National Information Assurance Partnership

TM

Common Criteria Evaluation and Validation Scheme Validation Report

IBM Rochester, MN

IBM i5/OS V5R3M0 running on IBM eServer models 520, 550, and 570 with Software Feature Code 1930

Report Number: CCEVS-VR-05-0111
Dated: 10 August 2005
Version: 1.0

National Institute of Standards and Technology
Information Technology Laboratory
100 Bureau Drive
Gaithersburg, MD 20899

National Security Agency
Information Assurance Directorate
9800 Savage Road STE 6740
Fort George G. Meade, MD 20755-6740

ACKNOWLEDGEMENTS

Validation Team

Mr. Daniel P. Faigin
The Aerospace Corporation
El Segundo, California

Mr. Stephen Butterfield
Mitretek Corporation
McLean, Virginia

The Validation Team also thanks Ms. Victoria Ashby, *MITRE Corporation*, for the work she performed when she was Lead Validator, and Mr. Kenneth Elliott for his work as Senior Validator.

Common Criteria Testing Laboratory

Ms. Tammy Compton, Lead Evaluator
Science Applications International Corporation
Columbia, Maryland

Table of Contents

1 Executive Summary

This report documents the assessment of the National Information Assurance Partnership (NIAP) validation team of the evaluation of IBM i5/OS V5R3M0 running on IBM eServer models 520, 550, and 570 with Software Feature Code 1930 (henceforth referred to as i5/OS)[1]. It presents the evaluation results, their justifications, and the conformance results. This Validation Report is not an endorsement of the Target of Evaluation by any agency of the U.S. government, and no warranty is either expressed or implied.

The evaluation was performed by the Science Applications International Corporation (SAIC) Common Criteria Testing Laboratory (CCTL) in Columbia, Maryland, United States of America, and was completed in August 2005. The information in this report is largely derived from the Evaluation Technical Report (ETR) and associated test reports, all written by SAIC. The evaluation determined that the product is both **Common Criteria Part 2 Extended and Part 3 Conformant**, and meets the assurance requirements of EAL 4 augmented with ALC_FLR.2. The product is also conformant with the Controlled Access Protection Profile (CAPP), Version 1.d, 8 October 1999. All security functional requirements are derived from Part 2 of the Common Criteria or expressed in the form of Common Criteria Part 2 requirements.

i5/OS is a complete operating system that operates on a variety of IBM iSeries hardware platforms, of which three models are covered by this evaluation. The i5/OS Operating System is object-based and, in the evaluated configuration, implements approximately 50 object types. Data access and system management is controlled via access controls on the available objects, but only after the responsible user has been identified and authenticated by i5/OS and if the user has the required authorities. Additionally, i5/OS can audit security-relevant actions, including authentication attempts, access attempts, and security management functions. This validation assumes the TOE has been configured as described in the iSeries *Configure Your System For Common Criteria Security (Version 5 Release 3)* document. Furthermore, the TOE is a subset of the appropriately configured product since the product includes a number of applications that fall outside the scope of the TOE, and hence have not been evaluated, as they can have no effect on the TOE security functions.

The Target of Evaluation (TOE) identified in this Validation Report has been evaluated at a NIAP approved Common Criteria Testing Laboratory using the Common Methodology for IT Security Evaluation (Version 1.0) for conformance to the Common Criteria for IT Security Evaluation (Version 2.1). This Validation Report applies only to the specific version of the TOE as evaluated. The evaluation has been conducted in accordance with the provisions of the NIAP Common Criteria Evaluation and Validation Scheme and the conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence provided.

[1] Note that i5/OS was previously OS/400, with the hardware being designated AS/400. The hardware was renamed iSeries in 2000. In 2004, OS/400 was renamed i5/OS to coincide with the eServer i5 models introduced in that same year. IBM has moved to the new name, but some documentation (and evaluation evidence) still refers to the older names.

The validation team monitored the activities of the evaluation team, observed evaluation testing activities, provided guidance on technical issues and evaluation processes, and reviewed the individual work units and successive versions of the ETR. The validation team found that the evaluation showed that the product satisfies all of the functional requirements and assurance requirements stated in the Security Target (ST). Therefore the validation team concludes that the testing laboratory's findings are accurate, the conclusions justified, and the conformance results are correct. The conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence produced.

The SAIC evaluation team concluded that the Common Criteria requirements for Evaluation Assurance Level (EAL 4 augmented with ALC_FLR.2) have been met.

The technical information included in this report was obtained from the Evaluation Technical Report (ETR) Part 1 (non-proprietary) produced by SAIC, the i5/OS Security Target, and analysis performed by the Validation Team.

2 Identification

The CCEVS is a joint National Security Agency (NSA) and National Institute of Standards effort to establish commercial facilities to perform trusted product evaluations. Under this program, security evaluations are conducted by commercial testing laboratories called Common Criteria Testing Laboratories (CCTLs) using the Common Evaluation Methodology (CEM) for Evaluation Assurance Level (EAL) 1 through 4 in accordance with National Voluntary Laboratory Assessment Program (NVLAP) accreditation.

The NIAP Validation Body assigns Validators to monitor the CCTLs to ensure quality and consistency across evaluations. Developers of information technology products desiring a security evaluation contract with a CCTL and pay a fee for their product's evaluation. Upon successful completion of the evaluation, the product is added to NIAP's Validated Products List.

Table 1 provides information needed to completely identify the product, including:

- The Target of Evaluation (TOE): the fully qualified identifier of the product as evaluated.
- The Security Target (ST), describing the security features, claims, and assurances of the product.
- The conformance result of the evaluation.
- The Protection Profile to which the product is conformant.
- The organizations and individuals participating in the evaluation.

Table 1: Evaluation Identifiers

Item	Identifier
Evaluation Scheme	United States NIAP Common Criteria Evaluation and Validation Scheme

Item	Identifier
TOE:	IBM i5/OS V5R3M0 running on IBM eServer models 520, 550, and 570 with Software Feature Code 1930
Protection Profile	Controlled Access Protection Profile (CAPP), Version 1.d, 8 October 1999.
ST:	*IBM i5/OS V5R3 Security Target* , Version 1.0, July 8, 2005
Evaluation Technical Report	*Evaluation Technical Report for IBM i5/OS:* • *Part 1 (Non-Proprietary), Version 4.0, August 8, 2005* • *Part 2 (Proprietary), Version 5.0, August 8, 2005*
CC Version	Common Criteria for Information Technology Security Evaluation, Version 2.1 Part 2: Evaluation Methodology, Supplement: ALC_FLR- Flaw Remediation, Version 1.1, February 2002, CEM-2001/0015R
Conformance Result	CC Part 2 conformant, CC Part 3 conformant
Sponsor	IBM Rochester
Developer	IBM Rochester
Common Criteria Testing Lab (CCTL)	SAIC, Columbia, MD
CCEVS Validators	Daniel P. Faigin, The Aerospace Corporation, El Segundo, CA Stephen Butterfield, Mitretek Systems, McLean VA

3 Security Policy

The Security Functional Policies (SFPs) implemented by i5/OS are based upon the basic set of policies provided in the Controlled Access Protection Profile. These include policies that permit protection of user data, provide for authenticated user access, provide accountability for actions, and protect the mechanism that provides the security policies.

Note: Much of the description of the i5/OS security policy has been extracted and reworked from the i5/OS Security Target.

3.1 User Data Protection

i5/OS is object oriented and implements approximately 50 object types. Each of the objects has associated operations and access modes that can be configured so that individual users and groups of users can be restricted so that they can perform only selected operations on given objects. Access to objects is controlled using *authorities* and *authorization lists*. In addition, the system also requires a user to have *special authorities* to access certain external objects through the provided interfaces.

