OS/390 & z/OS

SECURITY TECHNICAL IMPLEMENTATION GUIDE

Version 5, Release 1

Volume 1 of 2

21 January 2005

Developed by DISA for the DOD

This page is intentionally left blank.

TABLE OF CONTENTS

Page

LIST OF TABLES (A)

This page is intentionally left blank.

SUMMARY OF CHANGES (A)

Changes in this document since the previous release (Version 4, Release 1) in August 2003 are listed below.

General

Minor wording, grammar, formatting, and typographical changes and corrections are not included in the Summary of Changes.

The major changes to Volumes I and II involved the addition of bullets for the PDIs (Potential Discrepancy Items) identified in the VMS database. These changes were made to keep the OS/390 and z/OS STIG in line with the STIGs published for all other technologies. Additional changes included the creation of new PDIs for breaking apart Roman Numeral PDIs.

All PDIs greater than 8 characters in length were changed to 8 characters to allow use as directory entries in partitioned data sets for future automation.

Many changes were made in the area of removal of DISA Standards.

Preface

Made minor changes.

List of Tables (A)

Updated the List of Tables.

Section 1. Introduction

Made section changes based on recent STIG consistency efforts. Sections 1.11- 1.15. Moved or deleted the following sections in Section 1; Organizational Relationships, Security Administration, OS/390 and z/OS Security Design, System Authorization Facility, Security Controls, Development and Test Domains, Education and Awareness Programs, Extensions, STIG Distribution, Document Review Process, Document Revisions, and Related Documentation.

Section 1.1. Background

Updated Background to include information concerning the technology.

Section 1.2. Scope

Updated the Scope of the STIG.

Section 1.3. Writing Conventions

Corrected the Writing Conventions of the paper and included a note concerning the version “xx” reference.

Section 1.4. Authority

Updated this section to reflect correct Authority.

Section 1.5. STIG Distribution

Updated this section with current information.

Section 1.6. Document Revisions

Updated this section with the current information referring to the correct email address.

2.1.2 Software Integrity

Added CAT 1 PDI for vendor unsupported software.

Added PDI to have documented procedures for applying software patches.

2.5.2.6.2 Unprivileged Users and Groups

Added restriction for the use of OMVS Default UID with FTP socket applications.

2.5.3.2 Defining Users and Groups

Added for ACF2 that DFTGROUP and DFTUSER Options to be used only for FTP socket applications on non classified systems.

2.5.4.2 Defining Users and Groups

Added for RACF that BPX.DEFAULT.USER profile will only be used for FTP socket applications on non classified systems.

2.5.5.2 Defining Users and Groups

Added for TSS that OMVSUSR and OMVSGRP control options will only be used for FTP socket applications on non classified systems.

Section 3.1.5.3. Sensitive Utility Controls

Updated table.

- Included WHOIS command into the *Sensitive Utility Table, Section 3.1.5.3*

3.4.1 Standard Global Options (Control Options)

Included requirement that Top Secret FACILTY Control Option all have MODE=FAIL specified in STANDARD GLOBAL OPTIONS table in *Section 3.4.1*.

4.3 MQSERIES/WebSphere MQ

Removed section 4.5 WebSphere MQ and merged it with MQSERIES in section 4.3. Adjusted numbering for the rest of section 4.

This page is inetionslly left blank.

1. INTRODUCTION

1.1 Background

OS/390 Security Design for most mainframe information systems deployed throughout DOD use the International Business Machines (IBM) OS/390 or z/OS operating system. Controls within OS/390 and z/OS have been developed and documented in IBM references to ensure operating system integrity is maintained. This document is in the process of transitioning from OS/390 to z/OS. Any and all references to OS/390 will apply to both OS/390 and z/OS.

Security mechanisms that provide MAC II Sensitive level controls for the OS/390 and z/OS operating environments are implemented by the addition of Access Control Products (ACPs). The ACPs currently in use throughout DOD are listed below:

- Access Control Facility 2 (ACF2) - Computer Associates (CA)
- Resource Access Control Facility (RACF) - IBM Corporation[1]
- TOP SECRET (TSS) - Computer Associates (CA)

To maintain the integrity of the site, the ACP must be properly installed and configured. Options specified during the installation and techniques involved in the administration of these products can reduce the assurance introduced into the individual operating environment. As a result, guidance is needed on how these products should be configured in the operational environment.

The System Authorization Facility (SAF) provides an installation with centralized control over system security processing through a system service called the OS/390 router. The OS/390 router provides a focal point for all products that provide resource management. Access to the OS/390 router is via the **RACROUTE** macro, which invokes the router program itself. The router in turn invokes the ACP to determine if authorization exists for the resource being tested.

This concept provides a single interface that encourages the use of common functions across products and platforms. Products that interface via SAF calls can be protected with any of the three ACPs discussed in this document without modification of their interface code.

All new software acquired for or developed by DOD will fully utilize the SAF interface. Existing software that fails to utilize the SAF interface will be converted to do so where possible.

1.2 Authority

DOD Directive 8500.1 requires that “all IA and IA-enabled IT products incorporated into DOD information systems shall be configured in accordance with DOD-approved security configuration guidelines” and tasks DISA to “develop and provide security configuration guidance for IA and IA-enabled IT products in coordination with Director, NSA.” This document is provided under the authority of DOD Directive 8500.1.

[1] IBM has renamed RACF as the OS/390 Security Server. In the interest of brevity, clarity, and continuity this document continues to refer to the product as RACF.

The use of the principles and guidelines in this STIG will provide an environment that meets or exceeds the security requirements of DOD systems operating at the MAC II Sensitive level, containing unclassified but sensitive information.

1.3 Scope

The requirements set forth in this document will assist Information Assurance Managers (IAMs), Information Assurance Officers (IAOs), System Administrators (SAs), and systems programming personnel in support of protecting OS/390 and z/OS operating systems, ACPs, and IA-enabled products in the following sites:

- Systems Management Centers (SMCs)
- Computing Services Processing Element (CSPE)
- Systems Support Offices (SSOs)
- DOD Components
- Other DOD customers

1.4 Writing Conventions

Throughout this document, statements are written using words such as “**will**” and “**should**.” The following paragraphs are intended to clarify how these STIG statements are to be interpreted.

A reference that uses “**will**” implies mandatory compliance. All requirements of this kind will also be documented in the italicized policy statements in bullet format, which follow the topic paragraph. This will make all “**will**” statements easier to locate and interpret from the context of the topic. The IAO will adhere to the instruction as written. Only an extension issued by the Designated Approving Authority (DAA) will table this requirement. The extension will normally have an expiration date, and does not relieve the IAO from continuing their efforts to satisfy the requirement.

A reference to “**should**” is considered a recommendation that further enhances the security posture of the site. These recommended actions will be documented in the text paragraphs but not in the italicized policy bullets. Nevertheless, all reasonable attempts to meet this criterion will be made.

For each italicized policy bullet, the text will be preceded by parentheses containing the italicized Short Description Identifier (SDID), which corresponds to an item on the checklist and the severity code of the bulleted item. An example of this will be as follows "(*G111: CAT II*). "If the item presently has no Potential Discrepancy Item (PDI), or the PDI is being developed, it will contain a preliminary severity code and "N/A" for the SDID (i.e., "*[N/A: CAT III]*").

1.5 Vulnerability Severity Code Definitions

Category I	Vulnerabilities that allow an attacker immediate access into a machine, allow superuser access, or bypass a firewall.
Category II	Vulnerabilities that provide information that have a high potential of giving access to an intruder.
Category III	Vulnerabilities that provide information that potentially could lead to compromise.
Category IV	Vulnerabilities, when resolved, will prevent the possibility of degraded security.

1.6 STIG Distribution

Parties within the DOD and Federal Government's computing environments can obtain the applicable STIG from the Information Assurance Support Environment (IASE) web site. This site contains the latest copies of any STIG, as well as checklists, scripts, and other related security information. The NIPRNet URL for the IASE site is http://iase.disa.mil/. The National Institute of Standards and Technology (NIST) site is http://csrc.nist.gov/pcig/cig.html. Access to the STIGs on the IASE web server requires a network connection that originates from a **.mil** or **.gov** address. The STIGs are available to users that do not originate from a **.mil** or **.gov** address by contacting the FSO Support Desk at DSN 570-9264, commercial 717-267-9264, or e-mail to **fso_spt@disa.mil**.

1.7 Document Revisions

Comments or proposed revisions to this document should be sent via e-mail to **fso_spt@disa.mil**. DISA FSO will coordinate all change requests with the relevant DOD organizations before inclusion in this document.

This page is intentionally left blank.

2. OS/390 INTEGRITY

Integrity of the environment consists of securing the system-level processes and the data-level processes. The following sections discuss each of these in detail.

2.1 System-Level Integrity

System-level integrity consists of protecting hardware resources and software resources.

2.1.1 Hardware Integrity

Every operating environment is composed of hardware resources. These include facilities such as the central processing units (CPUs), direct access storage devices (DASD), tapes, consoles, hard-copy logs, printers, and communications devices.

When handled improperly, these components can create exposures within the operating environment that cannot be controlled with any software process. This could result in outages or access to data without proper authorization.

This document is not intended to address the resolution of the integrity of the hardware environment. Access controls are designed and implemented as part of the physical security plan for the site. The concept of I&A is the principal mechanism for controlling these resources.

2.1.2 Software Integrity

Operating Systems will be maintained at a supported level.

- *(AAMV0012: CAT I) The IAO will ensure that unsupported system software is removed or upgraded prior to a vendor dropping support.*

- *(AAMV0014: CAT II) The IAO will ensure that the site has a formal migration plan for removing or upgrading OS systems prior to the date the vendor drops security patch support.*

Maintaining the security of a Mainframe system requires frequent reviews of security bulletins. Many security bulletins mandate the installation of a software patch (hot fix) to overcome security vulnerabilities.

System Programmers and IAOs should regularly check OS vendor web sites for information on new security patches that are applicable to their site. All applicable security patches will be scheduled to be applied to the system. A security patch is deemed applicable if the product is installed, even if it is not used or is disabled.

FSO does not test or approve patches or service packs. It is the site’s responsibility to test vendor patches within their test environment.

- *(AAMV0016: CAT II) The IAO will subscribe to the DOD-CERT/VCTS (Vulnerability Compliance Tracking System) bulletin mailing list.*

- *(AAMV0018: CAT I) The IAO will ensure that all security related software patches are scheduled to be applied and documented.*

IBM has created a formal integrity statement that defines the integrity philosophy of the operating system. Controlling potential integrity exposures is the user's responsibility. As such, recommendations have been developed to limit the exposure potential for each of the operating system elements.

The IBM manuals z/OS MVS Programming: Authorized Assembler Services Guide (SA22-7608), Chapter 21, Protecting the System and MVS Planning: Security (GC28-1439), Chapter 5, MVS System Integrity, lists several areas of concern regarding the integrity of an MVS operating system.

A product usually does not harm a system's integrity if the product does the following:

- Uses only non-authorized and non-restricted OS/390 interfaces.
- Runs only as a problem program.
- Does not modify OS/390 in some way.

A product that performs any of the following actions, however, could introduce a system integrity exposure:

- Runs authorized or with special privileges so it can use OS/390 facilities restricted to authorized programs.

- Requires the use of a new Supervisor Call routine (SVC), Program Call routine (PC), installation exit routine, or I/O appendage routine.

- Modifies MVS in any way.

- Requires the use of the Authorized Program Facility (APF).

- Requires that the name of the program be placed in the MVS Program Properties Table (PPT).

- Runs in Supervisor State.

- Runs with a program status word (PSW) protection key between **0** through **7**.

- Runs with a userid that has special security privileges within the ACP.

If any of the above conditions is true, then the possibility exists that installing the product on MVS could introduce system integrity exposures. In such cases, follow the IBM system integrity guidelines to ensure that exposures are not created.

Areas of critical importance to OS/390 integrity, and the steps required to secure each of them, are discussed in the following sections. Details regarding a specific ACP implementation of these steps are discussed in that product's section of this document.

To facilitate maintaining operating system integrity, use the following guidelines:

(1) All products and exits should be installed and maintained with a process that utilizes a Change Management Process(CMP). IBM’s System Modification Program/Extended (SMP/E) is a very highly recommended product on the IBM mainframe and is the product of choice for sites implementing a CMP experience a stable environment by ensuring proper control and tracking of maintenance and changes.

(2) Test all products for security impacts in a test environment before being authorized for production.

