 system

Applicable Features

The CA-ACF2/VM service machine is responsible for the identification and authentication of users before they are allowed to access any resources that the TCB protects. Six commands are generally available to a user of a VM system before logon. They are: CP, DIAL, LOGOFF, MESSAGE, SLEEP, and LOGON.

The prelogon CP command is used to execute commands, prior to performing the logon procedure. The VM system only allows a user to execute class "any" commands with the prelogon CP command. Class "any" commands are available to a user before logon, by definition; thus the prelogon CP command does not provide any additional capabilities to the user.

The prelogon DIAL command is used to logically connect a terminal to a server virtual machine. Once the connection is made, the terminal operates entirely under the control of the server to which the virtual machine connected. The establishment of the connection between the terminal and a server virtual machine is security relevant, since the terminal now accesses a TCB protected resource.

Because of its security relevance, the system's Trusted Facility Manual (TFM) recommends that a generalized resource rule be used to prevent use of the prelogon DIAL command before system validation of a user occurs.

Use of the prelogon LOGOFF command generates the system banner page on the terminal which issued the command. The prelogon LOGOFF command does not provide any functional capability to a user.

The prelogon MESSAGE command is used to transmit a message to a specified userid or to the primary system operator. To insure that only properly authorized and authenticated users are capable of sending messages, the TFM recommends that use of the MESSAGE command before logon be disallowed.

The SLEEP command is used to place a virtual machine into a dormant state but allow messages to be displayed. The SLEEP command does not represent a security exposure.

The LOGON command is used to identify a user to the VM system and thus access the system. When a user attempts to logon to the system, he enters his logonid, which is used to form a User Identification string (UID). The UID is then compared to generalized resource rules to determine if the attempted system access should be allowed. The SOURCE and SHIFT rules can indicate information such as valid logon terminals and times. The user then enters a password. The password is encrypted and compared with a stored encrypted password. If the access rules allow entry and the password is correct, the user is allowed access to the system.

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluation as a C2 system

The LOGONID record consists of a series of fields which are used to form a UID string. These fields identify and provide information regarding user privileges and access statistics. It is also possible to define additional site dependent fields. Some of the security relevant fields of interest include:

Identification

LID - Logonid of the user.

NAME - Written name of the user. Additional contact information, such as phone number and address, may be added.

PASSWORD - Encrypted password.

UID - A pseudo field which consists of all or some subset of the LID field, plus all or part of other defined fields elsewhere in the LOGONID record. These additional fields must be defined at system installation.

Privileges

ACCOUNT, AUDIT, CONSULT, LEADER, SECURITY fields are one-bit fields that define special CA-ACF2/VM system privileges for the user with this logonid.

UIDSCOPE - Specifies which other logonid values this user may exercise privileges over.

Access Record

ACC-CNT - Number of system accesses since account creation.

ACC-DATE - Date of last access.

ACC-SRCE - Address of last input device used for login.

ACC-TIME - Time of last system access.

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluation as a C2 system

Statistics

PSWD-DAT - Date of last invalid password attempt.

PSWD-TOD - Date and time password was last changed.

PSWD-VIO - Number of password violations which occurred on PSWD-DAT.

SEC-VIO - Total number of security violations for this user.

UPD-TOD - Time this logonid was last updated.

The LOGON command also has an option which allows users to enter their password on the LOGON command line. When configured in this manner, the system does not suppress users' passwords (i.e., the system echoes the password to the screen). To provide the necessary protection, the TFM requires that the password suppression facility be used.

Conclusion

CA-ACF2/VM satisfies the C2 Identification and Authentication requirement.

Audit

Requirement

The TCB shall be able to create, maintain, and protect from modification or unauthorized access or destruction an audit trail of accesses to the objects it protects. The audit data shall be protected by the TCB so that read access to it is limited to those who are authorized for audit data. The TCB shall be able to record the following types of events: use of identification and authentication mechanisms, introduction of objects into a user's address space (e.g., terminal ID) shall be included in the audit record. For events that introduce an object into a user's address space and for object deletion events the audit record shall include the name of the object. The ADP system administrator shall be able to selectively audit the actions of any one or more users based on individual identity.