Authorities can be granted to users, groups, and to all users. There are four types of authorities:

- **Owner authority** – Authority of the owning user profile. Each object has an associated owning user profile and by default the owning user profile is the user profile of the creating process and all authorities are granted to the owner.
- **Primary group authority** – Authority of the primary group. An object may optionally have a primary group; if it does, the primary group authority is stored with the object and not with the group profile.
- **Private authority** – Authority that is explicitly granted to a user or group profile. Private authorities to objects are stored in the user and group profiles and if both a user and group profile have private authorities for the same object, the user profile takes priority.
- **Public authority** – Authorities that apply to users that don't have explicit authority to an object. Public authorities are used only in the absence of owner, primary group, and private authorities.

There are also two categories of authorities: *object authorities* and *data authorities*. Object authorities pertain to operations that are performed on the object as a whole. These can include the ability to look at the description of an object, use an object, manage an object, reference an object, and change the attributes of object. Data authorities pertain to operations that can be performed on the contents of the object. This includes the traditional notions of read, write, and execute, as well as database specific abilities (update, delete) and the ability to specific exclude access to the data. i5/OS provides the ability to use authorities individually, as well as providing a number of predefined authority groups to simplify system management.

Authorization lists are a special type of object, and are used to assign specific authorities for different users and groups to a set of objects. All objects except profiles and authorization lists can be secured by an authorization list. Furthermore, an object can have only a single authorization list while a single authorization list can be used to secure multiple objects. The system provides authorized users with the ability to grant or revoke any authority to a given object.

Details on the process of evaluating authorities may be found in the i5/OS Security Target, as well as in IBM documentation.

All storage objects (memory, disks, workstations, optical drives, magnetic tapes and printers) used in i5/OS are cleared when they are allocated. Input/output processor and other device buffers are controlled by keeping track of how much data is present and disallowing read attempts beyond the current data. The system appears to provide no mechanisms that permit residual data to be transmitted through device status buffers.

3.2 Identification and Authentication

In the evaluated configuration, each user must provide a user name and password before they are allowed to exercise any i5/OS commands, regardless of the mechanism used to communicate with i5/OS. Once a user has been authenticated, i5/OS maintains the identity and other attributes with the resulting session to ensure proper access controls are enforced

and individual accountability is maintained. Access control is enforced for both direct user access and network access.

i5/OS defines users and groups using profiles, which can only be created or deleted by authorized administrators. Each profile is an object with field-level access controls in order to ensure that only authorized administrators can change security-relevant profile information. Security relevant fields in the profile include the profile name, password, status indicators, password expiry information, user class information, group profile information, and auditing level information.

A user who is not an authorized administrator can only change their own password through the interfaces provided by i5/OS. Security-relevant roles are provided via the user class.

i5/OS requires all users to identify and authenticate themselves before they are allowed to access system resources. Users are identified by a user profile and authenticated by a password of 6 to 128 characters. There are eleven system values that control passwords. These system values require users to change passwords regularly and help prevent users from assigning trivial, easily guessed passwords. The administrator and user guidance document provide recommendations for password construction.

A user can obtain access to the i5/OS by signing on; specifically, user authentication occurs when the following functions are used:

- AUTOSTART
- SIGNON command
- STRxxxJOB or SBMxxxJOB commands.
- FTP Sign on
- TELNET Sign on
- RUNRMTCMD

All of these functions obscure the password (i.e., do not echo it) with the exception of RUNRMTCMD[2]. In order to be successfully authenticated, the user identity must correspond with an existing user profile and the provided password must match the password stored in the profile. Additionally, the user profile must be enabled and have the required access to resources associated with the connection attempt (e.g., access to the workstation device). If the user's password has expired, it must be changed before the sign-on can be completed. Finally, if the user is signing on interactively, via workstation or TELNET, the user is provided information regarding the date and time of their last sign-on as well as the number of unsuccessful sign-on attempts since then along with the number of days before their password will expire.

[2] Note that RUNRMTCMD does not obscure the password until the user presses the enter key. This is consistent with the application note in the CAPP that states "Obscured feedback implies the TSF does not produce a visible display of any authentication data entered by a user, such as through a keyboard (e.g., echo the password on the terminal). It is acceptable that some indication of progress be returned instead, such as a period returned for each character sent. Some forms of input, such as card input based batch jobs, may contain human-readable user passwords." Appropriate cautions with respect to this are provided in user documentation.

Once a user is successfully authenticated, a process is instantiated with their user profile, which includes any group profile(s)–a special form of user profile–that the user may hold, to operate on their behalf. The security attributes contained within applicable user and group profiles are associated with the process. Any commands issued by the user subsequently execute in the context of the user's profile with the following two exceptions. A trusted subject, such as an authorized administrator, can change the user profile associated with a process thread and thereby change the security attributes. Any process can potentially augment its security attributes by calling a program that adopts authority. Such programs can be created by a user and assigned attributes such that when another user executes that program the associated process can acquire the authorities of the program's owner.

Adopted authority is added to any other authority found for the user. Only the authorities of the owner are adopted. If the owner has a group profile, the group's authorities are not considered. Adopted authority is a program attribute that is specified when the program is created. If program adoption is specified, then the authorities associated with the program owner's user profile are checked to determine whether authority is sufficient to access the object. The system may use the adopted authority from the original program the user called or from earlier programs in the program stack. If the adopted authority check locates sufficient authority, then access is granted. If the result is insufficient, then access is denied.

3.3 Security Audit

i5/OS has an audit mechanism that is invoked for access checks, authentication attempts, administrator functions, and at other times during its operation. When invoked, the date, time, responsible individual and other details describing the event are records to the audit trail.

i5/OS can be configured to halt when data can't be written to the audit log or to discard the data and continue processing. i5/OS can also be configured so that when audit log spaces (called journal receivers) fill, the system will automatically generate new ones so that audit data is not lost. The audit trail is composed of potentially many journal receivers. The audit entries associated with a filled journal receiver remain available for administrator review. i5/OS allows an Administrator to configure an audit level parameter so that only selected types of auditable events will be collected and undesired audit events will not cause the audit trail to become full unnecessarily.

Tools are provided so that an administrator can effectively review the audit trail, including searching and sorting by user identities.

The journal receivers and security journal are all objects that are protected from unauthorized access or destruction using the discretionary access control mechanism. Only an authorized administrator may create the security audit journal and a journal receiver. These objects do not exist when a system is initially installed. i5/OS provides a command to create the initial security journal receiver and the security audit journal in such a way that non-administrators are prevented from accessing the audit data, and with attributes that

ensure, when the receiver is full, that transition to a new receiver occurs without loss of audit data.

Each i5/OS component that implements security-relevant functions is responsible to collect the necessary data and to call the i5/OS auditing routines. The auditing routines will only send audit data to the journal receiver if auditing is active and the security action is pre-selected.

There are numerous security relevant events that are auditable by i5/OS, including but not limited to those identified in the Controlled Access Protection Profile. Each recorded event includes the date and time of the event, the event type, and identification of users and objects involved. Authorized administrators may configure auditing to audit or not audit specific events based on user identity, object type, and event type.

i5/OS provides systems-wide, user profile, and object based auditing levels to control the collected audit data,. These settings allow an authorized administrator to enable and disable auditing, as well as to select actions to take when an audit record can't be deposited in to the security audit log for any reason. In particular, an authorized administrator can configure the system to shut down when the system is unable to deposit the audit record into the journal receiver.