(3) Integrity assurances are required to be obtained for any products or applications that require or provide operating system exits, SVCs (Supervisor Calls), I/O (input/output) appendages, special PPT (Program Properties Table) privileges, APF (Authorized Program Facility) authorization, or any other modifications to the operating system. This requirement may be satisfied by either of the following:

(a) A site should obtain a Vendor Integrity Statement (VIS) from the vendor of the product. A VIS meets the code review requirement for COTS as stated in Paragraph (b) below. Each site should centrally maintain the VIS. Contact the DISA FSO Technical Library for further information about Vendor Integrity Statements.

(b) Obtain the written approval of the site DAA to install and use the product, application, or system modification. This requires submitting (prior to installation) the following items to the site DAA for analysis and review:

- Documentation describing the product or application
- Documentation and source code for the system modification

(4) Any locally-developed extension to the operating system environment (e.g., exit, SVC, etc.) requires the following:

(a) Advance written approval of the concept by the site DAA before its development.

(b) Review and written approval of the concept's implementation by the site DAA before installation and utilization. This requires submitting documentation and source code for the local system modification to DISA FSO for analysis and review.

DISA FSO is required to provide responses to (a) and (b) above within 30 days of receipt of each request.

- *(AAMV0450: CAT II) The IAO will ensure that FSO approved documentation of additional system exits, SVCs. I/O appendages, PPT privileges, and APF authorization is on file to preserve the integrity of the Operating System.*

2.1.2.1 APF

The Authorized Program Facility (APF) is a component of OS/390 that allows installations to specify programs permitted to use sensitive system functions. Sensitive functions include items such as accessing protected OS/390 control blocks, accessing or modifying data from a different system task, or executing restricted hardware-level instructions.

A program designated as APF-authorized has the abilities to circumvent or disable the ACP, modify audit trails, or modify other data, despite the presence of access control software.

APF authorization within OS/390 is validated based on the characteristics of the first program called within a job step. For the program to be authorized, it must reside in and be loaded from an APF-authorized library. Also, the program must have been linked as authorized (**AC=1**). In addition, all libraries from which programs for the job step are to be subsequently loaded (concatenated libraries) must also be APF authorized. None of the programs loaded for the job step are authorized if any one of these conditions is not met.

Control over APF authorization is specified within the operating system. The data set SYS1.PARMLIB members IEAAPF*xx* and PROGxx are used to specify the library names and the volumes on which they reside. (The *xx* is the suffix designated by the APF and PROG parameters in the IEASYS*xx* member of SYS1.PARMLIB or overridden by the computer operator at system initial program load [IPL]).

NOTE: An entire library is listed as authorized, and not the individual modules themselves.

Use the following recommendations and techniques to control the exposures created by the APF facility:

(1) In SYS1.PARMLIB(IEASYS*xx*), use the parameter LNKAUTH=APFTAB so that all APF libraries are specified in the IEAAPFxx and PROGxx members of parmlib.

- *(AAMV0030: CAT II) The systems programmer will ensure that LNKAUTH=APFTAB is specified in the IEASYSxx member(s) in the currently active parmlib data set(s).*

(2) The IEAAPFxx and PROGxx members will contain only required libraries. On a semi-annual basis, Software Support should review the volume serial numbers, and should verify them in accordance with the system catalog. Software Support will remove all non-existent libraries. The IAO should modify and/or delete the rules associated with these libraries.

- *(AAMV0040: CAT IV) The systems programmer will ensure that only existing libraries are specified in the APF list of libraries.*

- *(AAMV0060: CAT IV) The systems programmer and IAO will ensure that procedures are in place to review APF-authorized Libraries and are reviewed at least on a semi-annual basis.*

(3) Before a library and a volume serial number are added to IEAAPFxx and PROGxx, the IAO will protect the data set from unauthorized access. Systems programming personnel will specify the requirements for users needing *read* or *execute* access to this library. Comparisons among all the APF libraries will be done to ensure that an exposure is not created by the existence of identically named modules. Address any sensitive utility concerns with the IAO, so that the function can be restricted as required. The IAO will build the appropriate protection into the ACP.

- *(AAMV0050: CAT III) The IAO will ensure that Duplicate sensitive utility(ies) and/or program(s) do not exist in APF-authorized libraries.*

(4) All *update* and *alter* access to the APF-authorized libraries will be logged using the ACP’s facilities. Only systems programming personnel will be authorized to update the APF-authorized libraries. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

- *(ACP00060: CAT II) The IAO will ensure that Update and allocate access to all APF-authorized libraries are limited to system programmers only, unless a letter justifying access is filed with the IAO, all update and allocate access is logged.*

The libraries specified in the APF list and the following additional libraries are authorized by OS/390:

SYS1.LINKLIB
SYS1.SVCLIB
SYS1.IMAGELIB (only when accessed by the appropriate SVC)
SYS1.LPALIB (only during the IPL process)

- *(ACP00020: CAT II) The IAO will ensure that update and allocate access to SYS1.LINKLIB is limited to system programmers only, unless a letter justifying access is filed with the IAO, and all update and allocate access is logged.*

- *(ACP00030: CAT II) The IAO will ensure that update and allocate access to SYS1.SVCLIB is limited to system programmers only, unless a letter justifying access is filed with the IAO, and all update and allocate access is logged.*

- *(ACP00040: CAT II) The IAO will ensure that update and allocate access to SYS1.IMAGELIB is limited to system programmers only, unless a letter justifying access is filed with the IAO, and all update and allocate access is logged.*

- *(ACP00050: CAT II) The IAO will ensure that update and allocate access to SYS1.LPALIB is limited to system programmers only, unless a letter justifying access is filed with the IAO, and all update and allocate access is logged.*

Several OEM products, such as OMEGAMON and CA-LOOK, provide the capability to perform dynamic APF library updates on an operational MVS system. This ability to modify the APF in real-time presents the danger of a severe integrity exposure. Restrict access to these products and sensitive functions to the absolutely minimum number of personnel.

The capability to perform dynamic APF library maintenance was introduced as an inherent feature of OS/390. The SYS1.PARMLIB(PROG*xx*) member controls this mechanism (which is similar in function to the dynamic APF library modification features discussed above).

The PROG*xx* member is an alternate way to define libraries in the APF list. Either or both can be used in a static APF environment. Use of the dynamic APF facility requires the installation of DFSMS/MVS 1.1 and exclusive use of the PROG*xx* member. Use of the APF facility also mandates the use of dynamic APF services by programs and system utilities accessing the APF list.

The command SET PROG=*xx* dynamically changes the APF list based on the information in the specified PROG*xx* member. The SETPROG APF command provides the capability to change the APF list between dynamic and static formats, and to selectively add and delete libraries from the APF list. The CSVAPF macro provides equivalent functionality to these commands.

The STIG recommendation is to **not** use the dynamic APF capabilities offered by OS/390 until proper controls restricting operator and console authorities are implemented. With the introduction of OS/390, controls are available for operator commands. If the security controls outlined in *Section 3.1.2.8, MCS Console Userids*, *Section 3.1.2.9, OS/390 System Operator Userids*, *Section 3.1.5.4, Dynamic List Controls*, *Section 3.1.5.5, MCS Console Controls*, and *Section 3.1.5.6, OS/390 System Command Controls*, are in place, installations may use the PROG*xx* APF facility.

Refer to *Section 2.1.2, Software Integrity*, for information on integrity statement requirements.

2.1.2.2 TSO APF Authorization

A legitimate need exists for Time Sharing Option (TSO) users to occasionally execute authorized programs. Although some of these (e.g., DELETE) are intended for general use, others (e.g., OMEGAMON) are sensitive utilities to which access shall be strictly controlled using the ACP. (Refer to *Section 3.1.5.3, Sensitive Utility Controls*, for further information.)

The TSO Terminal Monitor Program (TMP), IKJEFT01, initially runs as an APF-authorized program. It turns off authorization prior to invoking external commands and programs, although it continues running with a system storage-protect key. However, TSO provides a means to allow APF-authorized programs to be executed in a TSO user's address space. In an OS/390 environment, the process is controlled by the presence of the program name in one of two MVS load modules. These modules are IKJEFTE2 (used for callable programs) and IKJEFTE8 (used for actual TSO commands). Both modules reside in SYS1.LINKLIB. In an MVS/XA, MVS/ESA, or OS/390 environment, the programs for authorization are documented in SYS1.PARMLIB(IKJTSO*xx*) and in load modules IKJTABLS and IKJEFTAP, which are located in SYS1.LPALIB.

The TMP uses the RSAPF keyword of the ATTACH macro to invoke an external command or program in the proper APF state. The MVS Supervisor resets the TMP job step as authorized if the invoked program (1) has been link-edited with AC(1), and (2) comes from an authorized library concatenation or from the system Linklist.

Use the following recommendations and techniques to ensure that valid program names are specified for TSO authorization:

(1) Systems programming personnel are responsible for determining the programs required for TSO authorization by reviewing the product/program documentation.

(2) Use a CMP to install and maintain any user modifications (usermods) or programs that require TSO authorization.

(3) Review any programs requiring TSO authorization for potential impact to the operating environment. Provide documentation to the IAO to limit program access using sensitive utility controls.

(4) Perform any user modifications or updates to the TSO authorization modules using the CMP.

(5) Procedures to perform reviews of TSO authorized program(s) will exist and should be performed on a semi-annual basis.

- *(AAMV0090: CAT IV) The systems programmer and IAO will ensure that procedures are in place to review authorized TSO programs and are reviewed at least on a semi-annual basis.*

Refer to *Section 2.1.2, Software Integrity*, for information on integrity statement requirements.

2.1.2.3 PPT

Some programs require extraordinary privileges not normally permitted by the operating system. The Program Properties Table (PPT) contains the names and properties of these special programs. Programs in the PPT can bypass security software mechanisms such as password protection. Only programs that require special authorizations are coded in the PPT.

The PPT is maintained differently depending upon the level of MVS. Use the following recommendations and techniques to provide protection for the PPT:

(1) As part of standard MVS maintenance, systems programming personnel will review the IEFSDPPT module and all programs that IBM has, by default, placed in the PPT to validate their applicability to the execution system. *Appendix B, Sample Program Properties Table (PPT)*, shows the standard values for a sample PPT, as provided by IBM in module IEFSDPPT for MVS/ESA, Version 4.

Modules for products not in use on the system will have their special privileges explicitly revoked. Do this by placing a PPT entry for each module in the SYS1.PARMLIB(SCHEDxx) member, specifying no special privileges. The PPT entry for each overridden program will be in the following format, accepting the default (unprivileged) values for the sub-parameters:

```
PPT  PGMNAME(<program name>)
```

The Software Support team will assemble documentation regarding these PPT entries, and the IAO will keep it on file. Include the following in the documentation:

- The product and release for which the PPT entry was made
- The last date this entry was reviewed to authenticate status
- The reason the module's privileges are being revoked

- *(AAMV0140: CAT III) The systems programmer and IAO will ensure that procedures are in place to review PPT entries and are reviewed at least on a semi-annual basis.*

- *(AAMV0160: CAT II) The systems programmer will ensure that any invalid entries in the PPT via IEFSDPPT module or invalid entries in the SCHED PPT are nullified by (a) nullifying the invalid IEFSDPPT entry ensuring that there is a corresponding SCHED entry which confers no special attributes, or (b) removing the SCHED PPT entry which is no longer valid if it only exists in this member.*

(2) Systems programming personnel will review any additional module to be placed in the PPT to validate its requirements. Use the SYS1.PARMLIB(SCHEDxx) member for all programmer PPT entries. The Software Support team will assemble documentation regarding the PPT attributes. The IAO will keep this documentation on file. It will include the following:

- The product and release for which the PPT entry is made
- The file name in which this module resides
- The parameter(s) requiring that this be a PPT entry (e.g., NOPASS, NOSWAP)
- The last date this entry was reviewed to authenticate its need
- A reference to the documentation that further explains the requirement that this be a PPT entry (i.e., the specific product installation manual)

- *(AAMV0170: CAT III) The IAO will ensure that documentation for each module contained in the PPT is available.*

(3) Implement security controls to ensure that only authorized personnel can update the following:

- The library where the IEFSDPPT module resides (SYS1.LINKLIB)
- The library where the SCHEDxx member resides (SYS1.PARMLIB)

(4) All *update* and *alter* access to libraries containing programs specified in the PPT will be logged using the ACP's facilities. Only systems programming personnel will be authorized to update the libraries containing programs specified in the PPT. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

- *(ACP00100: CAT II) The IAO will ensure that update and allocate access to libraries containing PPT modules is limited to system programmers only, unless a letter justifying access is filed with the IAO, and all update and allocate access is logged.*

Refer to *Section 2.1.2, Software Integrity,* for information on integrity statement requirements.

2.1.2.4 SVCs

An Supervisor Calls (SVC) is a low-level language instruction that initiates an operating system interrupt. The operating system passes control to the SVC code to perform its processing. All SVC code runs in Supervisor State, which means that SVC code can potentially violate system integrity.