Applicable Features

The CA-ACF2/VM audit mechanism can capture all command execution, generation of output, object accesses, violations, and database modifications. The audit data is recorded as System Management Facility (SMF) files on a minidisk belonging to the CA-ACF2/VM service machine. This minidisk and the audit information is protected from unauthorized access, modification and destruction by access rules (CA-ACF2/VM DAC).

The SMF files on a minidisk are accessible to a user with AUDIT or SECURITY privilege. CA-ACF2/VM provides these users with audit log reduction tools in the form of report generators to allow the review of the audit data. There are a variety of report generators to display selected data in various formats. The following is a list of the report generators and their functions:

ACFRPTS CL - (Command Limiting Journal) - formats each command and diagnose limiting logging or violation record.

ACFRPTS CT - (ACFSERV Command Tracking) - identifies each ACFSERV command and type of command issued, and the logonid of the user who issued the command.

ACFRPTS DS - (Dataset Access Journal) - formats the violation and logging records for accesses to minidisks and DASD.

ACFRPTS EL - (Information Storage Update) - reports modifications made to the Infostorage Database.

ACFRPTS IX - (Access Index Report) - selects information regarding changes made to access rules that affect a specific user.

ACFRPTS LL - (Logonid Modification) - reports modifications made to the Logonid database.

ACFRPTS PW - (Invalid Password/Authority) - reports unsuccessful system access attempts.

ACFRPTS RL - (Rule Modification) - reports modifications made to the Rules database.

ACFRPTS RV - (Generalized Resource Event) - formats the resource violation and logging reports.

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluation as a C2 system

ACFRPTS RX - (Logonid Access Report) - reports all access rules that apply to a specific user.

ACFRPTS SL - (Selected Logonid List) - lists all LOGONID records matching the selection criteria specified.

ACFRPTS XR - (Cross-Reference Report) - provides a listing of users that have access to a specified object.

All of the records generated from the report generators include the date/time of the event, the logonid of the user responsible for the event, a reason or reason code for the success or failure of the event, and the action/command performed/executed.

Conclusion

CA-ACF2/VM satisfies the C2 Audit requirement.

System Architecture

Requirement

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering (e.g., by modification of its code or data structures). Resources controlled by the TCB may be a defined subset of the subjects and objects in the ADP system. The TCB shall isolate the resources to be protected so that they are subject to the access control and auditing requirements.

Applicable Features

The System/370 machines provide two states, problem state and supervisor state. When executing instructions within the CP, the real computer is in supervisor state. At all other times, it is in problem state.

Any attempt by a virtual machine to execute a privileged instruction will generate an interrupt which will be intercepted by the CP. The CP will then simulate the instruction if the virtual machine which caused it is in virtual supervisor state. If the virtual machine causing the interrupt is in virtual problem state, then the CP will reject the instruction and signal the interrupt back to the originating virtual machine as an exception.

VM provides fetch and store protection of all real storage at all times. VM also supports Dynamic Address Translation, which ensures that paging occurs in a secure fashion. A virtual machine may

address anywhere within its virtual storage space when operating in virtual supervisor state; however, it may not address any location outside of its virtual storage space with the exception of two well-defined, TCB-mediated events.

The first exception is the IUCV facility, which can be controlled by CA-ACF2/VM. The second of these is the DIAGNOSE mechanism. CA-ACF2/VM must be set to control execution of the DIAGNOSE instructions specified in the TFM as being security-relevant. Once these mechanisms are in place, no untrusted process will have the ability to corrupt data outside its address space.

Conclusion

CA-ACF2/VM satisfies the C2 System Architecture requirement.

System Integrity

Requirement

Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TCB.

Applicable Features

The CA-ACF2/VM security add-on package runs on IBM System/370 and compatible hardware. IBM provides software which can validate, on site, the correct operation of the hardware and firmware elements of the TCB. The IBM test suite differs from machine to machine in the System/370 product line, but it is basically as follows:

On the 308X and 3090 processors, the microcode validation routines are called VT's (validation tests). ARs, or Analysis Routines, are used once a fault has occurred in order to determine its cause.

ST370 (System Test 370) is a software diagnostic program that tests processors, channels, control units and I/O devices. 370 processor models and the 3031 processor are tested using microcode diagnostics and ST370.