3.4 Security Management

i5/OS offers an extensive set of tools to manage and otherwise use its security services. i5/OS supports the notion of roles by assigning various special authorities to specific users. Access to essentially all of the i5/OS objects, including those used to store and manipulate the i5/OS security configuration, are protected using these authorities in conjunction with a discretionary access control policy.

i5/OS allows users to be assigned to roles based on their user class; the user class controls the options that are available to the user. Predefined users classes include the *Security Officer*, who performs all security functions including creating security administrators; the *Security Administrator,* who performs all security functions including creating security administrators; the *System Programmer*, who performs system programming functions; the *System Operator*, who also performs system maintenance and operation functions and can back up the system and save and restore objects; and the *End User*, who performs application functions. Users may also be given special authorities to augment their roles.

3.5 Protection of the TOE Security Functions

i5/OS protects itself using a combination of hardware support and strict control over the set of available applications. i5/OS includes a translator and compiler that are specifically designed to ensure that a given program will only access resources it is supposed to (e.g., the application will not be allowed to access memory from another user or system process). i5/OS maintains a domain for its own execution, and separates this domain from the user domain, by a combination of the state and domain attributes implemented in software. i5/OS runs in system or inherit state. All user code runs in user state. Most i5/OS objects are created in system domain storage. Therefore, code running in user state cannot access system domain objects directly; instead, they must use the defined i5/OS interfaces.

Manipulation of the state and domain attributes requires use of blocked i5/OS instructions. Code written on the evaluated configuration cannot use the blocked i5/OS instructions because the translator in the evaluated configuration does not translate blocked i5/OS instructions. Code written on the evaluated configuration cannot issue hardware instructions directly since the availability of compilers and translators is carefully controlled. The administrator guidance provides procedures for the system administrator to guard against object code being restored (or otherwise introduced) to the system without retranslation, which will ensure the integrity of the domains.

i5/OS blocks some instructions, and these are analogous to machine instructions that can only be executed while the machine is in supervisor mode. MI instructions can be blocked at translate time or at runtime. Instructions that are blocked at translate time are those that the translator will not translate.

i5/OS administrator documentation warns that introducing a translator that is called by the MI instructions Create Program (CRTPG) and Create Module (CRTMOD) other than the evaluated translator removes the system from the evaluated configuration. Regardless, such a translator cannot create an encapsulated program object because MI programs can write data only into spaces and spaces cannot have the program MI object type. Further, the i5/OS restore function prevents programs and other objects from being restored that are not allowed on the evaluated configuration.

i5/OS creates objects using a hardware storage protection attribute. During execution of each RISC instruction, the hardware determines whether the page frame is hardware storage protected. In this way, user state programs can have hardware storage protection read-only access to objects such as the entry point table.

i5/OS uses hardware tag bits set to identify a valid pointer data object. Obtaining a System Pointer (SYP) gives a process addressability to an MI object. However, that pointer will only be valid as input to MI instructions that will operate on an object of the type addressed by the pointer. Any attempt to modify any type of tagged pointer, except with an MI instruction designed to modify a pointer, causes the tag bit to be cleared. The storage will no longer be viewed as a pointer by any MI instruction. This, along with translator control of addressability to space pointer machine objects, prevents non-pointer data from being used as pointer data.

Diagnostic tests exist to ensure that the hardware is functioning correctly. Some of the tests execute automatically during i5/OS initial program load (IPL); additional tests can be exercised by an authorized administrator when necessary.

4 Assumptions

The assumptions underlying the evaluation of i5/OS are all based upon those present in the Controlled Access Protection Profile

4.1 Usage Assumptions

Authorized users are assumed to possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.

It is assumed that there will be one or more competent individuals assigned to manage the TOE and the security of the information it contains. These administrators are assumed not to be careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrative documentation.

4.2 Environmental Assumptions

The processing resources of the TOE are assumed to be located within controlled access facilities that will prevent unauthorized physical access. All connections to peripheral devices are assumed reside within those boundaries. CAPP-conformant TOEs only address security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths to access points such as terminals are assumed to be adequately protected.

Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints. CAPP-conformant TOEs are applicable to networked or distributed environments only if the entire network operates under the same constraints and resides within a single management domain. There are no security requirements that address the need to trust external systems or the communications links to such systems. There is also no assumption that networks into which the TOE is connected consist of homogeneous systems, although there is an assumption that they have common management and common policies.

Lastly, it is assumed that the TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

4.3 Overarching Policies

The security requirements enforced by the TOE were designed based on the following overarching security policies:

- **Accountability**. The users of the system shall be held accountable for their actions within the system.
- **Authorization**. Only those users who have been authorized to access the information within the system may access the system.
- **Need to Know**. Only those authorized users that have a 'need to know' for information will be provided access to the protected resources.

5 Architectural Information

Note: The following architectural description is based on the description presented in Part I of the i5/OS ETR and in the Security Target.

The TOE is physically composed of the IBM eServer models 520, 550, and 570 machines, upon which the i5/OS software operates, and a console and keyboard, physically connected to the iSeries machine. The eServer models include memory, disk drives, integrated network and disk controllers, tape drive, and CD-ROM drive. There are no additional external peripherals besides the console and keyboard. There is an IBM coprocessor physically attached to the eServer machine used to support initialization, but this coprocessor is not covered by the evaluation and is considered part of the IT Environment. Client workstations are connected to the iSeries machine, but are also considered part of the IT Environment.

The physical boundaries of the TOE occur at 1) the eServer machine, 2) the console and keyboard, 3) the interface between the iSeries machine and the IBM coprocessor; 4) the interface between the iSeries machine and client workstations. Figure 5-1 is provided to illustrate the separation between the TOE and its IT Environment. The TOE components are displayed in the dark grey shaded boxes, while the IT environment components are displayed in the light grey shaded boxes.

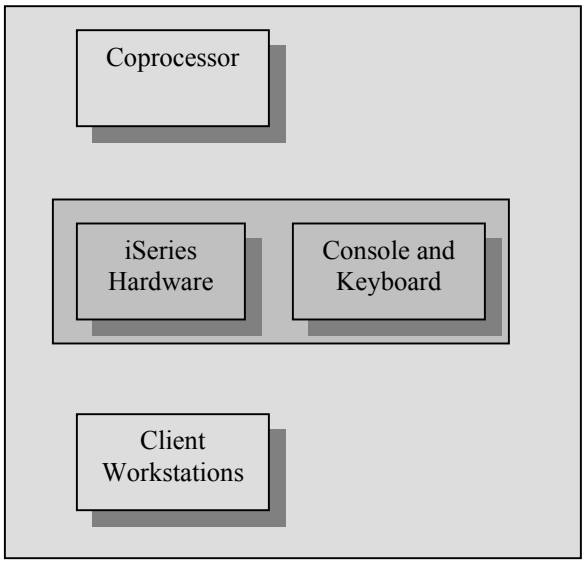

Figure 5-1. Physical TOE Boundaries

Physically, the iSeries hardware is connected to workstations that provide a primary interactive user interface, and to a network that offers network-oriented user services. Once connected to one of these interfaces, i5/OS software offers command line commands (CLs), application interfaces (APIs), and machine interface instructions (MIs) via available prompts, menus, and programs. For selected network services, well-defined protocols also serve as interfaces.

Figure 5-2 i5/OS Overview shows the layered architecture of i5/OS. As with most other operating systems, i5/OS consists of layers ranging from the most critical (hardware) to non-critical (user applications). The hardware is an IBM iSeries product and the lower layers (SLIC, MI) of i5/OS are designed to abstract hardware details away from the higher layers of i5/OS. As a result, the lower level interfaces are essentially static regardless of the underlying hardware; and it is these interfaces upon which i5/OS and user applications operate.