SVCs are divided into two categories. Unrestricted SVCs are available to all computer programs. Restricted SVCs are limited to authorized routines.

It is essential that all SVCs not provided by IBM with the MVS operating system are analyzed for integrity exposures. Each SVC provided by a third party or written locally should be examined to determine (1) its abuse potential, (2) whether any validity checking is being performed, and (3) the steps necessary to ensure protection.

SVCs can be limited to only allow use through authorized job steps. The SVC code itself may be modified to issue the TESTAUTH macro, restricting all or parts of the SVC code to programs running in an authorized mode. Refer to the IBM publications, *OS/390 MVS Security* and *OS/390 MVS System Programming Library (SPL),* for further information on this process.

Use the following recommendations and techniques to control SVCs:

(1) Systems programming personnel will review all SVCs on a semi-annual basis to confirm that the SVCs are required and correctly installed.

- *(AAMV0210: CAT IV) The systems programmer and IAO will ensure that procedures are in place to review SVCs and are reviewed at least on a semi-annual basis.*

- *(AAMV0230: CAT II) The IAO will ensure that SVCs are documented and registered.*

(2) Correct any deficiencies with an SVC, coordinating with the provider of the SVC.

(3) All *update* and *alter* access to the following libraries containing SVCs will be logged using the ACP's facilities. Only systems programming personnel will be authorized to update the libraries containing SVCs. Software Support will document all the requirements for a particular SVC. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

SYS1.SVCLIB
SYS1.NUCLEUS
SYS1.LPALIB
SYS1.LINKLIB
Any library defined in the LPALST*xx* member of SYS1.PARMLIB

- *(ACP00080: CAT II) The IAO will ensure that update and allocate access to SYS1.NUCLEUS is limited to system programmers only, unless a letter justifying access is filed with the IAO, and all update and allocate access is logged.*

(4) SVCs that combine both sensitive and non-sensitive functions shall include protection mechanisms to ensure that sensitive functions are adequately restricted. These protection mechanisms can include, but are not limited to, the use of the TESTAUTH macro.

(5) The IAO and the systems programming staff are to be aware of products that dynamically install their SVCs as part of product initialization, rather than including them in the SVC table definition.

(6) The IBM manual, *OS/390 MVS Planning: Security, GC28-1439, Chapter 5*, lists many common errors made in coding user (i.e., non-IBM supplied) SVCs. These common mistakes can present serious exposures to OS/390 integrity.

Refer to *Section 2.1.2, Software Integrity*, for information on integrity statement requirements.

2.1.2.5 I/O Appendages

An I/O appendage is a routine that provides additional control over system I/O operations. I/O appendages can examine the status of I/O operations and determine the actions to be taken for various conditions. An appendage may receive control in a variety of cases. Appendages receive control in the Supervisor State from the system EXCP processor. As such, they are very powerful and will be strictly controlled to avoid compromising system integrity. Appendages have the potential to circumvent or disable ACP files, to modify audit trails, or to modify other data despite the presence of access control software. CA-EXAMINE is the product chosen to automatically assemble a list of all I/O appendages.

Appendages must be members of either the SYS1.LPALIB or SYS1.SVCLIB data sets, and must be defined in the IEAAPP*xx* member of SYS1.PARMLIB, to be available to problem-state (unauthorized) programs. An unauthorized program is one that runs in a protection key greater than **7**, and has not been marked as authorized by the Authorized Program Facility.

Use the following recommendations and techniques to control the potential exposures created by the I/O appendage facility:

(1) The IEAAPPxx member will only contain valid I/O appendages required by unauthorized application programs. Software Support will remove appendages from IEAAPPxx when unauthorized programs no longer need them. Systems programming personnel will review all I/O appendages to confirm that the appendages are required and are correctly installed.

(2) Before I/O appendages are installed in the system libraries and added to IEAAPPxx, the IAO will verify that a valid requirement for their inclusion exists.

- *(AAMV0250: CAT III) The systems programmer and IAO will ensure that procedures are in place to review I/O appendages and are reviewed at least on a semi-annual basis.*

- *(AAMV0270: CAT III) The IAO will ensure that I/O appendages are properly documented or removed from the system.*

- *(AAMV0280: CAT III) The IAO will ensure that documentation of I/O appendages is maintained.*

Refer to *Section 2.1.2, Software Integrity*, for information on integrity statement requirements.

2.1.2.6 OS/390 and Other Product Exits

OS/390 and many other products provide exits that can be used to perform additional processing for an installation. Examples of these system-level exits are SMF, JES2, TSO, ISPF, CICS, OMEGAMON, and SDSF, to name but a few. Many of these exits have the potential to open integrity exposures since the code may be entered in an authorized state. Every exit point used within a product (especially MVS) needs to be validated so the code does not bypass the integrity of the operating environment. CA-EXAMINE is a product that can be used to automatically assemble a list of all product exits.

The capability to perform dynamic exit maintenance was introduced as an inherent feature of MVS/ESA, Version 5, Release 1. This mechanism is controlled by the CSVDYNEX macro, the SYS1.PARMLIB(PROG*xx*) member, the SET PROG=*xx* command, and the SETPROG EXITS command.

The command SET PROG=*xx* dynamically changes the exit definitions based on the information in the specified PROG*xx* member. The SETPROG EXITS command provides the capability to selectively add and delete exit routines from exit definitions.

Complete the following steps to maintain the integrity of the system:

(1) Document all exit points used. Provide this to the IAO for backup documentation purposes.

- *(AAMV0290: CAT III) The IAO will ensure that documentation of all installed OS/390 and other product exits are kept on file.*

(2) Using the ACP, protect the data sets associated with all product exits installed in the OS/390 environment. This reduces the potential of a hacker adding a routine to a library and possibly creating an exposure.

(3) Track all exits using a CMP. Develop usermods to include the source/object code used to support the exit.

(4) Systems programming personnel will review all OS/390 and other product exits to confirm that the exits are required and are correctly installed.

(5) All *update* and *alter* access to libraries containing OS/390 and other system-level exits will be logged using the ACP's facilities. Only systems programming personnel will be authorized to update the libraries containing OS/390 and other system-level exits. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

- *(ACP00240: CAT II) The IAO will ensure that update and Allocate access to Libraries containing EXIT modules is limited to system programmers only, unless a letter justifying access is filed with the IAO, all update and allocate access is logged.*

Refer to *Section 2.1.2*, *Software Integrity*, for information on integrity statement requirements.

2.1.2.7 Access Control Product Exits

The ACPs themselves present some of the greatest exposures. Each ACP allows installation exit points that can be used to support additional security-related requirements for a site. One example of such an exit is a password validation exit program restricting the contents of a password.

The process for controlling ACP exits is as follows:

(1) The site DAA approves the use of any ACP exit. Use of any exit without this approval is prohibited. All code will be provided to DISA FSO for review.

(2) DISA FSO reviews the exit for use and integrity, and evaluates the need for such an exit across all OS/390 platforms. If deemed a viable process, steps are taken to prepare the exit for distribution to applicable sites.

(3) Systems programming personnel reviews all ACP exits to confirm that the exits are required and are correctly installed.

(4) All *update* and *alter* access to libraries containing ACP exits will be logged using the ACP's facilities. Only systems programming personnel will be authorized to update the libraries containing ACP exits. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

- *(AAMV0290: CAT III) The IAO will ensure that Documentation of all installed OS/390 and other product exits are kept on file.*

- *(AAMV0450: CAT II) The IAO will ensure that FSO approved documentation of additional system exits, SVCs. I/O appendages, PPT privileges, and APF authorization is on file to preserve the integrity of the Operating System.*

- *(ACP00240: CAT II) The IAO will ensure that update and Allocate access to Libraries containing EXIT modules is limited to system programmers only, unless a letter justifying access is filed with the IAO, all update and allocate access is logged.*

Refer to *Section 2.1.2, Software Integrity*, for information on integrity statement requirements.

2.1.2.8 Link Pack Area

The system Link Pack Area (LPA) is the component of MVS that maintains core operating system functions resident in main storage. A security concern exists when libraries from which LPA modules are obtained require APF authorization.

Control over residence in the LPA is specified within the operating system in the following members of the data set SYS1.PARMLIB:

- LPALST*xx* specifies the names of libraries to be concatenated to SYS1.LPALIB when the LPA is generated at IPL in an MVS/XA or MVS/ESA system. (The *xx* is the suffix designated by the LPA parameter in the IEASYS*xx* member of SYS1.PARMLIB or overridden by the computer operator at system initial program load [IPL].)

- IEAFIX*xx* specifies the names of modules from SYS1.SVCLIB, the LPALST*xx* concatenation, and the LNKLST*xx* concatenation that are to be temporarily fixed in central storage in the Fixed LPA (FLPA) for the duration of an IPL. (The *xx* is the suffix designated by the FIX parameter in the IEASYS*xx* member of SYS1.PARMLIB or overridden by the computer operator at IPL.)

- IEALPA*xx* specifies the names of modules that will be loaded from the following:

  - SYS1.SVCLIB
  - The LPALST*xx* concatenation
  - The LNKLST*xx* concatenation as a temporary extension to the existing Pageable

LPA (PLPA) in the Modified LPA (MLPA) for the duration of an IPL. (The *xx* is the suffix designated by the MLPA parameter in the IEASYS*xx* member of SYS1.PARMLIB or overridden by the computer operator at IPL.)

Use the following recommendations and techniques to control the exposures created by the LPA facility:

(1) The LPALSTxx, IEAFIXxx, and IEALPAxx members will contain only required libraries. On a semi-annual basis, Software Support should review the volume serial numbers, and should verify them in accordance with the system catalog. Software Support will remove all non-existent libraries. The IAO should modify and/or delete the rules associated with these libraries.

- *(AAMV0325: CAT IV) The systems programmer will ensure that only existing libraries are specified in the LPA list of libraries.*

- *(AAMV0335: CAT IV) The systems programmer and IAO will ensure that procedures are in place to review LPA Libraries and are reviewed at least on a semi-annual basis.*

(2) Before a library is added to LPALSTxx, IEAFIXxx, or IEALPAxx, the IAO will protect the data set from unauthorized access. Systems programming personnel will specify the requirements for which users require *read* or *execute* access to this library. Comparisons among all the LPA libraries will be done to ensure that an exposure is not created by the existence of identically-named modules. Address any sensitive program concerns to the IAO so that the function can be restricted as required. The IAO will build the appropriate protection into the ACP.

(3) All *update* and *alter* access authority to the LPA libraries will be logged using the ACP's facilities. Only systems programming personnel will be authorized to update the LPA libraries. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

- *(ACP00070: CAT II) The IAO will ensure that update and allocate access to all LPA libraries is limited to system programmers only, unless a letter justifying access is filed with the IAO, and all update and allocate access is logged.*

Several OEM products, such as OMEGAMON and CA-LOOK, provide the capability to perform dynamic LPA updates on an operational MVS system. This ability to modify the LPA in real-time presents the danger of a severe integrity exposure. Restrict access to these products and sensitive functions to the absolutely minimum number of necessary personnel.

OS/390 V2 introduced the capability to dynamically update the LPA as a standard feature of MVS. Refer to *Section 3.1.5.6, OS/390 System Command Controls*, for additional information on operator command controls.

Refer to *Section 2.1.2, Software Integrity*, for information on integrity statement requirements.

2.1.2.9 Linklist

The Linklist is a default set of libraries that MVS searches for a specified program. This facility is used so that a user does not have to know the library names in which utility types of programs are stored. Control over membership in the Linklist is specified within the operating system. The data set SYS1.PARMLIB(LNKLST*xx*) is used to specify the library names. (The *xx* is the suffix designated by the LNK parameter in the IEASYS*xx* member of SYS1.PARMLIB, or overridden by the computer operator at IPL.)

Use the following recommendations and techniques to control the exposures created by the LINKLIST facility:

(1) Avoid inclusion of sensitive libraries in the LNKLSTxx member unless absolutely required.

(2) The LNKLSTxx and PROGxx (LNKLST entries) members will contain only required libraries. On a semi-annual basis, Software Support should review the volume serial numbers, and should verify them in accordance with the system catalog. Software Support will remove all non-existent libraries. The IAO should modify and/or delete the rules associated with these libraries.

- *(AAMV0350: CAT IV) The systems programmer will ensure that only existing libraries are specified in the Linklist list of libraries.*

- *(AAMV0355: CAT IV) The systems programmer and IAO will ensure that procedures are in place to review Linklisted Libraries and are reviewed at least on a semi-annual basis.*

(3) Before a library is added to LNKLSTxx, the IAO will protect the data set from unauthorized access. Systems programming personnel will specify the requirements, for which users need *read* or *execute* access to this library. Comparisons among all the Linklist libraries will be done to ensure that an exposure is not created by the existence of identically-named modules. Address any sensitive program concerns to the IAO so that the function can be restricted as required. The IAO will build the appropriate protection into the ACP.