ST4300 (System Test 4300) is the 43XX version of ST370. 43XX systems are tested using a combination of microcode diagnostics and ST4300.

NST (New System Test) is another software diagnostic program, which has functions similar to ST370. NST has extensive processor checks, and can be run in place of ST370. It is used in testing the 3032, 3033 and 308X processors.

Final Evaluation Report Computer Associates CA-ACF2/VM Evaluation as a C2 system

On the 3090 processors, PCX (Processor Complex Exerciser) and CSX (Channel Subsystem Exerciser) are used instead of ST370 and NST. PCX is a combination of firmware and software used to test the processor. CSX is a software diagnostic program used to test the I/O subsystem. Similar packages are available for other machines of the same architecture (see appendix B); the packages listed here apply only to the evaluated configuration.

Conclusion

CA-ACF2/VM satisfies the C2 System Integrity requirement.

Security Testing

Requirement

The security mechanisms of the ADP system shall be tested and found to work as claimed in the system documentation. Testing shall be done to assure that there are no obvious ways for an unauthorized user to bypass or otherwise defeat the security protection mechanisms of the TCB. Testing shall also include a search for obvious flaws that would allow violation of resource isolation, or that would permit unauthorized access to the audit or authentication data.

Applicable Features

System Testing

The evaluation team arranged to use one of the developmental computers at Computer Associates' Chicago Development and Support Center for system security testing. The testing required a two week session and a later one week session for completion. During the first session, system hardware integrity tests were run on the test system to assure its proper function. The team then used the information in the components of the Trusted Facility Manual, along with IBM VM/SP HPO operator manuals and system programmers' manuals, the DIRMAINT manual, and the VM Batch manual to bring up release 4.2 of VM/SP HPO, release 2.0 of DIRMAINT, and release 1, modification 5 of the VM Batch Subsystem in the manner required for operation as a C2 system. The team then installed version 3.1 of CA-ACF2/VM in the configuration required for operation as a C2 system. The Trusted Facility Manual was used to identify the proper C2 environment. Once the system had been completely installed, the team executed the entire functional test suite.

Additional Tests

Although the team inspected the test plan and is confident that the tests therein cover all security relevant events, the team ran several additional tests. These tests include the following:

The team wrote several self-modifying channel programs and determined that modifications to the channel program are made to the copy of the channel program which exists in the user's virtual storage, and that the actual channel program, which is being executed from CP-owned storage, is not affected.

The team filled the system spool space, then generated a large number of SMF journaling records in order to overflow the minidisk where CA-ACF2/VM maintains the audit files. When this situation occurs, CA-ACF2/VM handles it by attempting to spool audit log files to the system maintenance logonid. Since the system spool space was filled, this resulted in a series of messages requesting that the spool space be cleared being sent to the system operator. After all available disk space for the ACF2/VM service machine was filled with audit messages that could not be spooled, the system stopped allowing the execution of any CP command or DIAGNOSE code which was included in the CA-ACF2/VM command limiting list, as well as any event which required CA-ACF2/VM validation, such as logons or data accesses. A check of the audit data revealed that a trivial amount of audit data was lost (the event which caused the overflow and the event immediately beforehand).

The team checked that the command limiting feature of CA-ACF2/VM intercepts abbreviations of CP commands as well as their long names (for example, MSG as well as the long name, MESSAGE).

The team FORCED the DIRMAINT service machine off the system, used the editor to modify a directory entry, then brought DIRMAINT's service machine back on line. DIRMAINT detected that the directory had been tampered with and generated an audit record to that effect.

The team installed a new CP command and defined its syntax by creating a syntax model using Syntax Model Command Language (SMCL). The team then wrote and tested command limiting rules to determine that CA-ACF2/VM can successfully restrict use of user-defined commands which have been added to the CA-ACF2/VM databases.

The team inspected parameters passed during communications between the CA-ACF2/VM service machine and other machines to determine that they were either valid parameters, or were rejected if they were invalid. The team verified that the parameter checking mechanisms functioned correctly.

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluation as a C2 system

The team created a file of commands that performed some CA-ACF2/VM audit functions (audit log switching), submitted that file to VMBATCH, then inspected the SMF logging record to see that the function was correctly audited. The auditing was performed correctly.