The I5/OS software architecture supports system components at four software layers divided by three interface layers. The i5/OS software and firmware is divided into components. The following paragraphs discuss the layering shown in the figure starting from the top of the figure.

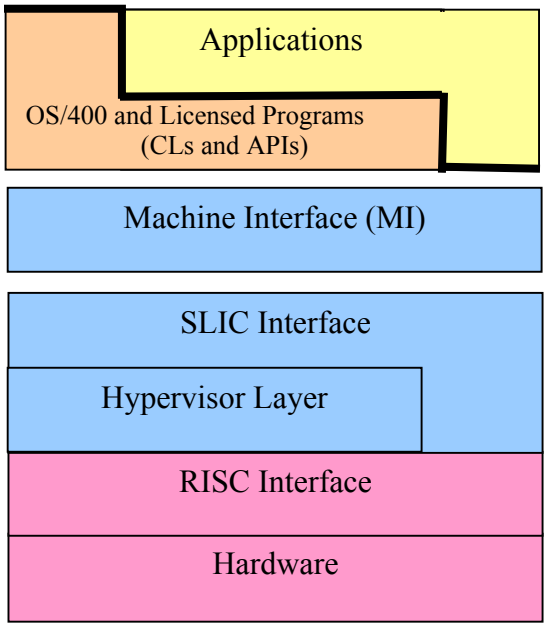

Figure 5-2 i5/OS Overview

The top layer is the actual operating system (still called Operating System/400 (OS/400) in the architectural diagrams) and the languages, utilities, and applications it supports. IBM separates licensed products (LPs) from i5/OS; this document uses i5/OS to mean both the operating system and the separately purchased LPs. From the user's point of view, i5/OS provides two interfaces for users: menus that allow selections, and direct entry of commands with parameter strings. The menus are based on programs or commands, while the direct entry of commands provides a means of interactively requesting services from i5/OS. In addition, a set of Application Programmer Interfaces (APIs) provide a means for programs to request services from i5/OS. Lastly, the system provides network commands that communicate via industry-standard network protocols.

To the end user, a unit of work on the i5/OS is known as a job. At the execution phase of a job, I5/OS causes a process to be initiated and the job becomes an active job. The active job-to-process relationship is one-to-one.

Unlike other systems in which the operating system interfaces directly with the hardware, i5/OS has a virtual machine interface (MI) under its operating system layer. When more conventional operating systems issue an instruction for execution by the hardware, i5/OS issues an MI instruction. This design makes MI look like the hardware interface to users, and also makes it possible to replace everything under the MI without affecting the user interfaces, user applications, or i5/OS itself.

The MI provides primitive instructions, two program models (original and new), and late binding capability. The MI also provides a consistent interface to low-level services. It is a logical, not a physical, interface, and is not executable. MI instructions are translated to Reduced Instruction Set Computer (RISC) executable instructions.

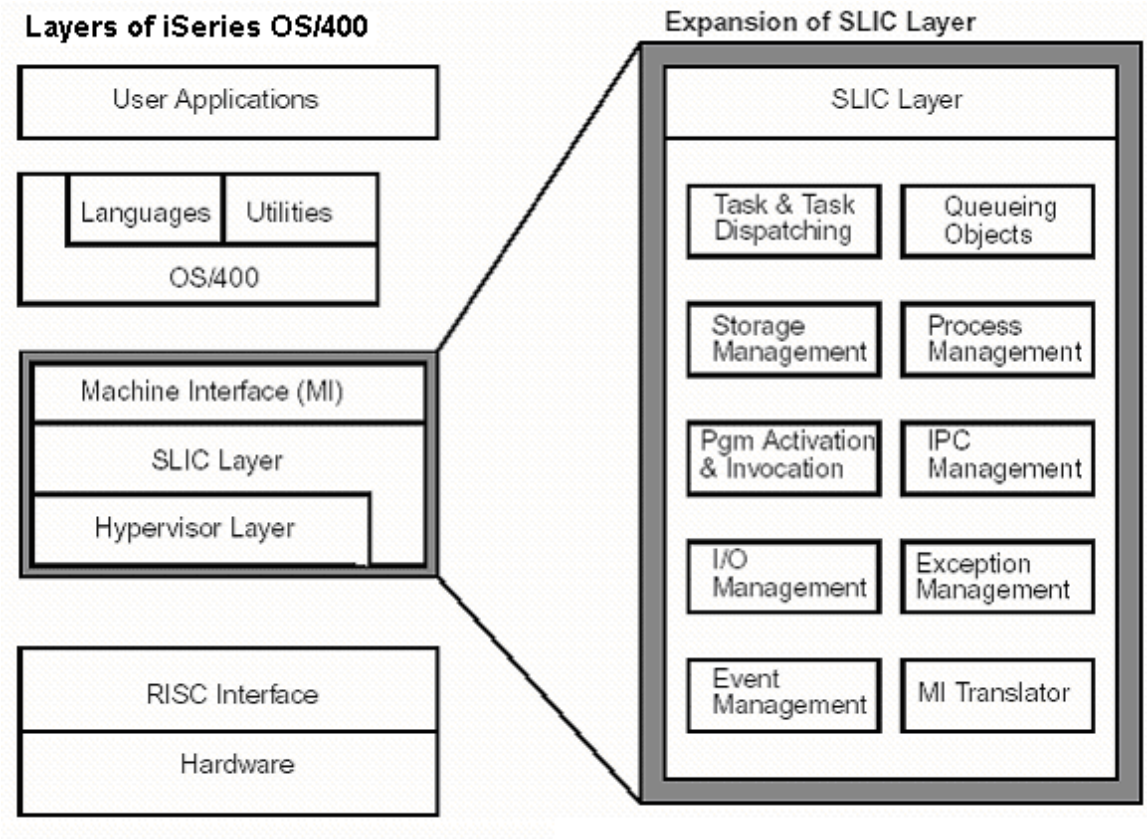

Figure 5-3 The SLIC Layer

Figure 5-3 The SLIC Layer shows the layers below the MI. These are the System Licensed Internal Code (SLIC) layer, Hypervisor layer, and the hardware. The RISC instruction set is the interface between the SLIC and hardware layers and between the

Hypervisor layer and the hardware layers. MI instructions are converted by the translator into RISC instructions that invoke the proper SLIC component.

The SLIC layer is made up of components that operate in defined ways upon a defined set of objects. These components include storage management, process management, task management, I/O management, exception management, interprocess communication, and database management support. MI objects and the MI instructions that operate upon those objects are owned by specific SLIC components. These SLIC components take the MI instructions they are designed to handle from the OS/400 layer and execute them in RISC instructions. The SLIC layer also provides a translator to translate the MI instructions in user-written programs into RISC code.

A process is known to the SLIC as a task. A task is a dispatchable unit of work. The SLIC layer sees both MI processes and SLIC tasks. Some SLIC tasks do not have a process or job associated with them; an example is an I/O task.

The Hypervisor layer is a layer of software that permits the resources of a single system to be partitioned into a set of logical partitions. Each partition runs a separate copy of SLIC and I5/OS, has its own memory, its own I/O resources, and its own load source. Hardware processors can be assigned to a single partition or shared between partitions. The Hypervisor provides functions for configuring the resources of a single system into multiple partitions and managing the communications between partitions.

The TOE can only be configured as a system with a single partition. In such a configuration, the SLIC layer and Hypervisor layer are considered as a single layer that performs the equivalent functions as the SLIC layer performed prior to the introduction of the Hypervisor.

The hardware supports a RISC instruction set. At execution time, the RISC instructions are interpreted and executed by the hardware.