(4) All *update* and *alter* access authority to Linklist libraries are logged using the ACP's facilities. Only systems programming personnel are authorized to update the Linklist libraries. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

- *(ACP00110: CAT II) The IAO will ensure that update and allocate access to LINKLIST libraries is limited to system programmers only, unless a letter justifying access is filed with the IAO, and all update and allocate access is logged.*

Refer to *Section 2.1.2, Software Integrity*, for information on integrity statement requirements.

The capability to perform dynamic Linklist maintenance was introduced as an inherent feature of OS/390, Version 1, Release 3. This mechanism is controlled by the SYS1.PARMLIB(PROG*xx*) member, the SET PROG=*xx* command, and the SETPROG LNKLST command.

The PROG*xx* member is an alternate way to define libraries in the Linklist. Either or both can be used in a static APF environment. Use of the dynamic Linklist requires exclusive use of the PROG*xx* member. Normally a job step continues to use the same Linklist until it terminates, even if the operator has defined a new Linklist.

The command SET PROG=*xx* dynamically changes the Linklist based on the information in the specified PROG*xx* member. The SETPROG LNKLST command provides the capabilities to selectively add and delete libraries from the existing Linklist, activate an entirely new Linklist, or update an address space to use the redefined Linklist.

2.1.2.10 SMF Data Collection

DODI 8500.2 requires the collection and retention of audit data to support technical analysis relating to misuse, penetration reconstruction, or other investigations. The STIG requires the audit trail to record the identity of the user, time of access, interaction with the system, and sensitive functions that might permit a user or program to modify, bypass, or negate security safeguards. SMF data presents a critical component in providing the required audit trails to maintain OS/390 system integrity. All relevant SMF data will be collected and retained for at least one year to ensure that adequate audit trails are available.

At a minimum, always collect SMF data relating to the following areas:

Table A-1. SMF DATA COLLECTION (2.1.2.10)

SMF DATA COLLECTION	
AREA	RECORD TYPE(S)
IBM Products	See *Appendix C* for minimum requirements
ACF2	230 (vendor-supplied default) or as specified in the ACFFDR
TOP SECRET	80 (Uses IBM Record)
TSOMON	199

NOTE: The above table lists only those SMF record types that contain security or security-related information. It is not a comprehensive list of all SMF data to be collected for all purposes (e.g., accounting or capacity planning).

- *(AAMV0380: CAT II) The IAO will ensure that SMF recording options are consistent with those outlined in this STIG.*

Options for SMF data recording are controlled by the parameters of **SYS1.PARMLIB(SMFPRM*xx*)**. These parameters control the frequency of collection, the checkpoint interval for long-running tasks, the level of detail recorded, and the SMF record types to be collected, among others.

The settings for several parameters are critical to the collection process:

ACTIVE	Activates the collection of SMF data.
JWT(15)	The maximum amount of consecutive time that an executing job may spend as ineligible to use any CPU resources before being canceled for inactivity. The STIG requirement for Job Wait Time is 15 minutes. (This may be extended if controlled through other means, e.g., a Session Manager or ACP.)
MAXDORM(0500)	Specifies the amount of real-time that SMF allows data to remain in an SMF buffer before it is written to a recording data set.
SID	Specifies the system ID to be recorded in all SMF records.
SYS(DETAIL)	Controls the level of detail recorded.
SYS(INTERVAL)	Ensures the periodic recording of data for long-running jobs.
SYS	Specifies the types and sub-types of SMF records that are to be collected. SYS(TYPE) indicates that the supplied list is inclusive (i.e., specifies the record types to be collected). Record types not listed are not collected. SYS(NOTYPE) indicates that the supplied list is exclusive (i.e., specifies those record types not to be collected). Record types not listed are not collected. The site may use either form of this parameter to specify SMF record type collection. However, at a minimum all record types listed in the *SMF Data Collection* table above will be collected.

(1) The SMFPRMxx member will specify, at a minimum, all record types noted in the *SMF Data Collection* table above. SMF data collection will be activated. Specify a unique system ID for each domain to ensure that SMF data can be discretely identified to, and associated with, the domain where it originated.

- *(AAMV0370: CAT II) The IAO will ensure that collection options for SMF Data are consistent with options specified in this STIG.*

(2) All *update* and *alter* access authority to SMF files (MANx) will be logged using the ACP's facilities. Only systems programming personnel will be authorized to update the SMF files (MANx). The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

- *(ACP00180: CAT II) The IAO will ensure that update and allocate access to SMF collection files (i.e.SYS1.MANx) is limited to system programmers and/or batch jobs that perform SMF dump processing, unless a letter justifying access is filed with the IAO, and all dataset access is logged.*

(3) To ensure that all SMF data is collected in a timely manner, and to reduce the risk of data loss, the site will ensure that automated mechanisms are in place to collect and retain all SMF data produced on the system. Dump the SMF files (MANx) in DOD systems based on the following guidelines:

(a) Dump each SMF file as it fills up during the normal course of daily processing.

(b) Dump all remaining SMF data at the end of each processing day.

- *(AAMV0400: CAT II) The IAO will ensure that an automated process is in place to collect SMF data.*

(4) In OS/390 systems, SMF data is the ultimate record of system activity. Therefore, SMF data is of the most sensitive and critical nature. While the length of time for which SMF data will be retained is not specifically regulated, it is imperative that the information is available for the longest possible time period in case of subsequent investigations. The statute of limitations varies according to the nature of a crime. It may vary by jurisdiction, and some crimes are not subject to a statute of limitations. Apply the following guidelines to the retention of SMF data for all DOD systems:

(a) Retain at least two (2) copies of the SMF data.

(b) Maintain SMF data for a minimum of one year.

(c) All *update* and *alter* access authority to SMF history files will be logged using the ACP’s facilities. Only systems programming personnel and batch jobs that perform SMF functions will be authorized to update the SMF files. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

- *(ACP00190: CAT II) The IAO will ensure that update and allocate access to datasets used to backup and/or dump SMF collection files is limited to system programmers and/or batch jobs that perform SMF dump processing, unless a letter justifying access is filed with the IAO, and all dataset access is logged.*

Refer to *Section 3.1.5.1, Data Set Controls*, for further information.

2.1.2.11 Logical Parmlib

The IAO will implement controls to specify the valid users authorized to update the SYS1.PARMLIB concatenation. (Refer to the specific Access Control Product section regarding data set access controls.) All *update* and *alter* access to libraries in the concatenation will be logged using the ACP's facilities. Only systems programming personnel will be authorized to update the SYS1.PARMLIB concatenation. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports.

- *(ACP00010: CAT II) The IAO will ensure that update and allocate access to SYS1.PARMLIB is limited to system programmers only, unless a letter justifying access is filed with the IAO, and all update and allocate access is logged.*

The capability to perform dynamic logical parmlib maintenance was introduced as an inherent feature of OS/390, Version 1, Release 2. This mechanism is controlled by the IEFPRMLB macro, the SYS1.PARMLIB(LOAD*xx*) member, and the SETLOAD command.

The LOAD*xx* member is a way to define libraries in a logical parmlib, i.e., a concatenation of libraries to be searched in place of SYS1.PARMLIB.

The command SETLOAD *xx* dynamically changes the logical parmlib based on the information in the specified LOAD*xx* member.

2.1.2.12 MCS Consoles

The capability to control OS/390 system commands with the ACP was introduced as an inherent feature of MVS/ESA SP, Version 4, Release 3. As part of that support, the installation can require the operator to log on to an MCS console prior to entering any operator commands. The SYS1.PARMLIB(CONSOL*xx*) member controls this mechanism.

Use the following recommendations and techniques to provide protection for MCS consoles:

(1) Give every console an explicit console ID, and define that ID to the ACP as a user with only those access rights required for use of the console. Define every console, including extended MCS consoles, with AUTH(INFO).

(2) In SYS1.PARMLIB(CONSOL*xx*), specify the parameter LOGON(REQUIRED) on the DEFAULTS statement so that all operators are required to log on prior to entering OS/390 system commands. At the discretion of the IAO, LOGON(AUTO) may be used, provided the console userids are only authorized to use the CONTROL, DISPLAY, MONITOR, STOPMN, STOPTR, and TRACK commands, and their access is limited to *read* level.

(3) The IAO will implement and document controls as described in *Section 3.1.5, Resource Controls*.

- *(ACP00291: CAT II) The system programmer will ensure that the CONSOLxx members are properly configured.*

- *(ACP00292: CAT II) The IAO will ensure that all consoles identified in the CONSOLxx members are defined to the ACP.*

- *(ACP00282: CAT II) The IAO will ensure that OS/390 Sensitive System Commands are defined to the OPERCMDS resource class. Only limited number of authorized people are able to issue these commands. All access is logged.*

2.1.3 Object Reuse

Within the criteria for achieving MAC II Sensitive compliance, a requirement exists for ensuring the integrity of an object when reused. According to *A Guide to Understanding Object Reuse in Trusted Systems, NCSC-TG-018*, a definition of object reuse is "…. to prevent a user who is allocated storage from accessing information put there by a previous user." *DODI 8500.2* does not specify how object reuse requirements are to be technically implemented, but allows the system flexibility in meeting the requirement.

To meet the criteria for the entire operating environment, the use of an automatic erasing facility is required since the operating system itself cannot provide this level of integrity. However, the use of such a facility can cause considerable system overhead in actually erasing the data from a device. Additionally, such a tool cannot erase data from magnetic tape media or from other removable media, thus requiring additional physical security controls.

2.1.3.1 Unclassified Systems

The CIO (Chief Information Officer) has granted an exemption from the object reuse requirement for unclassified systems. For these systems the automatic ERASE option within the ACP may be disabled. Optionally, the site may enable the option if so dictated by customer requirements.

The customer community should determine the sensitivity of their data according to applicable policies and regulations, and than submit their object reuse requirements to the site. The site is responsible for providing the means by which the requirement is then met.

The site should determine their object reuse requirements and provide the mechanisms to ensure the requirements are met. For systems processing **unclassified** and **sensitive** data, a solutions should be provided to allow the site to meet the object reuse requirements for identified sensitive data sets.

The site should consider optionally protecting some of its own sensitive data with automatic erasure. As an example, the following files would be candidates for object reuse protection:

- System Management Facility data
- Any file possessing information
- Security audit data and reports
- All billing-related data

2.1.3.2 Classified Systems

For any system that processes classified information, object reuse requirements shall be met by enabling the automatic ERASE option within the ACP. This ensures that all DASD data is overwritten as files are deleted.

Any data on removable media (reel tape, tape cartridges, diskettes, etc.) are to be handled manually using approved degaussing procedures. For information regarding approved degaussers, consult the current National Computer Security Center (NCSC) *Evaluated Products List*.

The use of RAID (Redundant Array of Inexpensive Disks) technology presents a special problem to classified processing. Not all implementations of RAID technology support automatic, in-place erasure and destruction of data. Some implementations simply change table pointers and recover the storage locations for reuse, without destroying the stored data.

It is likely that as RAID technology advances, methods for recovering the data may be developed to recover inadvertently deleted data. Such methods would enable the recovery of any data not destroyed in place, potentially compromising the information. For these reasons, only those implementations of RAID technology that support the automatic erasure features of the ACPs and perform in-place destruction of stored data shall be used on classified systems.

2.1.4 Auditing

In an effort to minimize the risks inherent with the OS/390 environment, DOD has directed that a process be developed and implemented to identify the potential exposures presented by MVS, and that measures be taken to ensure that the system's integrity is not compromised.

As part of the requirement to review OS/390, documentation supporting the performance of a review should be maintained by the site.

2.1.4.1 Review and Documentation Requirements

All locally written exits or modifications to the operating system, or products that act as extensions to the operating system, be provided to DISA FSO for analysis and approval prior to installation in a production environment.

Whenever possible, vendor software should be accepted as trusted and approved providing the site DAA has a Vendor Integrity Statement (VIS) on file. If a VIS is not on file or cannot be obtained from the vendor, source code for vendor-supplied software should be obtained and submitted to DISA FSO for an integrity analysis. All locally written exits or modifications to the OS must have the source code and other applicable documentation submitted to DISA FSO for program integrity analysis.

Refer to *Section 2.1.2, Software Integrity*, for information on integrity statement requirements.

2.1.4.2 Documentation Process

The site should ensure that a repository for documentation of the following system resources are filed at the site:

- APF-Authorized TSO table entries
- SVCs
- Exits
- I/O appendages

2.1.4.3 Audit Logs

The review of audit logs is an essential function of comprehensive security. Audit logs contain the information necessary to detect unauthorized attempts to gain access to a computer system or the computer systems resources. Audit logs record events such as logon attempts, data set access attempts, resource access attempts, etc.