The SMF log entry generated whenever the MESSAGE command is used includes the text of the message sent from one user to another. The team generated a message longer than the apparent largest message buffer. This resulted in the creation of two consecutive audit records, with the overflow from the first audit record being stored in the subsequent audit record.

The team determined that turning on of Virtual=Real does not prevent the system from running in a secure mode, provided that the Virtual=Real virtual machine is constrained by the rules specified in the TFM.

The team used CMS commands to allocate a T-disk, then checked that it had been zeroed before allocation by attempting to read information from the disk. The team examined the T-disk and found it to be all zeroes.

The team set up a service machine then issued a DIAL command to access it. The audit records were checked to insure that the DIAL command was properly logged, which it was.

The team set up conflicting rule sets and determined that CA-ACF2/VM correctly chose the most specific rule, thus enforcing the principle of least privilege.

Conclusion

CA-ACF2/VM satisfies the C2 Security Testing requirement.

Security Features User's Guide

Requirement

A single summary, chapter, or manual in user documentation shall describe the protection mechanisms provided by the TCB, guidelines on their use, and how they interact with one another.

Applicable Features

The CA-ACF2/VM Release 3.1 *Overview, General Information Manual, New Features and Enhancements Manual*, and *User's Guide* collectively meet the Security Features User's Guide requirements. These manuals are targeted at the system user and provide sufficient descriptions of and guidelines for the use of the protection mechanisms provided by the TCB.

These documents also provide guidance to the user on logging into and out of the system and on using the discretionary access control mechanism (access rules). The procedures and syntax for entering access rules into the Rules database are included in the CA-ACF2/VM User's Guide.

Conclusion

CA-ACF2/VM satisfies the C2 Security Features User's Guide requirement.

Trusted Facility Manual

Requirement

A manual addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility. The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit event shall be given.

Applicable Features

The *Trusted Facilities Manual for CA-ACF2/VM Release 3.1 Installations* provides for a system administrator a good description of what a C2 system is, how the TCB may be configured to run as a C2 system, and how the system may be maintained in a secure fashion. The TFM also explains the audit records which CA-ACF2/VM is capable of generating, and how different audit report generators may be used to compile specialized audit logs. In addition, the document offers suggestions about ways to decrease risk of system subversion by system operators and system programmers.

There are several other manuals which are necessary for the installation of VM, DIRMAINT, and the VM Batch subsystem. A comprehensive list of these appears in Appendix B of the *Trusted Facilities Manual for CA-ACF2/VM Release 3.1 Installations*.

Conclusion

CA-ACF2/VM satisfies the C2 Trusted Facility Manual requirement.

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluation as a C2 system

Test Documentation

Requirement

The system developer shall provide to the evaluators a document that describes the test plan, test procedures that show how the security mechanisms were tested, and results of the security mechanisms' functional testing.

Applicable Features

CA-ACF2/VM Test Documentation

The functional test of CA-ACF2/VM's security features is described in the document *CA-ACF2/VM 3.1 Test Plan*. This test plan is specifically organized to demonstrate compliance with the evaluation criteria in the *Trusted Computer System Evaluation Criteria*, particularly in the areas of object introduction and deletion, object reuse, identification and authentication, and audit logging. In fact, the current test plan evolved through four cycles during which Computer Associates developed a prospective test plan, submitted it to the evaluation team for comment, then rewrote it.

As a result of this development process, the test plan submitted by Computer Associates pays particular attention to testing the functionality of the elements of the TCB in both the CA-ACF2/VM product and in the underlying operating system. In particular, the test plan provides tests for proper functioning of CP and DIRMAINT in clearing storage objects before reuse. The VM Batch Subsystem job submission system is tested. The parts of CMS which are used by CA-ACF2/VM are also tested.

The software tests described in the test plan are not automated due to the highly interactive structure of the VM operating system and of CA-ACF2/VM itself. While some of the tests could be more fully automated than they are at present, there would always be a significant number of manual tests. The existence of at least two separate types of tests, with consequent confusion of how to handle the differences, would be undesirable; therefore, the lack of automation should not be considered a problem with the test plan.