Physically, the iSeries hardware is connected to workstations that provide a primary interactive user interface, and a network, that offers network-oriented user services. Once connected to one of these interfaces, i5/OS software offers command line commands (CLs), application interfaces (APIs), and machine interface instructions (MIs) via available prompts, menus, and programs.

6 Documentation

The following documentation was used as evidence for the evaluation of the i5/OS V5R3:[3]

6.1 Design documentation

Document	Version	Date
IBM Corporation OS/400 OS400 Audit Methodology	1.1	6 February 2005

[3] This documentation list is based on the list provided in the Evaluation Technical Report, Part 1, developed by SAIC.

IBM Corporation OS/400 Design Documentation	0.41	24 February 2005
IBM Corporation iSeries Operating System/400 Commands (pdf files)	V5R3	(none provided)
IBM Corporation V5R3 MI Instruction Documentation	V5R3	(none provided)
IBM Corporation OS/400 Interfaces	0.1	17 March 2005
proprietary PowerPC AS Documentation	2.01	September 2003
IBM OS/400 V5R3 Security Policy Model	0.3	29 April 2005
All Unblocked MIs Mapped SFRs via exceptions spreadsheet	0.3	(none provided)
CMD and API Mapping	-	16 November 2004
MI Mapping Notes	0.1	23 November 2004
OS400 API List spreadsheet	2.0	16 November 2004
• IBM iSeries Security Code Samples	1.0	1 July 2004
iSeries Configure Your System For Common Criteria Security, Document Number SC41-5336-00	V5R3	13 May 2005
iSeries Security Reference, Document Number SC41-5336-00	V5	(none provided)
RFC854, Telnet Protocol Specification		May 1983
RFC959, File Transfer Protocol (FTP)		October 1985
RFC1123, Requirements for Internet Hosts - Application and Support		October 1989
RFC1579, Firewall-Friendly FTP		February 1994
RFC1635, How to Use Anonymous FTP		May 1994
RFC2228, FTP Security Extensions		October 1997
RFC2389, Feature negotiation mechanism for the File Transfer Protocol		August 1998
RFC2577, FTP Security Considerations		May 1999
Rexec Design Specification	V1	29 July 2005

6.2 Guidance documentation

Document	Version	Date
iSeries Configure Your System For Common Criteria Security, Document Number SC41-5336-00	V5R3	13 May 2005
IBM iSeries Operating System/400 Commands	V5R3	May 2004
IBM iSeries Security Reference, Document Number SC41-5302-07	V5	(none provided)

The iSeries Information Center (http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html; in particular, the following documents all available under http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/info/: • rbapk/rbapkrbapk631usersviewsecurity.htm#rbapk631 • apis/api.htm • rbam6/rbam6clmain.htm • rzaiu/rzaiuicbackup.htm • db2/rbafzmst02.htm • rzai2/rzai2kickoff.htm • rzahgictcp2.htm • rbapk/rbapkrbapk003planninguser.htm • rbapk/rbapkrbapkaaacomface.htm • rbapk/rbapkprvaut.htm#prvaut • rbapk/rbapkrbapk103physicalsec.htm • rbapk/rbapkrbapk004planningresourc • rbapk/rbapkrbapkc17.htm	-	(none provided)

6.3 Configuration Management and Lifecycle documentation

Document	Version	Date
IBM iSeries OS/400 Configuration Management Plan	2.0	11 May 2005
IBM iSeries OS/400 System Life Cycle Document	2.0	16 April 2004
IBM Proprietary Compiler Documentation	(various)	(various)
IBM Proprietary Change Control Documentation	(various)	(various)

6.4 Delivery and Operation documentation

Document	Version	Date
IBM iSeries OS/400 V5R3 Common Criteria System Delivery Procedures	1.0	15 January 2004
iSeries Configure Your System For Common Criteria Security, Document Number SC41-5336-00	V5R3	13 May 2005

6.5 Test documentation

Document	Version	Date
IBM iSeries OS/400 Common Criteria Test Plan	1.1	22 December 2004
TestcaseInfo.xls (test coverage)	-	(none provided)
OSMInstructionTests.xls (test coverage)	-	(none provided)
Test Cases as referenced by TestcaseInfo.xls	-	(none provided)
Test Results as referenced by test cases	-	(none provided)

6.6 Vulnerability Assessment documentation

Document	Version	Date
IBM OS/400 V5R3 Vulnerability Analysis	0.2	29 April 2005
IBM OS/400 V5R3 Misuse Analysis	0.1	18 October 2004
iSeries Configure Your System For Common Criteria Security, Document Number SC41-5336-00	V5R3	13 May 2005
IBM iSeries Security Reference, Document Number SC41-5302-07	V5	(none provided)

6.7 Security Target

Document	Version	Date
IBM i5/OS Security Target	V1.0	8 July 2005

7 IT Product Testing

This section describes the testing efforts of the developer and the Evaluation Team. It is derived from information contained in the Evaluation Team Test Plan for the IBM i5/OS V5R3 Product [12], and has been reviewed to ensure it does not contain vendor proprietary information.

7.1 Developer Testing

Evaluator analysis of the developer's test plans, test scripts, and test results indicated that the developer's testing is adequate to satisfy the requirements of EAL4, augmented with AVA_VLA.2.

The developer's tests were automated and interface-based. There were test cases for each interface that addressed the following security areas:

1. Discretionary access control tests
2. Parameter validation tests
3. Audit tests
4. Special functional verification tests, such as authority manipulation, and TSF domain protection,

The evaluation team verified that each test area addressed both breadth and depth of coverage. Breadth was addressed by mapping all of the TSFI security checks and effects to a test. Test depth was addressed by the descriptions of the test families. These descriptions explain algorithms, combinations, and sequence that are applied to each of the specific test variations that are identified by interfaces and associated properties (e.g., parameters). Together, these test areas were designed to provide coverage of the security functions.

The test documentation included a high-level test plan detailing the philosophy and a description of the test areas. The test plan also provided descriptions about the test tools

and procedures for running the test cases. The actual details about the individual tests cases were found in the test source code.

For each of the developer tests, the evaluators analyzed the test procedures to determine whether the procedures were relevant to, and sufficient for the function being tested. They also verified that the test documentation showed results that were consistent with the expected results for each test script.

7.2 Evaluation Team Independent Testing

In addition to developer testing, the CCTL conducted its own suite of tests. Tests were conducted on all three platforms included in the evaluation - 520, 550, and 570, with Version 5, Release 3 (V5R3) of the i5/OS with Feature Code 1930.

The CCTL installed i5/OS in accordance with the guidance provided in the iSeries Configure Your System For Common Criteria Security (Version 5 Release 3) document. They then configured the system in accordance with the guidance provided in the iSeries Configure Your System For Common Criteria Security (Version 5 Release 3) document. Testing was conducted in late April 2005.

During its testing, the evaluation team ran a portion of the vendor test suite. The team selected its vendor test sample to include all vendor tests that were added or updated as a result of their ATE analysis, as well as an additional 20% of tests selected at random. The evaluation team then verified that the randomly selected tests included tests from all of the test areas. The team verified that all the selected tests passed, or a justification was provided as to why that test was not required to pass in the evaluated configuration.

The evaluation team also developed nineteen (19) independent tests. These tests focused on the behaviour of the various security functional policies. These tests identified no failures of the functions in the TOE. Testing was witnessed by a representative of the validation team.

7.3 Evaluation Team Penetration Testing

The CCTL also conducted penetration testing. Tests were conducted on all three platforms included in the evaluation - 520, 550, and 570, with Version 5, Release 3 (V5R3) of the i5/OS with Feature Code 1930.