On OS/390 platforms, SMF records comprise the primary audit logs. Refer to *Section 2.1.2.10, SMF Data Collection*, for more information. As events occur on the system initiated by individual users, batch jobs, and STCs, OS/390 and the ACP create SMF records of this activity and writes them to the SMF record file.

Although each ACP is unique in its implementation, each can be configured to provide essentially the same information in the SMF records it creates. Each ACP also provides a set of reports that are used to report on security exceptions. The reports use the SMF records created by the ACP as input to create the reports. Using the ACP's native reports is not a requirement however. Depending on the situation, it is sometimes preferable to use a different mechanism for reporting, such as collecting the records/reports in a database and designing custom reports. A domain's IAO has the responsibility of reviewing a domains audit logs regardless of the reporting method. Due to the potentially large amount of audit records, the IAO may employ other IAOs or security administrators to assist in the review of the audit reports.

Following are the requirements concerning the review of audit logs/security reports. Sites will keep samples of the review on file to demonstrate compliance.

On a daily basis, the following audit entries will be reviewed:

- Data Set Access Violations – Sensitive data sets (i.e., APF authorized libraries, ACP libraries, LPA libraries, LINKLIST libraries, etc.) are the top priority. Look for patterns of denied access. Does the same user or group of users keep showing up in the log?

- Resource Violations – This is a varied category that includes items such as transactions (e.g., CICS, DB2, IDMS, CA1, etc.), to job classes, to operator commands, etc.

- Program Use Violations – Applies primarily to Sensitive Utilities (refer to *Section 3.1.5.3, Sensitive Utility Controls)*. Attempts to use these by unauthorized users and jobs need to be questioned.

On a weekly/monthly basis (more often if time permits), the following audit entries will be reviewed:

- Failed Logon Attempts – Password miss-types are common. The reviewer should be looking for excessive violations for a single user or a block of users such as a department.

- Special Privileges – Changes to logonids where special privileges or attributes (e.g., SECURITY for ACF2, ACCESS[ALL] for TOP SECRET, SPECIAL for RACF) were given need to be reviewed for legitimacy.

On a quarterly basis, the following audit entries will be reviewed:

- The accesses from TRUSTED STCs to ensure there is no abuse occurring
- FTP userids/logonids/ACIDs to ensure their accesses are appropriate

Global Control Options – Any changes need to be evaluated to ensure they were authorized and legitimate.

- *(ACP00320: CAT II) The IAO will ensure that the ACP audit logs are reviewed as specified above in the STIG.*

2.2 Data-level Integrity

The concept of data-level integrity involves the protection of actual data loaded on the system. Data-level integrity is composed of data integrity and data labels.

This STIG is not intended to address data-level integrity in detail, but to provide techniques that can be used to ensure security of the data residing under the control of MVS.

2.3 Control Requirements

Data set controls play a critical role in maintaining the integrity of MVS systems. Several key areas of control requirements are discussed in the following sections.

2.3.1 Data Set Integrity

Data set integrity is a key factor in the protection of MVS systems. Critical system data sets that will be protected include, but are not limited to, the following:

- System catalogs (Master Catalog and User Catalogs)
- System libraries:
  - OS/390 libraries (e.g., Linklist, LPA, SVC, parmlib concatenation, IODF [Input/Output Definition File])
  - System-level product libraries (e.g., CA-1)
  - OS/390 and product installation libraries (e.g., the SMP/E CSI, DLIBs)
- Access Control Product files and databases
- JES2 SPOOL file (SYS1.HASPACE)
- JES2 SPOOL checkpoint file (SYS1.HASPCKPT)
- User attribute data set (SYS1.UADS)
- SMF data files (SYS1.MANx)
- System and subsystem trace data sets (e.g., GTF, OS/390 Component Trace)
- System dump data sets (SYS1.DUMP*xx*)

- Logs
- Backups, dumps, and off-loads of the above (e.g., JES2 SPOOL off-loads, external writer output from SYSLOG, SMF dumps, system DASD dumps)
- System page data sets (PLPA, COMMON, and LOCAL)
- Parameter

Enforce control restrictions for these files to ensure that (1) only those routines or users with a legitimate need are granted access, and (2) the access granted is restricted to the minimum level necessary. For example, DASD management routines should have access to backup files, and systems programming personnel should have access to system dump data sets.

- *(ACP00150: CAT II) The IAO will ensure that update and allocate access to JES2 System datasets (spool, checkpoint, and parmlib datasets) are limited to system programmers only, unless a letter justifying access is filed with the IAO.*

- *(ACP00170: CAT II) The IAO will ensure that allocate access to SYS1.UADS is limited to system programmers only, read and update access to SYS1.UADS is limited to system programmer personnel and/or security personnel, unless a letter justifying access is filed with the IAO, and all dataset access is logged.*

- *(ACP00200: CAT II) The IAO will ensure that update and allocate access to SYS1.DUMP data set(s) is limited to system programmers only, unless a letter justifying access is filed with the IAO.*

- *(ACP00210: CAT II) The IAO will ensure that update and allocate access to System backup files is limited to system programmers and/or batch jobs that perform DASD backups, unless a letter justifying access is filed with the IAO.*

- *(ACP00220: CAT II) The IAO will ensure that update and allocate access to SYS1.TRACE is limited to system programmers only, unless a letter justifying access is filed with the IAO.*

- *(ACP00230: CAT II) The IAO will ensure that update and allocate access to SYSTEM PAGE datasets (i.e., PLPA, COMMON, and LOCALx) is limited to system programmers only, unless a letter justifying access is filed with the IAO.*

2.3.2 File Location

File location is an often-overlooked factor in system integrity. It is important to ensure that the effects of hardware failures on system integrity and availability are minimized. Avoid collocation of files such as primary and alternate databases. For example, the loss of the physical volume containing the ACP database should not also cause the loss of the ACP backup database as a result of their collocation. Files that will be segregated from each other on separate physical volumes include, but are not limited to, the ACP database and its alternate or backup file.

- *(AAMV0410: CAT II) The systems programmer will ensure that placement of ACP files are on a separate volume from its backup and recovery data sets to provide backup and recovery in the event of physical damage to a volume.*

2.3.3 File Backup

Adequate backup scheduling is also an often-overlooked integrity exposure. Back up system files on a regular schedule. Store the backups off-site to prevent concurrent loss of the live production system and the backup files. Backup scheduling will vary depending on the requirements and capabilities of the individual data center.

While the requirements of Data Owners may necessitate more frequent backups, a recommended schedule is as follows:

- Weekly and monthly full-volume backup of volumes with low update activity, such as the operating system volumes

- Nightly backup of high *update* activity data sets and volumes, such as application system databases and user data volumes

- *(AAMV0430: CAT II) The IAO will ensure that procedures are in place to backup the operating system and all its subsystems on a regularly scheduled interval as required to recover the environment.*

At a minimum, nightly backup of the ACP databases, and of other critical security files (such as the ACP parameter file). More frequent backups (two or three times daily) will reduce the time necessary to effect recovery. The IAO will verify that the backup job(s) ran successfully.

- *(AAMV0420: CAT II) The IAO will ensure that procedures are in place to backup all ACP files needed for recovery on a scheduled basis.*

2.3.4 File Recovery

The logical complement to adequate backup of data is to have a written, verified, and regularly practiced recovery procedure for each manner of backup. Responsible personnel should have timely and adequate access to the recovery procedures, and should be thoroughly experienced in their execution to lessen the impact of recovery. These procedures should be tested annually.

2.4 Password Data Sets

Access to data sets on OS/390 systems can be protected using the OS password capability of MVS. This capability has been available in MVS for many years, and its use is commonly found in data centers. Since the advent of ACPs, the use of OS passwords for file protection has diminished, and is commonly considered archaic and of little use. The use of OS/390 passwords is not supported by all the ACPs.

- *(AAMV0440: CAT II) System programmers will ensure that the old OS Password Protection is not used and any data protected by the old OS Password technology is removed and protection is replaced by the ACP.*

2.5 OS/390 UNIX System Services

OS/390 UNIX System Services, abbreviated by IBM as OS/390 UNIX, provides a UNIX environment to OS/390 users. It is now a base component of the OS/390 operating system, conforms to the XPG4 UNIX 1995 standard (with UNIX 98 elements), and offers services designed to support applications written to open systems standards. OS/390 UNIX also provides OS/390 users the traditional UNIX structure for data storage through the Hierarchical File System (HFS). Finally OS/390 UNIX supports the UNIX User Identifier (UID) and Group Identifier (GID) concepts that establish identity in the UNIX environment.

In OS/390 UNIX, security is handled, in part, through the UID and GID constructs that identify users and groups. This security impacts file access and process (e.g., OS/390 task) control. While it is possible in some environments for multiple users to be assigned the same UID, this does not provide a desirable level of security.

OS/390 UNIX provides an operating environment that can host many services such as File Transfer Protocol (FTP) and OS/390 UNIX Telnet servers. In addition, OS/390 components such as Communications Server provide support to OS/390 UNIX. This section of this document is intended to describe the security considerations for the OS/390 UNIX environment and does not cover these supporting and supported components in appropriate detail. Please check other sections of this document and the pertinent vendor documentation for security considerations for these other components.

2.5.1 General Considerations

Because of the scope of OS/390 UNIX and its difference from the traditional MVS environment, there are a number of considerations that must be addressed to understand the security implications. In this section, security considerations for the following areas are discussed:

- User Identity – UID and GID Assignment
- Data Storage – HFS Directories and Files
- Interactive Environment – The UNIX Shell
- Background Processes – Daemons and Servers
- Miscellaneous Considerations

These considerations are discussed in general to explain the OS/390 UNIX environment. This background is used in *Section 2.5.2, Security Controls,* when discussing the specific controls that are used to implement security policy. Readers are advised to review the IBM OS/390 UNIX documentation that is noted in *Appendix A, Related Publications*, for a complete discussion of these items.

2.5.1.1 User Identity

Within UNIX systems, users are assigned a user name and password that allow identification and authentication when the system is accessed. Each user is also assigned a numeric identifier that is known as the UID. Users are members of one or more groups; each of these groups has a name and a numeric identifier that is known as the GID. While it is possible in some environments to assign multiple users the same UID, this is not done where meaningful security is desired.

There are no software-specific UID or GID numbers, with one exception. If a user is assigned a UID value of 0 (zero), the user has *superuser* status and effectively bypasses all security checks. There are a limited number of instances where superuser status is actually needed, and OS/390 UNIX provides some security resources that can be used to further limit the need to assign UID(0) to users.

During a UNIX shell session or during the execution of commands with certain attributes, it is possible for a user to temporarily use a different UID or GID value than what was assigned. The userid defined to the security system and used at system sign-on is referred to as the real ID. The temporary userid used for a specific period or process is referred to as the effective ID. For this reason it is important to check the effective ID when researching access control issues.

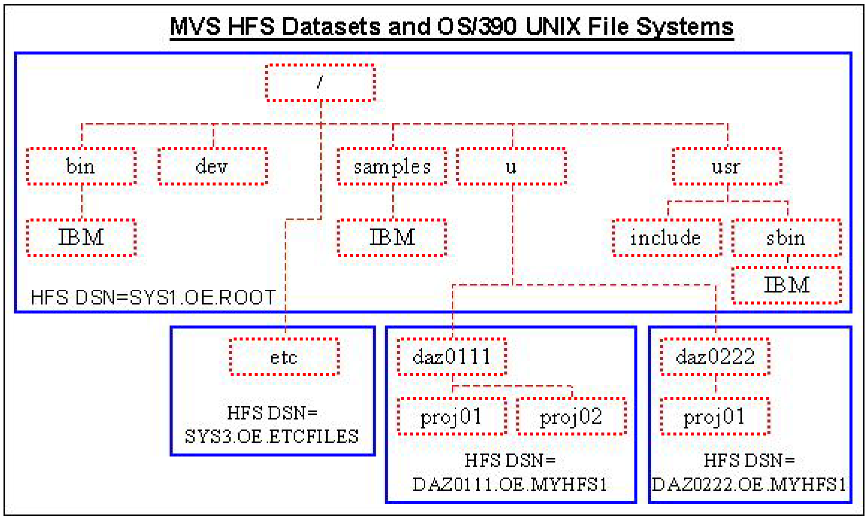
2.5.1.2 Data Storage – HFS Directories and Files

This section discusses the considerations related to data storage in the OS/390 UNIX environment. These considerations include the logical and physical structures, file access permissions, extended attributes for executable files, and audit attributes. Understanding these considerations is important to setting and maintaining data and command security.