The document has an opening discussion of the information provided above, a system requirements section which establishes the environment in which the tests must be run, and then four sets of tests. The first and fourth sets are identical, except that one set of tests is for ABORT mode and the second is for RULE mode with ABORT defaults. These are functional tests of CA-ACF2/VM features. ABORT mode is the normal system operating mode under which unauthorized access or execution attempts will be denied and logged. The use of RULE mode with ABORT defaults allows an installation to migrate rules to ABORT mode on an individual rule set basis.

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluation as a C2 system

Each of the four sets of tests has a separate environment section describing how to set up the mode, then a series of tests. The second set of tests are for identification and authentication. Both CP commands and CA-ACF2/VM functions are involved in identification and authentication, so these are tests of the interaction of CP and CA-ACF2/VM functions. These tests include those for LOGON, AUTOLOG, and DIAL commands. Group Logonid and FORCEID are also tested for proper functioning.

The third set of tests is for data integrity of the audit logs, and the individual selectivity of the audit mechanism via the TRACE attribute in a user's logonid record.

Appendix A of the test plan gives source listings for several utility programs that aid the testing team in generating test commands, the text of EXEC's used for VMBATCH tests, and CA-ACF2/VM EXEC's that facilitate testing LOGON, AUTOLOG, and Group Logon features. For example, the first test program, DIAGTEST, is an assembler language program that reads a DIAGNOSE code, tries to invoke it, and returns a message that describes one of four possible results. This is not a fully automated test since the the reduction program must be run on the CA-ACF2/VM audit log; the resulting report must be checked to make sure that the DIAGTEST program message coincided with the actual audit log entry. Other test programs facilitate testing of VMCF and IUCV.

Appendix B of the test plan provides tests for the proper working of CP functions on which CA-ACF2/VM depends for integrity of the system as a whole. These include tests for real storage protection, prevention of a virtual machine from using real supervisor state, protection against reuse of uncleared minidisks and virtual storage, and enforcement of read only LINKs to minidisks. The format of these system tests is as similar to the format of tests in parts 1 through 4 as possible, given that verification that a test worked properly may have to be accomplished by means other than inspection of the audit logs.

A typical test description starts with a checklist of commands to issue and expected responses, which comprise the test procedure. A second checklist is given which shows how to run the audit reduction program on the CA-ACF2/VM audit log files to determine that all auditable events are recorded during the test. Responses sent by the system to the terminal are included with the checklist components. A listing of all audit reduction reports is available for comparison with the reports actually generated during testing. These reports still require the testing team to determine whether the audit report properly reflects the events that appear in the test plan. Due to the clear format of these reports, this is not difficult.

Conclusion

CA-ACF2/VM satisfies the C2 Test Documentation requirement.

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluation as a C2 system

Design Documentation

Requirement

Documentation shall be available that provides a description of the manufacturer's philosophy of protection and an explanation of how this philosophy is translated into the TCB. If the TCB is composed of distinct modules, the interfaces between these modules shall be described.

Applicable Features

The CA-ACF2/VM Security System CA-ACF2/VM Release 3.1 *Program Logic Manual* is the design document for the CA-ACF2/VM add-on package. It contains the CA-ACF2/VM philosophy of protection, a detailed description of how CA-ACF2/VM interfaces with CP, and a thorough explanation of each of the CA-ACF2/VM modules included in the evaluated configuration. The descriptions of CA-ACF2/VM modules include a functional summary, inputs and outputs, the subroutine entry points, and a cross-reference listing of all modules which call or are called by each module.

The portions of the TCB which are IBM products are documented by several different IBM publications. These are listed in the CA-ACF2/VM design document, in the chapter entitled "The Base VM Operating System." In addition to the documents listed there, the team has the VM/SP and VM/SP HPO source code itself, and used the module prolog comments as documentation which helped to describe the module interfaces. Taken as a whole, the IBM documentation available for the VM operating system constitutes acceptable design documentation for a C2 system.

Conclusion

CA-ACF2/VM satisfies the C2 Design Documentation requirement.

EVALUATORS' COMMENTS

Configuration Management

Both VM and CA-ACF2/VM use the same configuration accounting system to insure that updates to the system are applied correctly. Updates to the system are contained in update files. These files are identified in an auxiliary update file in the proper order to assure that they will be applied in the order intended. The auxiliary update files are identified in a CNTRL file which also contains names of the macro libraries needed to assemble the modules that need to be updated. A command macro (EXEC) called VMFASM performs the updates and assemblies.