The CCTL installed i5/OS in accordance with the guidance provided in the iSeries Configure Your System For Common Criteria Security (Version 5 Release 3) document. They then configured the system in accordance with the guidance provided in the iSeries Configure Your System For Common Criteria Security (Version 5 Release 3) document. Testing was conducted in April 2005 and August 2005.

Prior to developing its tests, the CCTL followed well-established penetration test development procedures. This resulted in a set of twelve (12) penetration test procedures. These tests identified no failures of the functions in the TOE. Testing was witnessed by a representative of the validation team.

The validation team also developed penetration tests to specifically address the protocol level interfaces. These tests identified no failures of the protocols against the claimed SFRs.

8 Evaluated Configuration

The evaluated configuration, as defined in the Security Target, is IBM i5/OS V5R3M0 running on IBM eServer models 520, 550, and 570 with Software Feature Code 1930, including memory, disk drives, integrated network and disk controllers, tape drive, and CD-ROM drive.

To use the product in the evaluated configuration, the product must be configured as specified in the *iSeries Configure Your System For Common Criteria Security (Version 5 Release 3)* document. This document notes that the i5/OS options included in the TOE are as follows:

Description	Libraries
General Purpose Library	QGPL
User Library	QUSRSYS
Extended Base Support	QQALIB, QSYS2
Online Information	QHLPSYS
Extended Base Directory Support	QSYSDIR, QSYSCGI
Example Tools Library	QUSRTOOL
AFP™ Compatibility Fonts	QFNTCPL
*PRV CL Compiler Support	QSYSVxRyMz
I5/OS GDDM®	QGDDM
System Openness Includes	QSYSINC
Extended NLS Support	QSYSLOCALE
Cryptographic Service Provider[4]	QCCA

The ST and ETR note that not all available licensed programs were included in the TOE. The following licensed programs are covered by this evaluation:[8]

Product Number	Option	Description	Version
5722SS1		OS/400 Library QGPL	V5R3M0
		OS/400 Library QUSRSYS	V5R3M0
	1	OS/400 Extended Base Support	V5R3M0
	2	OS/400 Online Information	V5R3M0
	3	OS/400 Extended Base Directory Support	V5R3M0
	7	OS/400 Example Tools Library	V5R3M0
	8	OS/400 AFP™ Compatibility Fonts	V5R3M0
	9	OS/400 *PRV CL Compiler Support	V5R3M0

[4] Note that cryptographic algorithms are not covered by the evaluation.

	12	OS/400 Host Servers	V5R3M0
	13	OS/400 System Openness Includes	V5R3M0
	14	OS/400 GDDM	V5R3M0
	21	OS/400 Extended NLS Support	V5R3M0
	26	OS/400 DB2® Symmetric Multiprocessing	V5R3M0
	27	OS/400 DB2 Multisystem	V5R3M0
	35	OS/400 CCA Cryptographic Service Provider[5]	V5R3M0
	36	OS/400 PSF/400 1-45 IPM Printer Support	V5R3M0
	37	OS/400 PSF/400 1-100 IPM Printer Support	V5R3M0
	38	OS/400 PSF/400 Any Speed Printer Support	V5R3M0
	39	OS/400 International Components for Unicode	V5R3M0
	43	OS/400 Additional Fonts	V5R3M0
5722AC3	*BASE	IBM Cryptographic Acess Provider 128-bit for iSeries[5]	V5R3M0
5769FNT	*BASE	IBM Advanced Function Printing™ Fonts for AS/400® with all available options	V4R2M0
	1	AFP Fonts – Sonoran Serif	V4R2M0
	2	AFP Fonts – Sonoran Serif Headliner	V4R2M0
	3	AFP Fonts – Sonoran Sans Serif	V4R2M0
	4	AFP Fonts – Sonoran Sans Serif Headliner	V4R2M0
	5	AFP Fonts – Sonoran Sans Serif Condensed	V4R2M0
	6	AFP Fonts – Sonoran Sans Serif Expanded	V4R2M0
	7	AFP Fonts – Monotype Garamond	V4R2M0
	8	AFP Fonts – Century Schoolbook	V4R2M0
	9	AFP Fonts – Pi and Specials	V4R2M0
	10	AFP Fonts – ITC Souvenir	V4R2M0
	11	AFP Fonts – ITC Avant Garde Gothic	V4R2M0
	12	AFP Fonts – Math and Science	V4R2M0
	13	AFP Fonts – DATA1	V4R2M0
	14	AFP Fonts – APL2®	V4R2M0
	15	AFP Fonts – OCR A and OCR B	V4R2M0
5769FN1	*BASE	Advance Function Printing DBCS Fonts/400 with all available options	V4R2M0
	1	AFP DBCS Fonts – Japanese	V4R2M0
	2	AFP DBCS Fonts – Korean	V4R2M0
	3	AFP DBCS Fonts – Traditional Chinese	V4R2M0
	4	AFP DBCS Fonts – Simplified Chinese	V4R2M0
	5	AFP DBCS Fonts – Thai	V4R2M0
5722QU1	*BASE	Query	V5R3M0
5722ST1	*BASE	IBM DB2 Query Manager and SQL Development Kit for iSeries	V5R3M0
5722TC1	*BASE	IBM TCP/IP Utilities	V5R3M0
5722XE1	*BASE	IBM eServer™ iSeries Access for Windows®	V5R3M0
5722XW1	*BASE	IBM eServer iSeries Family	V5R3M0

[5] Note that cryptographic algorithms are not covered by this evaluation.

	1	ISeries Access Enablement Support	V5R3M0

9 Results of the Evaluation

The evaluation was conducted based upon the Common Criteria (CC), Version 2.1, dated August 1999 [1,2,3,4]; the Common Evaluation Methodology (CEM), Version 1.0, dated August 1999 [6]; and all applicable International Interpretations in effect on 19 November 2003. The evaluation confirmed that the IBM i5/OS V5R3M0 product running on IBM eServer models 520, 550, and 570 with Software Feature Code 1930 is compliant with the Common Criteria Version 2.1, functional requirements (Part 2), Part 2 extensions, and assurance requirements (Part 3) for EAL4 augmented with AVA_VLA.2. The details of the evaluation are recorded in the CCTL's evaluation technical report, Evaluation Technical Report for the i5/OS V5R3, Part 1 (Non-Proprietary) and Part 2 (Proprietary). The product was evaluated and tested against the claims presented in the i5/OS V5R3 Security Target v1.0, 8 July 2005.

The validator followed the procedures outlined in the Common Criteria Evaluation Scheme publication number 3 for Technical Oversight and Validation Procedures. The validator has observed that the evaluation and all of its activities were in accordance with the Common Criteria, the Common Evaluation Methodology, and the CCEVS. The validator therefore concludes that the evaluation team's results are correct and complete.

The following evaluation results are extracted from the non-proprietary Evaluation Technical Report provided by the CCTL, and are augmented with the validator's observations thereof.

9.1 Evaluation of the Security Target (ASE)

The evaluation team applied each ASE CEM work unit. The ST evaluation ensured the ST contains a description of the environment in terms of policies and assumptions, a statement of security requirements claimed to be met by the IBM i5/OS product that are consistent with the Common Criteria, and product security function descriptions that support the requirements.

The validator reviewed the work of the evaluation team, and found that sufficient evidence and justification was provided by the evaluation team to confirm that the evaluation was conducted in accordance with the requirements of the CEM, and that the conclusion reached by the evaluation team was justified.