Hierarchical File System (HFS) is a tree structure consisting of multiple file systems. A file system is a logical collection of directories and files. The highest level directory in the hierarchy is the root directory; it is often kept in a file system with only a few other directories. Each file system is made available by a process known as mounting the file system. It is mounted at a *mount point* that is actually just a directory in the higher-level file system.

The entire file hierarchy is made up of a collection of HFS data sets. Each physical HFS data set is actually a mountable file system. This means that it can be attached to the HFS tree at a mount point that is in the root directory or at a mount point further down in the hierarchy. Each HFS data set needs data set access rules defined to protect it.

The following diagram illustrates the relationship between MVS HFS data sets and OS/390 UNIX File Systems. This is an example with four MVS data sets (SYS1.OE.ROOT, SYS3.OE.ETCFILES, DAZ0111.OE.MYHFS1, and DAZ0222.OE.MYHFS1) corresponding to four OS/390 UNIX file systems (*root*, etc., daz0111, daz0222).



To provide granularity in access control, there are three sets of permission bits to accommodate three categories of users whose access can be individually controlled:

Owner – The user whose UID matches the UID in the FSP
Group – A member of the group whose GID matches the GID in the FSP
Other – Anyone else

When permission bits are displayed in command output or used as command operands, they sometimes appear as a string of alphabetic characters and sometimes as a string of octal digits that correspond to these categories. For example, a file can have permissions set to "rwx r-- ---", where "rwx" applies to the owner, "r--" to the group, and "---" to other. This would be expressed digitally as 740 where 7 applies to the owner, 4 to the group, and 0 to other.

The following tables show the permission bits, their alphabetic symbolic notation, their octal values, and their meaning:

Table A-2. PERMISSION BITS (2.5.1.2 a)

PERMISSION BITS			
PERMISSION	SYMBOLIC NOTATION	OCTAL VALUE	MEANING FOR FILE OR DIRECTORY
read	r	4	Directory: Allows the user to read, but not search, contents. File: Allows the user to read or print contents. *NOTE*: Running shell scripts requires read and execute.

PERMISSION BITS			
PERMISSION	SYMBOLIC NOTATION	OCTAL VALUE	MEANING FOR FILE OR DIRECTORY
write	w	2	Directory: Allows the user to change the directory, adding or deleting members. File: Allows the user to change the file, adding or deleting data.
execute	x	1	Directory: Allows the user to search the directory. File: Allows the user to run the executable program. *NOTE:* Running shell scripts requires read and execute.
no access	-	0	No access allowed.

There are additional permission bits that are used for special purposes. When in use, these bits may be displayed alphabetically in the *execute* position, with lower case indicating that the execute bit and special bit are both on. When displayed or used in a command in digital form, the value for these bits appears as an additional first digit in the string.

Table A-3. SPECIAL PERMISSION BITS (2.5.1.2 b)

SPECIAL PERMISSION BITS			
PERMISSION	SYMBOLIC NOTATION	OCTAL VALUE	MEANING FOR FILE OR DIRECTORY
set-user-ID set-group-ID	s/S	4 2	Used for an executable file, sets the effective userid and/or group ID of the user process executing the program to that of the file being executed. Allows a program to have temporary access to files (or potentially commands) that are not normally accessible.
sticky bit	t/T	1	Directory: Allows only the file owner, directory owner, or superuser to delete or rename files. File: Causes the search for an executable in the current STEPLIB, link pack area, or link list (the data in the HFS file is not loaded as the program).

These permissions are combined as required to allow the desired access.

The chown, chgrp, and chmod shell commands are provided. Refer to *Section 2.5.1.3, Interactive Environment – The UNIX Shell,* for information on these commands.

NOTE: The ACF2 and TOP SECRET ACPs offer an option called CA SAF HFS security. If this option is enabled, file mode checking is bypassed in favor of access rules written for the ACP. However, because CA SAF HFS can be disabled, the standard UNIX file permissions must be maintained for system sensitive directories and files.

OS/390 UNIX adds the feature of *extended attributes* that are meaningful for executable files. These extended attributes include the following:

Table A-4. EXTENDED ATTRIBUTES (2.5.1.2 c)

EXTENDED ATTRIBUTES		
EXTENDED ATTRIBUTE	SYMBOLIC NOTATION	DESCRIPTION
APF-authorized	a	Executable program acts as if loaded from an APF-authorized MVS library.
Program-controlled	p	Executable program acts as if defined to program control in the ACP.
Shared	s	Executable foreground program runs in the same MVS address space as the user's OS/390 shell. ***NOTE***: This bit is on as the default for all executable files.

To maintain the extended attributes, the extattr shell command is provided. Refer to *Section 2.5.1.3, Interactive Environment – The UNIX Shell*, in this document for information on this command.

OS/390 UNIX adds a security extension in the form of audit attributes for files or directories. Audit attributes determine whether or not accesses to the object are audited by the System Authorization Facility (SAF) interface. The attributes can be set to audit successful access attempts (**s**), audit failed access attempts (**f**), audit all accesses (**a**), or do not audit access (**-**). To allow for both user and system auditing functions, there are two sets of audit attributes to accommodate two categories—user-requested and auditor-requested.

Within each category of audit attributes, the audit controls are as follows:

Table A-5. AUDIT BITS (2.5.1.2 d)

AUDIT BITS		
AUDIT FLAG	ALPHA NOTATION	DESCRIPTION
Read	s/f/a/-	Audit attempts for *read* access
Write	s/f/a/-	Audit attempts for *write* access
Execute	s/f/a/-	Audit attempts for *execute* access

To maintain the audit attributes, the **chaudit** shell command is provided. Refer to *Section 2.5.1.3, Interactive Environment – The UNIX Shell,* in this document for information on this command.

2.5.1.3 Interactive Environment – The UNIX Shell

The OS/390 UNIX shell is a command processor that allows users to do the following:

- Invoke shell commands or utilities
- Write shell scripts using the shell programming language
- Run shell scripts and C-language programs in the foreground, in the background, or in batch

This section describes the security considerations for the OS/390 UNIX shell, including shell commands, shell access, interoperability between the shell and TSO/E, and built-in shell variables.

As with other interactive environments, there are certain commands available in the OS/390 shell that have security implications. Most of these commands impact data security by altering security attributes for a directory or file; others impact system operation and user privileges. The most important of these commands are as follows:

Table A-6. SECURITY IMPACT SHELL COMMANDS (2.5.1.3 a)

SECURITY IMPACT SHELL COMMANDS		
COMMAND	DESCRIPTION	USER RESTRICTIONS
at[2]	Allows a user to run a series of commands at a specified later time, under control of the cron daemon.	Can be used by the superuser or users listed in the /usr/lib/cron/at.allow file.
automount	Configures the automount facility that mounts file systems at time of access.	Can only be used by a superuser. Started from /etc/rc.
batch	Allows a user to run a series of commands at a later time when the system is not busy, under control of the cron daemon.	Same as at command.
chaudit	Changes the audit attributes of files or directories. Audit attributes determine whether accesses to a file are audited by SAF.	Can only be used by the file owner or a superuser for non-auditor-requested audit attributes.
chgrp	Changes the GID for the specified file or directory.	By default, can be used only by the file owner or a superuser. The file owner must be a member of the group the file or directory is being changed to.

[2] The at, batch, and crontab commands are used to manipulate the functions of the cron daemon. The default specified environment disables cron. The information is included here for the sake of completeness.

SECURITY IMPACT SHELL COMMANDS		
COMMAND	DESCRIPTION	USER RESTRICTIONS
chmod	Changes the file modes (permission bits) for the specified file or directory.	By default, can be used only by the file owner or a superuser.
chown	Changes the UID and optionally the GID for the specified file or directory.	By default, the UID can only be changed by a superuser. Changes to the GID follow the rules for the chgrp command.
chroot	Changes the root directory to that specified in the command.	Can only be used by a superuser or a user with access to the BPX.SUPERUSER resource.
crontab	Allows a user to schedule a series of commands to be run on a regular basis, under control of the cron daemon.	Can be used by the superuser or users listed in the /usr/lib/cron/cron.allow file.
extattr	Sets, resets, displays the extended attributes of executable files. Extended attributes include APF authorization, program control, and shared address space use.	Can only be used by the file owner or a superuser. The APF attribute requires access to the BPX.FILEATTR.APF resource. The program control attribute requires access to the BPX.FILEATTR.PROGCTL resource.
su su userid	Starts a new shell with the security attributes of the superuser or a different user. When a different user is specified, the MVS identity is changed and MVS data set access is changed to that of the new MVS user. When issued as superuser (i.e., UID(0)) and BPX.DAEMON is defined, userid is switched to the value in BPXPRMxx SUPERUSER.	Access to superuser status requires access to the BPX.SUPERUSER resource. Access to a different user requires that user’s password or access to the BPX.SRV.*userid* resource.
umask	Sets the file-creation permission mask. The mask specifies the default permissions that are not to be allowed when a file is created.	Not restricted

As indicated, security for each command depends on resource privileges that are accessible to the user. The default restrictions for these commands can change according to options available with the installed ACP. If CA SAF HFS security is enabled, commands that may have required superuser authority or access to UNIXPRIV class resources are controlled by BPX.CAHFS resources instead.

Access to the OS/390 shell is possible from multiple origins:

TSO/E OMVS command – TSO/E users can enter the OMVS command to access the shell via a 3270 terminal interface.

rlogin – Users from another system can use the rlogin command to access the shell via an asynchronous terminal interface. The use of rlogin access is not permitted.

telnet – Users from another system can use the telnet command to access the shell via an asynchronous terminal interface.

OS/390 Communication Server with an RS/6000 system – Users of terminals attached to serial ports on an RS/6000 that is connected to the host can log on directly via an asynchronous terminal interface.

While there are no implicit security implications to the access origin point, control of these facilities in their own environment may be desirable. There is a high degree of interoperability between MVS TSO/E and the OS/390 shell. The following capabilities are provided:

Data can be moved between MVS data sets and files in an OS/390 UNIX HFS file system.

Some TSO/E commands manipulate the HFS environment to perform tasks such as creating directories and mounting file systems.

TSO/E commands can be issued from the shell command line, from a shell script, or from a program.

MVS job control language (JCL) can include shell commands. The BPXBATCH utility provides this capability. For examples, refer to *The BPXBATCH Utility* in IBM's *OS/390 UNIX System Services User's Guide* document, and *Appendix C. Running Shell Scripts or Executable Files under MVS Environments* in IBM's *OS/390 UNIX System Services Command Reference*.

HFS files can be edited in TSO/E through ISPF/PDF or in the OS/390 shell through editors such as ed, sed, and vi.

Extensions to the REXX language allow REXX programs to access callable services in the TSO/E, batch, shell, or C program environments.

The primary security implication resulting from these capabilities is that file and command access is based on the value of the OS/390 userid and/or the OS/390 UNIX UID and GID that are in effect at the time of file access or command execution.

Behavior within the OS/390 shell can be altered by the values of data from built-in shell variables. Variables that have security implications are as follows:

Table A-7. SECURITY IMPACT SHELL VARIABLES (2.5.1.3 b)

SECURITY IMPACT SHELL VARIABLES		
VARIABLE	DESCRIPTION	IMPLICATION
HOME	The user's home directory set from values specified by the security system.	The user's home directory contains that user's personal files and scripts that establish any unique environment settings.
LOGNAME	The user's login name, set from values specified by the security system.	Child processes, by default, receive names based on LOGNAME.
SHELL	The full pathname of the shell program set from values specified by the security system.	An invalid shell program name would prevent system access. A compromised program could reduce system security.
PATH	The list of directories the system searches to find executable commands.	An improper sequence of directories could cause the wrong version of a program to be executed.
STEPLIB	For value = current: Currently, active TASKLIB, STEPLIB, or JOBLIB allocations are passed on. For value = none: No STEPLIB to be used in the search order. For value = *dsn1:dsn2:dsn3*: Use the specified, **cataloged**, user-accessible MVS load libraries. Default value = current.	Executables with the set-user-ID or set-group-ID bit set can only use STEPLIB data sets specified by the STEPLIBLIST parameter in BPXPRMxx.
_BPX_ACCT_DATA	The account data to be used for processes being created.	Could require additional access permissions if the use of account data is secured.

SECURITY IMPACT SHELL VARIABLES		
VARIABLE	DESCRIPTION	IMPLICATION
_BPX_JOBNAME	The MVS jobname to be used for processes being created.	Requires superuser authority or access to BPX.JOBNAME to be effective. Allows a user/process to start a child process that, by virtue of name, may have other security issues. ***NOTE***: When the _BPX_JOBNAME variable is not set, processes created by fork or spawn are assigned jobnames consisting of the userid followed by a number (1-9).
_BPX_USERID	The OS/390 user identity to be used for processes being created, effective only for users who have authority for the setuid() function.	Requires access to the BPX.DAEMON resource to be effective. Allows a user/process to start a child process using a different security context.