Each update file is identified by a unique Symptom Tracking And Reporting (STAR) number. Updates that differ between VM/SP and VM/SP HPO are shipped on identical update tapes, but the source text contains release dependent differences that are handled by the assembler.

Prior to Computer Associates' attainment of the C2 rating for CA-ACF2/VM, configuration management was done in an ad hoc manner; however, the existing configuration accounting system will support the type of configuration management system required for the Ratings Maintenance Program. In particular, steps have been taken to bring the test plan and documentation under configuration control. The manuals and code for the underlying operating system (VM/SP and VM/SP HPO) are already under configuration control by its vendor.

Mixing DAC and Audit

Since a system can be configured to allow users to write rules for their own minidisk accesses, it is possible for users to change rules permitting access to their own minidisks from LOG to ALLOW. A system set up in this fashion allows its users to control when and for whom accesses to their data will be audited, which may be undesirable in some environments. However, the system can be configured to disallow any number of users from writing rules, thus limiting this feature. Also, even on a system which generally allows this feature, individual users can be prevented from storing access rules and thus logging of all accesses to their minidisks may be assured.

DAC Flexibility

The CA-ACF2/VM discretionary access control system exceeds the requirements in the Criteria by allowing finer granularity of access control. In addition to controlling access to an object by named user or group, access can be controlled by time of day (SHIFT), day of week (DAY), by date, and by terminal (SRC) or group of terminals (GRP).

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluators' Comments

Control of Site-defined Commands

It is possible (by modifying the TCB) to add site specific commands that will be interpreted by CP. While this is not a recommended practice for a secure system, CA-ACF2/VM can be used to control use of such commands by use of its command limiting feature. The syntax of the command must be specified to CA-ACF2/VM using SMCL (Syntax Modeling Command Language). Then ordinary command limiting rules can be written to control which subjects may use the new command.

Audit Reduction Tools

The audit reduction tools that inspect the SMF records generated by CA-ACF2/VM are both comprehensive and easy to use. Different types of reports can be run on the same SMF log (or multiple logs) to isolate use of one or more commands or DIAGNOSE instructions. Reports involving subjects, such as password reports, can be run to show only log entries for a single user or group of the current test plan evolved through users. The formats of the various reports, while dependent on the type of report being run, are consistent with each other and easy to use.

EVALUATED HARDWARE COMPONENTS

The CA-ACF2/VM formal evaluation team conducted testing in Computer Associates' office in Rosemont, Illinois, using the following hardware:

IBM 4341-II processor

IBM 3330 disk drives IBM 3203 printer

Storage Technology Corporation model 4600 tape drives

Courier C270 terminals and control unit

In addition to the tested configuration, the following devices are capable of supporting a C2-secure system configuration.

Processors:

System/370 Model 135-3

System/370 Model 138

System/370 Model 145-3

System/370 Model 148

System/370 Model 155-II

System/370 Model 158

System/370 Model 158-3

System/370 Model 165-II

System/370 Model 168

System/370 Model 168-3

3031

3032

3033

3042-2

3081

3083

3084

3090 (models 120, 150, 180, and 200; model 400 when partitioned into two model 200's)

Vector processor on 3090

4321

4331

4341

4361

4381

9370 (all models)

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluated Hardware Components

Direct Access Storage Devices:

2305
2314
2319
3330
3333
3340
3345 on 370/145-3 and 370/148
3350
3310
3370
3375
3380
9332
9335

Direct Access Storage Control Units:

2835
2844
3830
3880
Integrated Storage Control on S/370 model 158 for 3330, 3333, 3340 and 3350 DASD
Integrated Storage Control on S/370 model 168 for 3330, 3333, 3340 and 3350 DASD

Magnetic Tape Drives:

2401, 2402 and 2403
2415
2420
3410/3411
3411
3420
3422
3430
3480
9809
9347

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluated Hardware Components

Magnetic Tape Control Units:

2803
2804
3411
3422
3430
3480
3803

Unit Record Devices:

1403 Printer
1443 Printer
3203 Printer 3
3211 Printer
3262 Printer (model 5 only)
3800 Printer
4248 Printer
2501 Card Reader
2520 Card Reader
2540 Card Reader
3505 Card Reader
3525 Card Punch

Unit Record Control Units:

2821
3811

Terminals and Display Stations:

2741 Terminal
1050 Data Communication System
3101 Terminal 3180 Display Station
3178 Display Station
3179 Display Station
3191 Display Station
3192 Display Station
3193 Display Station

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluated Hardware Components

3194 Display Station
3275 Display Station
3276 Control Unit Display Station
3277 Display Station, via 3271 Control Unit
3277 Display Station, via 3174 or 3274 Control Unit
3278 Display Station, via 3174 or 3274 Control Unit
3278 Display Station, via 3276 Control Unit
3279 Color Display Station, via 3174 or 3274 Control Unit
3279 Color Display Station, via 3276 Control Unit
3290 Information Panel, via 3174 or 3274 Control Unit
3767 Terminal

Consoles:

2150 Console with 1052 Printer-Keyboard
3066 System Console
3210 Console Printer-Keyboard
3215 Console Printer-Keyboard
IBM System Console for S/370 Models 138, 148 and 158
7412 Console
3036 Console
3278 Model 2A Console
3279 Model 2C Console

NOTE: Consoles are not acceptable for use as user terminals, as they may possess controls which allow a user access to the processor diagnostics and controls.

Transmission Control Units:

2701 Data Adapter Unit
2702 Transmission Control Unit
2703 Transmission Control Unit
3704, 3705-I, 3705-II and 3725 Communications Controllers
7171 Transmission Control Unit

Other Devices:

Two-channel switches

Final Evaluation Report Computer Associates CA-ACF2/VM
Evaluated Hardware Components

Other Printers :

CP is not capable of direct control of the following models of printer. These printers may be employed in an evaluated configuration, but they may only be used as devices ATTACHED to individual virtual machines by a privileged user. These virtual machines may then make exclusive use of the printer with the condition that they not act as print servers since this would bypass the system's capability to audit the production of printed output.

The printers which CP is not capable of controlling are:

3262 models 3 and 13
3268 models 2 and 2C 4250 (all models)
4245 models D12 and D20
3287 models 1, 2, 1C and 2C
3289 models 1 and 2

EVALUATED SOFTWARE COMPONENTS

The following software was evaluated:

CA-ACF2/VM release 3.1, with:

DIRMAINT release 2.0, Program Update Tape (PUT) 8704,

VM Batch Subsystem, Release 1, Modification 5,

and either: VM/SP release 4.0, PUT 8704,

or: VM/SP HPO release 4.2, PUT 8704.

This rating does not apply to any system running any other software which modifies CP, has "hooks" into CP, or is allowed to execute in real supervisor state.

Note that the evaluated configuration does not apply to any machine which is running in the eXtended Architecture (XA) addressing mode. CA-ACF2/VM and VM/SP do not at the present time support the XA mode. Machines which offer XA addressing should have this feature disabled.

BIBLIOGRAPHY

Computer Associates International, CA-ACF2/VM Release 3.1 *Command Limiting Reference Guide*, publication number AAP0074

Computer Associates International, CA-ACF2/VM Release 3.1 *Field Definition Record Generation Manual*, publication number AAP0032

Computer Associates International, CA-ACF2/VM Release 3.1 *General Information Manual*, publication number AAG0033

Computer Associates International, CA-ACF2/VM Release 3.1 *Messages Manual*, publication number AAP0038

Computer Associates International, CA-ACF2/VM Release 3.1 *New Features and Enhancements Manual*, publication number AAP0073

Computer Associates International, CA-ACF2/VM Release 3.1 *Program Logic Manual*, internal publication

Computer Associates International, CA-ACF2/VM Release 3.1 *System Programmer's Guide*, publication number AAL0035

Computer Associates International, CA-ACF2/VM Release 3.1 *Test Plan*, internal publication

Computer Associates International, CA-ACF2/VM Release 3.1 *Trusted Facilities Manual*, publication number AAP0082

Computer Associates International, CA-ACF2/VM Release 3.1 *User's Guide*, publication number AAP0037