9.2 Evaluation of the Configuration Management Capabilities (ACM)

The evaluation team applied each EAL 4 ACM CEM work unit. The ACM evaluation ensured the TOE is identified such that the consumer is able to identify the evaluated TOE. The evaluation team ensured the adequacy of the procedures used by the developer to accept, control and track changes made to the TOE implementation, design documentation, test documentation, user and administrator guidance, security flaws and the CM documentation. The evaluation team ensured the procedure included automated support to

control and track changes to the implementation representation. The procedures reduce the risk that security flaws exist in the TOE implementation or TOE documentation. To support the ACM evaluation, the evaluation team received Configuration Management (CM) records from IBM and performed a CM audit.

The validator reviewed the work of the evaluation team, and found that sufficient evidence and justification was provided by the evaluation team to confirm that the evaluation was conducted in accordance with the requirements of the CEM, and that the conclusion reached by the evaluation team was justified.

9.3 Evaluation of the Delivery and Operation Documents (ADO)

The evaluation team applied each EAL 4 ADO CEM work unit. The ADO evaluation ensured the adequacy of the procedures to deliver, install, and configure the TOE securely. The evaluation team ensured the procedures addressed the detection of modification, the discrepancy between the developer master copy and the version received, and the detection of attempts to masquerade as the developer. The evaluation team followed the Configuration Guide to test the installation procedures to ensure the procedures result in the evaluated configuration.

The validator reviewed the work of the evaluation team, and found that sufficient evidence and justification was provided by the evaluation team to confirm that the evaluation was conducted in accordance with the requirements of the CEM, and that the conclusion reached by the evaluation team was justified.

9.4 Evaluation of the Development (ADV)

The evaluation team applied each EAL 4 ADV CEM work unit. The evaluation team assessed the design documentation and found it adequate to aid in understanding how the TSF provides the security functions. The design documentation consists of a functional specification, a high-level design document, a low-level design document, and a security policy model. The evaluation team also ensured that the correspondence analysis between the design abstractions correctly demonstrated that the lower abstraction was a correct and complete representation of the higher abstraction.

Additionally, the evaluation team ensured that the security policy model document clearly describes the security policy rules that were found to be consistent with the design documentation.

The validator reviewed the work of the evaluation team, and found that sufficient evidence and justification was provided by the evaluation team to confirm that the evaluation was conducted in accordance with the requirements of the CEM, and that the conclusion reached by the evaluation team was justified.

9.5 Evaluation of the Guidance Documents (AGD)

The evaluation team applied each EAL 4 AGD CEM work unit. The evaluation team ensured the adequacy of the user guidance in describing how to use the operational TOE. Additionally, the evaluation team ensured the adequacy of the administrator guidance in

describing how to securely administer the TOE. Both of these guides were assessed during the design and testing phases of the evaluation to ensure they were complete.

The validator reviewed the work of the evaluation team, and found that sufficient evidence and justification was provided by the evaluation team to confirm that the evaluation was conducted in accordance with the requirements of the CEM, and that the conclusion reached by the evaluation team was justified.

9.6 Evaluation of the Life Cycle Support Activities (ALC)

The evaluation team applied each EAL 4 ALC CEM work unit. The evaluation team ensured the adequacy of the developer procedures to protect the TOE and the TOE documentation during TOE development and maintenance to reduce the risk of the introduction of TOE exploitable vulnerabilities during TOE development and maintenance. The evaluation team ensured the procedures described the life-cycle model and tools used to develop and maintain the TOE. To support the ALC evaluation, the evaluation team performed an audit of the security measures at IBM.

In addition to the EAL 4 ALC CEM work units, the evaluation team applied the ALC_FLR.2 work units from the CEM supplement. The flaw remediation procedures were evaluated to ensure that flaw reporting procedures exist for managing flaws discovered in the TOE.

The validator reviewed the work of the evaluation team, and found that sufficient evidence and justification was provided by the evaluation team to confirm that the evaluation was conducted in accordance with the requirements of the CEM, and that the conclusion reached by the evaluation team was justified.

9.7 Evaluation of the Test Documentation and the Test Activity (ATE)

The evaluation team applied each EAL 4 ATE CEM work unit. The evaluation team ensured that the TOE performed as described in the design documentation and demonstrated that the TOE enforces the TOE security functional requirements. Specifically, the evaluation team ensured that the vendor test documentation sufficiently addresses the security functions as described in the functional specification and high level design specification. The evaluation team performed a sample of the vendor test suite, and devised an independent set of team test and penetration tests. The vendor tests, team tests, and penetration tests substantiated the security functional requirements in the ST.

The validator reviewed the work of the evaluation team, and found that sufficient evidence and justification was provided by the evaluation team to confirm that the evaluation was conducted in accordance with the requirements of the CEM, and that the conclusion reached by the evaluation team was justified.

9.8 Vulnerability Assessment Activity (AVA)

The evaluation team applied each EAL 4 AVA CEM work unit. The evaluation team ensured that the TOE does not contain exploitable flaws or weaknesses in the TOE based upon the developer strength of function analysis, the developer vulnerability analysis, the

developer misuse analysis, and the evaluation team's misuse analysis and vulnerability analysis, and the evaluation team's performance of penetration tests.

The validator reviewed the work of the evaluation team, and found that sufficient evidence and justification was provided by the evaluation team to confirm that the evaluation was conducted in accordance with the requirements of the CEM, and that the conclusion reached by the evaluation team was justified.

9.9 Summary of Evaluation Results

The evaluation team's assessment of the evaluation evidence demonstrates that the claims in the ST are met. Additionally, the evaluation team's performance of a subset of the vendor tests suite, the independent tests, and the penetration test also demonstrated the accuracy of the claims in the ST.

The validation team's assessment of the evidence provided by the evaluation team is that it demonstrates that the evaluation team followed the procedures defined in the CEM, and correctly verified that the product meets the claims in the ST.

10 Validator Comments/Recommendations

- Users of this TOE are cautioned as to the importance of the assumptions, and that this product was not designed for use in a potentially malicious environment, such as the Internet.
- Users of this TOE are reminded that prevention of denial of service is not an objective of this TOE. As such, testing of the TOE (in particular testing of the network protocols) did not investigate whether it was possible to transition the TOE into a denial of service state.
- The evaluation team identified one command, RUNRMTCMD, that failed to obscure the plaintext of the password. This is permitted by the CAPP through analogy to batch card input. Users of this TOE, however, are cautioned to be aware of this behaviour, and to ensure that any batch files containing invocations of RUNRMTCMD need to be adequately protected, as they contain plaintext passwords. Users, as always, should also be cautious when entering passwords to ensure they cannot be observed.
- The validators note that the MI translator plays a critical role in enforcement of domain protection in this product. Users of this TOE must use only MI translators approved for use in the evaluated configuration.
- The validators note that peripherals may serve as a vector for information transfer through device status registers. Although no such vector was uncovered with the peripherals currently included with the models covered by this evaluation, future peripherals may provide such avenues. Users of this TOE are directed to exercise caution when selecting peripherals for use with the product.

11 Annexes

Not applicable.

12 Security Target

The Security Target is identified as *IBM i5/OS V5R3 Security Target,* Version 1.0, 8 July 2005. The document identifies the security functional and assurance requirements (SFRs) necessary to conform with the Controlled Access Protection Profile (CAPP) v1.d at EAL 4 augmented with ALC_FLR.2.