2.5.1.4 Background Processes – Daemons and Servers

OS/390 UNIX supports the execution of processes in the background. Daemons and servers are distinguished from other background processes by the duration of execution and the privileges used. OS/390 UNIX daemons and servers correspond in function to MVS started tasks.

NOTE: OS/390 UNIX supports two levels of security—UNIX and OS/390 UNIX. UNIX-level security exists where the userids for daemons and servers are defined with a UID of **0** (i.e., superuser status) and the BPX.DAEMON and BPX.SERVER security resources are **not** defined. OS/390 UNIX-level security exists where the BPX.DAEMON or BPX.SERVER security resources are defined. This level provides a higher degree of security. OS/390 UNIX-level security must be configured so that the enhanced security is available.

A daemon is a background process that operates continuously or periodically to provide a system service. Daemons may be started at system initialization or in response to some event. Daemons must be assigned a userid with a UID of **0** (i.e., superuser authority) and have the appropriate permission to the BPX.DAEMON security resource. A daemon can use the seteuid, setuid, setreuid, or spawn (with change in userid requested) service to execute work using the security context of a user.

A server is a background process that operates continuously or periodically to provide an application service required by a client. Servers are typically started when the service they provide is required. Servers must have the appropriate permission to the BPX.SERVER security resource. A server can use the pthread-security-np service to create task-level security environments. If the server processes user requests without the client (e.g., user) password, the server acts as a surrogate and must have the appropriate permission to the BPX.SRV.*userid* (where *userid* is the OS/390 userid) security resource.

The security setup requirements for daemons and servers are as follows:

The daemon or server must be assigned a userid. For daemons, the userid must be assigned a UID of 0.

The assigned userid must have the appropriate access to the BPX.DAEMON or BPX.SERVER security resource and to the BPX.SRV.*userid* resource(s) as required.

The ACP’s Program Control feature must be active.

All programs to be loaded into the address space must be marked as controlled programs (i.e., defined to Program Control). Programs in HFS files must have the program-controlled extended attribute bit set.

Daemons are usually started in scripts executed at system initialization. These scripts contain commands that set up the environment and start the daemon. The commands used to start commonly used OS/390 UNIX daemons include the following:

Table A-8. DAEMON COMMANDS (2.5.1.4)

DAEMON COMMANDS		
COMMAND	DESCRIPTION	STARTUP
cron	Runs commands scheduled through at, batch, and crontab at specified dates and times.	At system initialization
inetd	Provides Internet service management for a network.	At system initialization
lm	Starts the login monitor daemon that starts the login process for logins initiated by Outboard Communications Server (OCS).	At system initialization
rlogind	Validates remote login (rlogin) requests.	By inetd
uucico	Processes uucp and uux file transfer requests.	By other processes including cron, uucpd, uucp, and uux
uucpd	Invokes uucico for TCP/IP connections from remote uucp systems.	By inetd

DAEMON COMMANDS		
COMMAND	DESCRIPTION	STARTUP
uuxqt	Runs commands from remote systems.	By uucico or cron

Unless justified and documented to the IAO, all of the daemons on this list, except for the inetd daemon, must be disabled. This policy improves system security by reducing the number of common targets of system attacks.

2.5.1.5 Miscellaneous Considerations

This section discusses miscellaneous security considerations for the OS/390 UNIX environment. These considerations include the following:

- SMF options
- Account data validation – IEFUJI
- Run-Time Library Services (RTLS)

2.5.1.5.1 SMF Options

In the OS/390 environment, SMF data is collected to identify access to the system and to measure the use of resources. This data can be critical to auditors investigating security incidents. SMF data can also be created by authorized applications; this function is controlled to preserve system integrity. The OS/390 UNIX environment is not exempt from SMF data collection.

For processes under OS/390 UNIX, SMF record type 30 contains data on user identity, program name, and file system activity. SMF record type 92 provides information on the I/O activity of a user or application against a specific file. SMF record types 30 and 92 must be recorded. Due to the potential for very high volumes, subtypes 10 and 11 of the type 92 record may be suppressed at the site's discretion. Refer to *Section 2, OS/390 Integrity*, in this document and to IBM's *OS/390 MVS System Management Facilities (SMF)* documentation for details and descriptions for these records.

SMF record types 34 and 35 are used to record TSO/E activity, but are also written by default when a new address space is created for a fork or spawn in the OS/390 UNIX environment. To eliminate errors in TSO/E accounting, IBM recommends that SYS1.PARMLIB(SMFPRMxx) be updated to suppress those records for OS/390 UNIX processes (e.g., the OMVS subsystem). Therefore, SMF record types 34 and 35 for OS/390 UNIX processes may be suppressed at the site's discretion.

User applications and non-IBM products that run under OS/390 UNIX can generate SMF records or check if SMF records are being generated. This is done by using the smf_record callable service. To be able to do this, an application must be running under a userid that has access to the BPX.SMF security resource. When the application or product is installed, the ACP must be updated to allow the access.

2.5.1.5.2 Account Data Validation – IEFUJI

IEFUJI is an OS/390 exit that validates job names and/or accounting information. If IEFUJI is being used, there are special considerations for OS/390 UNIX:

- OMVS should be defined as a subsystem in SYS1.PARMLIB(IEFSSNxx).

- IEFUJI should be set as an exit for subsystem OMVS in SYS1.PARMLIB(SMFPRMxx).

- The IEFUJI code should be adapted to exclude the names of some jobs and daemons started from /etc/rc.

- Refer to IBM's *OS/390 UNIX System Services Planning* document for details.

The use of IEFUJI has security implications when ACP rules are in use to validate job names or accounting data. The correct function of IEFUJI and the appropriate ACP access rules must be verified to ensure proper system operation and security.

2.5.1.5.3 RTLS

Members of IBM's Language Environment (LE) run-time library are used by OS/390 UNIX components (including the shell and utilities) and optionally by user applications running in the OS/390 UNIX environment. Access to the LE members can be made available through the system link list (LNKLSTxx) and LPA list (LPALSTxx), through STEPLIBs, or through an OS/390 feature known as Run-Time Library Services (RTLS).

If RTLS is used for OS/390 UNIX, the following three steps must be completed:

- The RUNOPTS parameter must be coded in SYS1.PARMLIB(BPXPRMxx).

- The RTLS feature must be configured in SYS1.PARMLIB(CSVRTLxx).

- Security resource profiles must be defined to the ACP:

  CSVRTLS.LIBRARY.*library.version* for each logical RTLS library to enable security checking,

  OR,

  CSVRTLS.NOSECCONNECT.*library.version* for each logical RTLS library to disable checking

OR,

CSVRTLS.NOSECCONNECT.* to disable all RTLS security checking.

If the other methods of access (i.e., link list or STEPLIB) to the LE members are used, the CSVRTLS profiles are not needed.

2.5.2 Security Controls

In contrast to the previous section of this document that discussed considerations, this section discusses the specific controls that are used to implement security policy for the OS/390 UNIX environment. These controls include the following:

SYS1.PARMLIB Requirements – The members and parameters that are coded in the SYS1.PARMLIB data set or concatenated Parmlib data sets.

/etc Requirements – The files and parameters that are coded in the **/etc** directory

Resource Profiles – The security resource profiles that are entered to the ACP to control OS/390 UNIX resources

OS/390 UNIX MVS Data Sets – The MVS data sets requiring control

OS/390 UNIX HFS Directories and Files – The HFS directories and files requiring control

Users and Groups – Guidelines for defining users and groups

OS/390 UNIX Started Tasks – The MVS started tasks requiring control

OS/390 UNIX Daemons and Servers – The UNIX daemons and servers requiring control

Operator Commands – The MVS operator commands requiring control

Sensitive TSO/E and Shell Commands and Environment Variable Settings – The TSO/E and UNIX shell commands and environment variables requiring control

2.5.2.1 SYS1.PARMLIB Requirements

Several members of the SYS1.PARMLIB data set are involved in controlling the configuration of OS/390 UNIX. In this section, BPXPRMxx, the primary member that configures OS/390, and other related members are discussed.

NOTE: These discussions do not cover all the details, only the controls that impact security are discussed. Please refer to other sections of this document and to IBM’s *OS/390 MVS Initialization and Tuning Reference* document for detailed information. The name SYS1.PARMLIB is used throughout this discussion, but the requirements apply to the members of all the data sets specified by the PARMLIB statement in the LOAD*xx* member of SYSn.IPLPARM or SYS1.PARMLIB.

2.5.2.1.1 SYS1.PARMLIB – BPXPRMxx

BPXPRMxx is the SYS1.PARMLIB member that contains the parameters that control the OS/390 UNIX environment. BPXPRMxx controls the way features work and it establishes logical access to data by configuring the HFS environment. This section discusses the controls in BPXPRMxx that impact security for the OS/390 UNIX environment.

SUPERUSER
The SUPERUSER parameter specifies the userid to be assigned to users when the **su** command is entered without a userid operand. The userid must be defined to the ACP as BPXROOT and have a UID of **0**.

TTYGROUP
The TTYGROUP parameter specifies the group name assigned to pseudo terminals (PTYs) and remote terminals (RTYs). The group must be defined to the ACP with a unique GID and users must not be assigned to this group. This group name is used by some shell commands (e.g., talk and write) when writing to the PTY or RTY being used by another user. The name TTY must be used.

STEPLIBLIST
The STEPLIBLIST parameter specifies the pathname of the HFS file that contains the list of MVS data sets that are used as step libraries for programs that have the set-user-id or set-group-id permission bit set. The use of STEPLIBLIST is at the site's discretion, but if used the value of STEPLIBLIST will be /etc/steplib. All *update* and *alter* access to the MVS data sets in the list will be logged and only systems programming personnel will be authorized to update the data sets. All products, applications, or locally-developed programs that reside in the MVS data sets in the list will either have a VIS on file from the vendor or have written approval from DISA FSO. Refer to *Section 2.1.2, Software Integrity*, for more information on software integrity guidance.

- *(ZUSS0033: CAT II) The IAO will ensure that update and allocate access to libraries residing in the /etc/steplib is limited to system programmers only, unless a letter justifying access is filed with the IAO, all update and allocate access is logged.*

USERIDALIASTABLE
The USERIDALIASTABLE parameter specifies the pathname of the HFS file that contains a list of userids and group names with their corresponding alias names. The alias table is intended primarily for use where mixed or lower case userids are used in the UNIX environment. Because the OS/390 MVS components support only upper case userids, the USERIDALIASTABLE will not be used.

FILESYSTYPE
The FILESYSTYPE parameter specifies the type of file system to be started. There may be multiple FILESYSTYPE parameters and each can have a number of sub-parameters. A

FILESYSTYPE with a TYPE(AUTOMNT) sub-parameter specifies that automount processing is active. If automount is active for user file systems, this alters the procedures used during initial userid setup. No mount point needs to be created for the file system, but the /etc/auto.master file and the associated MapName file(s) must be created.

ROOT
The ROOT parameter specifies data for the file system that is to be mounted as the root file system for OS/390 UNIX. ROOT can have a number of sub-parameters; the FILESYSTEM and SETUID|NOSETUID sub-parameters have security considerations. FILESYSTEM can be used to specify the name of the MVS HFS data set that holds the root file system. As the highest point in the HFS hierarchy, this file system is critical to system operations. Therefore appropriate ACP access rules must be written to protect the named data set. *Update* and *alter* access must be restricted to the OS/390 UNIX kernel and individual systems programming personnel. The SETUID|NOSETUID sub-parameter specifies whether or not the set-user-ID or set-group-ID permission bits are supported. SETUID|NOSETUID also impacts the APF-authorized and program-controlled extended attributes. For the root file system, SETUID must be specified for normal operations.

- *(ZUSS0031: CAT II) The Systems Programmer and IAO will ensure that data set rules for the data sets referenced in the ROOT and MOUNT statements in BPXPRMxx member in PARMLIB are properly restricted and access is restricted to appropriate system tasks or systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

MOUNT
The MOUNT parameter specifies data for a file system that is to be mounted by OS/390 UNIX. There are usually multiple MOUNT statements and each can have a number of sub-parameters. The FILESYSTEM, SETUID|NOSETUID, and SECURITY|NOSECURITY sub-parameters have significant security considerations. FILESYSTEM can be used to specify the name of the MVS HFS data set that holds the logical file system. Appropriate ACP access rules must be written to protect the named data set. *Update* and *alter* access must be restricted to the OS/390 UNIX kernel and to individual systems programming personnel. The SETUID|NOSETUID sub-parameter specifies whether or not the set-user-ID or set-group-ID permission bits are supported. SETUID|NOSETUID also impacts the APF-authorized and program-controlled extended attributes. SETUID may be specified for those file systems that contain only vendor-provided software or that have been documented to the IAO as requiring this support. Otherwise NOSETUID must be specified. The SECURITY|NOSECURITY sub-parameter specifies whether security checks are performed. SECURITY must be specified unless a specific exception for the file system has been identified and documented to the IAO. Regardless of IBM defaults, the values for SETUID|NOSETUID and SECURITY|NOSECURITY must be explicitly coded to protect against potential vendor changes and to simplify security reviews.