Computer Associates International, CA-ACF2/VM Release 3.1 *Utilities Manual*, publication number AAP0036

Computer Associates International, CA-ACF2/VM Release 3.1 *VM Overview Manual*, publication number AAG0042

Department of Defense, *Trusted Computing System Evaluation Criteria*, publication number CSC-STD-001-85

Final Evaluation Report Computer Associates CA-ACF2/VM
Bibliography

International Business Machines Corporation, *IBM Virtual Machine/Directory Maintenance Program Logic Manual*, publication number LY20-0889

International Business Machines Corporation, *IBM System/370 Principles of Operation*, publication number GA22-7000

International Business Machines Corporation, *IBM Virtual Machine/Directory Maintenance Installation and System Administrator's Guide*, publication number SC20-1840

International Business Machines Corporation *Virtual Machine/Directory Maintenance Program Logic Manual*, publication number LY20-0889

International Business Machines Corporation, *Virtual Machine Batch Subsystem, Program Description/Operations Manual*, publication number SH20-2652

International Business Machines Corporation, *Virtual Machine/System Product CP Command Reference for General Users*, publication number SC19-6211

International Business Machines Corporation, *Virtual Machine/System Product Data Areas and Control Block Logic*, publication number LY20-0891

International Business Machines Corporation, *Virtual Machine/System Product High Performance Option CP Command Reference for General Users*, publication number SC19-6227

International Business Machines Corporation, *Virtual Machine/System Product High Performance Option Data Areas and Control Block Logic*, publication number LY20-0896

International Business Machines Corporation, *Virtual Machine/System Product High Performance Option Planning Guide and Reference*, publication number SC19-6223

International Business Machines Corporation, *Virtual Machine/System Product High Performance Option Service Routines Program Logic*, publication number LY20-0898

International Business Machines Corporation, *Virtual Machine/System Product High Performance Option System Logic and Problem Determination Guide*, publication number LY20-0897

International Business Machines Corporation, *Virtual Machine/System Product High Performance Option System Programmer's Guide*, publication number SC19-6203

Final Evaluation Report Computer Associates CA-ACF2/VM
Bibliography

International Business Machines Corporation, *Virtual Machine/System Product Operator's Guide*, publication number SC19-6202

International Business Machines Corporation, *Virtual Machine/System Product Planning Guide and Reference*, publication number SC19-6203

International Business Machines Corporation, *Virtual Machine/System Product Service Routines Program Logic*, publication number LY20-0890

International Business Machines Corporation, *Virtual Machine/System Product System Logic and Problem Determination Guide Volume 1 (CP)*, publication number LY20-0892

International Business Machines Corporation, *Virtual Machine/System Product System Logic and Problem Determination Guide Volume 2 (CMS)*, publication number LY20-0893

International Business Machines Corporation, *Virtual Machine/System Product System Messages and Codes*, publication number SC19-6204

International Business Machines Corporation, *Virtual Machine/System Product System Programmer's Guide*, publication number SC19-6203

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS NONE	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Public Release - Distribution Unlimited	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CSC-EPL-87/007		5. MONITORING ORGANIZATION REPORT NUMBER(S) S228-570	
6a. NAME OF PERFORMING ORGANIZATION National Computer Security Center	6b. OFFICE SYMBOL (if applicable) C12	7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State, and ZIP Code) 9800 Savage Road Fort George G. Meade, MD 20755-6000		7b. ADDRESS (City, State, and ZIP Code)	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) (U) Final Evaluation Report, Computer Associates Internation CA-ACF2/VM			
12. PERSONAL AUTHOR(S) M. Gabriele, R.L. Brown, J. Bulger, R. Siebenaler			
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) 870909	15. PAGE COUNT
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
		NCSC TCSEC ACF2 VM/SP CA-ACF-2/VM	
		370 C2 EPL	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
Security features of CA-ACF2/VM Release 3.1 were evaluated against the requirements specified for a class C2 system in the Department of Defense Trusted Computer System Evaluation Criteria. This report presents the findings of that evaluation.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL LTC Lloyd D. Gary, USA		22b. TELEPHONE (Include Area Code) (301) 859-4458	22c. OFFICE SYMBOL C/C12