13 Glossary

The following definitions are used throughout this document:

- **Access Control List (ACL)**. A list associated with an object that identifies all the subjects that can access the object and their access rights. For example, an access control list is a list that is associated with a file that identifies the users who can access the file and that identified the users' access rights to that file.
- **Adopted Authority**. Authority given to the user by the object while the object is running. The object must be created with owner authority. These object types can have adopted authority: program, service program, and SQL package.
- **All Authority**. An object authority that allows the user to perform all operations on the object except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, and change the object.
- **Attribute.** A characteristic or trait of an entity that describes the entity; for example, the telephone number of an employee is one of that employee's attributes. An attribute may have a type, which indicates the range of information given by the attribute, and a value, which is within that range.
- **Audit Journal**. A journal used by the system to keep a record of security-relevant events that occur.
- **Audit Level**. The types of user actions that are currently being audited for the entire system or for specific users on the system. Actions that can be audited include authority failures and restoring objects. A record of each action is written to the audit journal.
- **Audit Trail.** Data, in the form of a logical path that links a sequence of events, used for tracing the transactions that affected the contents of a record.
- **Authentication.** Verification of the identity of a user or the user's eligibility to access an object.
- **Authority**. The right to access objects, resources, or functions.

- **Authority Checking**. A function of the system that looks for and verifies a user's authority to an object.
- **Authorization List**. A list of two or more user IDs and their authorities for system resources.
- **Common Criteria Testing Laboratory (CCTL)**. An IT security evaluation facility accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the CCEVS Validation Body to conduct Common Criteria-based evaluations.
- **Conformance**. The ability to demonstrate in an unambiguous way that a given implementation is correct with respect to the formal model.
- **Data Authority.** A specific authority to read, add, update, or delete data, to run a program, or to search a library or directory.
- **Evaluation**. The assessment of an IT product against the Common Criteria using the Common Criteria Evaluation Methodology to determine whether or not the claims made are justified; or the assessment of a protection profile against the Common Criteria using the Common Evaluation Methodology to determine if the Profile is complete, consistent, technically sound and hence suitable for use as a statement of requirements for one or more TOEs that may be evaluated.
- **Evaluation Evidence**. Any tangible resource (information) required from the sponsor or developer by the evaluator to perform one or more evaluation activities.
- **Feature.** Part of a product that is either included with the product or can be ordered separately.
- **Hypervisor.** A specialized portion of Licensed Internal Code that enables logical partitioning.
- **Job**. A separately executable unit of work defined by a user, and run by a computer.
- **Journal.** A system object that identifies the objects being journaled, the current journal receiver, and all the journal receivers on the system for the journal.
- **Licensed Internal Code.** The layered architecture below the machine interface (MI) and above the machine. The Licensed Internal Code is a proprietary system design that carries out many functions. These functions include but are not limited to storage management, pointers and addressing, program management functions, exception and event management, data functions, I/O managers, and security.
- **Licensed Program (LP).** A separately orderable program, supplied by IBM, that performs functions related to processing user data. Examples of licensed programs are iSeries Access for Windows, ILE COBOL, and Backup Recovery and Media Services (BRMS).
- **Machine Interface (MI).** The interface, or boundary, between the operating system and the Licensed Internal Code.

- **Object** *(in the i5/OS sense)***.** A named storage space that consists of a set of characteristics that describe the space and, in some cases, data. An object is anything that occupies space in storage and on which operations can be performed. Some examples of objects are programs, files, libraries, and folders. Objects also have a set of operations associated with that data.
- **Object Authority.** A specific authority that controls what a system user can do with an entire object. For example, object authority includes deleting, moving, or renaming an object. There are five types of object authorities: object operational, object management, object existence, object alter, and object reference.
- **Owner Authority**. The authority that the object's owner has to the object.
- **Primary Group Authority.** The authority that the primary group has to the object.
- **Private Authority.** The authority specifically given to a user for an object that overrides any other authorities, such as the authority of a user's group profile or an authorization list.
- **Profile.** Data that describes the characteristics of a user, group, program, device, or remote location.
- **Program Temporary Fix (PTF)**. For zSeries(R), iSeries, and pSeries(TM) products, a fix that is made available to all customers. A program temporary fix is tested by IBM. It contains a PTF record.
- **Public Authority.** The authority given to users who do not have any specific (private) authority to an object, who are not on the authorization list (if one is specified for the object), and whose group profile has no specific authority to the object.
- **Receiver Chain.** The journal receivers presently or previously attached to the same journal. Each journal receiver, except the first one, has a previous receiver that was attached before the current receiver. Each journal receiver, except the currently attached receiver, has a next receiver.
- **Reduced Instruction Set Computer (RISC).** A computer that uses a small, simplified set of frequently used instructions for rapid processing.
- **Space**. Any storage area that can be directly accessed, down to its individual (8-bit) bytes, by a machine interface user such as a program or procedure.
- **Special Authority**. The types of authority a user can have to perform system functions, including all object authority, save system authority, job control authority, security administrator authority, spool control authority, service authority, and system configuration authority.
- **System Pointer.** A pointer that contains addressability to a machine interface system object.
- **System State Program.** A program that can access a user domain object or a system domain object. The system state is reserved for IBM-supplied programs.

- **Target of Evaluation (TOE)**. A group of IT products configured as an IT system, or an IT product, and associated documentation that is the subject of a security evaluation under the CC.
- **Translator.** An i5/OS component that performs the final step in a program or module compilation.
- **User Class.** The classification of a user by the system task, such as security officer, security administrator, programmer, system operator, and user. Each user class has a set of special authorities depending on the security level of the system. The user class determines which options are shown on the IBM-supplied menus.
- **Validation**. The process carried out by the CCEVS Validation Body leading to the issue of a Common Criteria certificate.
- **Validation Body**. A governmental organization responsible for carrying out validation and for overseeing the day-to-day operation of the NIAP Common Criteria Evaluation and Validation Scheme.

14 Bibliography

The Validation Team used the following documents to produce this Validation Report:

[1] Common Criteria Project Sponsoring Organisations. *Common Criteria for Information Technology Security Evaluation: Part 1: Introduction and General Model*, Version 2.1, August 1999. CCIMB-99-031.

[2] Common Criteria Project Sponsoring Organisations. *Common Criteria for Information Technology Security Evaluation: Part 2: Security Functional Requirements*, Version 2.1, August 1999. CCIMB-99-032.

[3] Common Criteria Project Sponsoring Organisations. *Common Criteria for Information Technology Security Evaluation: Part 3: Security Assurance Requirements*, Version 2.1, August 1999. CCIMB-99-033.

[4] Common Criteria Project Sponsoring Organisations. *Common Evaluation Methodology for Information Technology Security* – Part 1: Introduction and general model, Version 0.6, 11 January 1997.

[5] Common Criteria Project Sponsoring Organisations. *Common Evaluation Methodology for Information Technology Security* – Part 2: Evaluation Methodology, Version 1.0, August 1999.

[6] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, *Guidance to Validators of IT Security Evaluations*, Scheme Publication #3, Version 1.0, January 2002.

[7] IBM Corporation. *iSeries Information Center Glossary*. http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/index.htm.

[8] IBM Corporation. *Configure Your System For Common Criteria Security*. V5R3M0. October 2004.

[9] Science Applications International Corporation. *IBM i5/OS V5R3 Security Target,* Version 1.0, 8 July 2005.

[10] Science Applications International Corporation. *Evaluation Technical Report for the IBM i5/OS, Part 1 (Non-Proprietary)*, Version 4.0, August 8, 2005.

[11] Science Applications International Corporation. *Evaluation Technical Report for the IBM i5/OS Part 2 (Proprietary)*, Version 5.0, August 8, 2005.

Note: This document was used only to obtain the names of additional evidence consulted that was not listed in Part 1.

[12] Science Applications International Corporation. *Evaluation Team Test Plan for the IBM i5/OS V5R3 Product, ETR Part 2 Supplement (SAIC and IBM Proprietary)*, Version 8,0, August 9, 2005.

Note: This document was used only to develop summary information regarding the testing performed by the CCTL.