- *(ZUSS0031: CAT II) The Systems Programmer and IAO will ensure that data set rules for the data sets referenced in the ROOT and MOUNT statements in BPXPRMxx member in PARMLIB are properly restricted and access is restricted to appropriate system tasks or* *systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

STARTUP_PROC
The STARTUP_PROC parameter specifies the name of the JCL procedure (PROC) that starts the OS/390 UNIX component. This started task must be defined to the ACP. The name OMVS must be used.

RUNOPTS
The RUNOPTS parameter specifies the runtime options string for programs that use the Language Environment (LE) library when accessed via RTLS (rather than through link list or STEPLIB). If used, the RUNOPTS parameter should be coded with RTLS(ON). RTLS requires the definition of security resources to the ACP. Refer to *Section 2.5.2.3.2, FACILITY Class Resources for RTLS*, in this document for details.

- *(ZUSS0012: CAT II) The Systems Programmer will ensure parmlib member BPXPRMxx follows the speciations specified for the above control parameters SUPERUSER, STEPLIBLIST, USERIDALIASTABLE, STARTUP_PROC, and MOUNT.*

2.5.2.1.2 SYS1.PARMLIB – Other Members

While SYS1.PARMLIB(BPXPRMxx) contains most of the controls for OS/390 UNIX, there are some other PARMLIB members that must be reviewed. The following table lists the members and their parameters that have security considerations:

Table A-9. OTHER PARMLIB MEMBERS (2.5.2.1.2)

OTHER PARMLIB MEMBERS	
MEMBER NAME	SECURITY CONSIDERATIONS
CSVRTLxx	If RTLS is to be used, the LE libraries must be defined in CSVRTL00, and profiles must be defined to the ACP for CSVRTLS resources.
IEASYSxx	The OMVS=xx parameter must be specified so that BPXPRMxx is used. If xx is not specified, OS/390 UNIX runs in minimum configuration mode and undesirable parameter defaults will cause some system components not to operate. If RTLS is to be used, the RTLS=xx parameter must be specified to indicate the properly modified CSVRTLxx member.
IEFSSNxx	The SUBSYS SUBNAME(OMVS) parameter must be coded if IEFUJI is being used according to the specification in SMFPRMxx.
SMFPRMxx	The SYS or SUBSYS parameter must be coded so that record types 30 and 92 are collected. Subtypes 10 and 11 of type 92 may be suppressed at the site's discretion. The SUBSYS(OMVS,NOTYPE(34,35)) parameter is recommended so that these records for OS/390 UNIX do not cause errors in TSO/E record accounting. The SUBSYS(OMVS,EXITS(IEFUJI)) parameter must be coded if checking of job names or accounting data for OS/390 UNIX tasks is desired.

NOTE: Only those parameters related to security considerations for OS/390 UNIX are listed. Please refer to other sections of this document and to IBM's *OS/390 MVS Initialization and Tuning Reference* document for detailed information.

- *(ZUSS0011: CAT II) The Systems Programmer will ensure parmlib member IEASYSxx specifies parameter OMVS and does not specify OMVS=DEFAULT.*

2.5.2.2 /etc Requirements

Within UNIX environments, the /etc directory is the conventional location for files that specify locally customized executive software controls. In this section the **/etc** files that have considerations for OS/390 UNIX security are discussed. Please refer to the appropriate IBM documentation, including *OS/390 UNIX System Services Planning* and *OS/390 SecureWay Communications Server IP Configuration*, for detailed information.

/etc/auto.master and /etc/*mapname*

The /etc/auto.master file is used by the automount facility. It contains the directory or directories that are managed by automount. For each directory that is specified in /etc/auto.master, a MapName (/etc/*mapname*) file is listed. A MapName file contains the mount parameters to be used when the directory is mounted. The *setuid* and *security* parameters have security considerations. The *setuid* parameter specifies whether or not the set-user-ID or set-group-ID permission bits are supported. The setuid yes setting can be specified for those file systems that contain only vendor-provided software or that have been documented to the IAO as requiring this support. Otherwise setuid no must be specified. The *security* parameter specifies whether security checks are performed. The security yes setting must be specified unless a specific exception for the file system has been identified and documented to the IAO. Regardless of IBM defaults, the values for *setuid* and *security* must be explicitly coded to protect against potential vendor changes and to simplify security reviews.

- *(ZUSS0013: CAT II) The Systems Programmer will ensure that if the /etc/auto.master HFS FILE is used that each /etc/mapname file listed specifies setuid no and security yes, unless a letter justifying a specific exception is filed with the IAO.*

/etc/inetd.conf

The /etc/inetd.conf file is used by the inetd daemon. It specifies how inetd is to handle service requests on network sockets. Specifically, there is one entry in inetd.conf for each service. Each service entry specifies several parameters. The *login_name* parameter is of special interest. It specifies the userid under which the forked daemon is to execute. This userid is defined to the ACP and it may require a UID(0) (i.e., superuser authority) value.

The /etc/inetd.conf file must be reviewed carefully to determine if every entry in the file represents a service that is actually in use. Services that are not in use must be disabled to reduce potential security exposures. The following services must be disabled in /etc/inetd.conf unless justified and documented with the IAO:

RESTRICTED NETWORK SERVICES							
Service	Port	Service	Port	Service	Port	Service	Port
Chargen	19	finger	79	shell	514	time	37
Daytime	13	login	513	smtp	25	timed	525
Discard	9	nameserver	42	systat	11	uucp	540
Echo	7	netstat	15	talk	517		
Exec	512	qotd	17	tftp	69		

- *(ZUSS0014: CAT II) The Systems Programmer or IAO will ensure that the restricted network services specified in the /etc/inetd.conf file listed in the above table are disabled, unless a letter justifying the use of the restricted network service is on file with the IAO.*

/etc/profile
The /etc/profile file is the system-wide profile that is executed for each user’s shell invocation. It provides a default environment for users. It sets environment variables and executes commands. Although there are several variables and commands that can be included, those with notable security considerations are the STEPLIB variable and the umask command. The STEPLIB variable should be assigned a value of none in /etc/profile unless a specific requirement for another value exists. The use of STEPLIB must be coordinated with the SYS1.PARMLIB(BPXPRMxx) STEPLIBLIST control, the /etc/steplib file, and the use of RTLS. The umask command must be executed in /etc/profile with a value of 077. This sets the file-creation permission-code mask so that a file creator has full permissions, group members have no permission, and other users have no permission. Exceptions to this may occur during software installation when the installation process demands a more permissive value, during database access by users, and during administrative actions. All requirements will be justified and documented with the IAO. Refer to *Section 2.5.1.3, Interactive Environment – The UNIX Shell*, for additional details on variables and commands.

- *(ZUSS0015: CAT II) The Systems Programmer will ensure that the umask command is executed with a value of 077 and the LOGNAME variable is marked read-only for the /etc/profile file, exceptions are documented with the IAO.*

/etc/rc
The /etc/rc file is the system initialization shell script. When OS/390 UNIX kernel services start, /etc/rc is executed to set file permissions and ownership for dynamic system files and to perform other system startup functions such as starting daemons. There can be many commands in /etc/rc. There are two specific guidelines that must be followed:

Any chmod or chaudit command must not result in less restrictive security than what is specified in *Section 2.5.2.5, OS/390 UNIX HFS Directories and Files*, later in this document.

- *(ZUSS0016: CAT II) The Systems Programmer will ensure that any chmod or chaudit command specified in the /etc/rc file does not result in less restrictive security than what is specified in Section 2.5.2.5, OS/390 UNIX HFS Directories and Files, later in this document.*

Immediately prior to each command that starts a daemon, the _BPX_JOBNAME variable must be set to match the daemon’s name (e.g., inetd, syslogd). The use of _BPX_USERID is at the site’s discretion, but is recommended.

- *(ZUSS0016: CAT II) The Systems Programmer will ensure that the_BPX_JOBNAME variable is set to match the daemon’s name (e.g., inetd, syslogd)*

/etc/steplib
The /etc/steplib file is specified by the STEPLIBLIST control in SYS1.PARMLIB(BPXPRMxx) to contain the list of MVS data sets to be used as step libraries for programs that have the set-user-id or set-group-id bit set. All *update* and *alter* access to the MVS data sets in the list will be logged and only systems programming personnel will be authorized to update the data sets. All products, applications, or locally-developed programs that reside in the MVS data sets in the list will either have a VIS on file from the vendor or have written approval from DISA FSO. Refer to *Section 2.1.2, Software Integrity*, for more information on software integrity guidance. The data sets must be coordinated with the value of STEPLIB if coded in /etc/profile.

- *(ZUSS0033: CAT II) The IAO will ensure that update and allocate access to libraries residing in the /etc/steplib is limited to system programmers only, unless a letter justifying access is filed with the IAO, all update and allocate access is logged.*

/etc/tablename
The /etc/tablename file is specified by the USERIDALIASTABLE control in SYS1.PARMLIB(BPXPRMxx) to contain the list of OS/390 userids and group names with their corresponding alias names. The alias table is intended primarily for use where mixed or lower case userids are used in the UNIX environment. Because the OS/390 MVS components support only upper case userids, /etc/tablename should not be used.

2.5.2.3 Resource Profiles

The implementation of security in OS/390 UNIX is enabled largely through a number of resource profiles that are defined to the ACP. As a result of the structure of OS/390 UNIX and the different ACPs, the resource profiles fall into the following three general groups:

- Profiles implemented by IBM and supported by all the ACPs
- Profiles implemented by IBM for use under IBM's Security Server (a superset of the RACF ACP)
- Profiles implemented by Computer Associates for use under ACF2 and/or TOP SECRET

The profiles from all three groups are discussed in the following sections.

Through the release history of OS/390, IBM has continued to add resource definitions for OS/390 UNIX. A major focus of the newer resources is adding granularity to the control of various privileges. Where the need for one privilege previously required assigning superuser status (i.e., UID(0)) that carried virtually unlimited authority, there are now individual resources that can be assigned instead. There have also been improvements targeted at better file access performance and easier system administration.

IBM has provided new security resources that support added functionality and performance enhancements for the RACF components of the OS/390 Security Server. These resources provide default values for user parameters and a mapping function for UID and userid look-ups.

In its ACF2 and TOP SECRET ACPs, CA has extended security capabilities by providing the CA SAF HFS facility. This facility allows OS/390 UNIX file access to be controlled by ACP resource rules instead of the conventional file permission bits. CA SAF HFS also defines resources for system functions and file functions to provide more granularity than the conventional superuser authority.

The following sections discuss all of the resources specific to OS/390 UNIX. These are grouped as follows:

FACILITY Class BPX Resources for General OS/390 UNIX Functions
FACILITY Class Resources for RTLS
SURROGAT Class BPX Resources
FACILITY Class Resources for CA SAF HFS (ACF2/TOP SECRET)
FACILITY Class Resources for Default User Values in IBM Security Server (RACF)
UNIXPRIV Class Resources

2.5.2.3.1 FACILITY Class BPX Resources for General OS/390 UNIX Functions

There are a number of resources available under OS/390 UNIX that must be secured in order to preserve system integrity while allowing effective application and user access. All of these resources might not be used in every configuration, but several of them have critical impacts.

The following table shows the FACILITY class resource names for OS/390 UNIX with a description of their usage:

Table A-10. GENERAL FACILITY CLASS BPX RESOURCS (2.5.2.3.1)

GENERAL FACILITY CLASS BPX RESOURCES	
RESOURCE NAME	DESCRIPTION/NOTES
BPX.DAEMON	Allows a daemon to use the seteuid, setuid, setreuid, and spawn services.
BPX.DEBUG	Allows a user to use ptrace (via dbx) to debug programs that run with APF authority or with BPX.SERVER authority.
BPX.FILEATTR.APF	Allows a user to set the APF-authorized attribute in an HFS file.
BPX.FILEATTR.PROGCTL	Allows a user to set the program-controlled attribute in a HFS file. This attribute is required, in most cases, for all programs executed by daemons or servers.
BPX.JOBNAME	Allows a user to set jobnames using the _BPX_JOBNAME environment variable or the inheritance structure on spawn.
BPX.SAFFASTPATH	Enables SAF fastpath support. This means that successful security checks are not audited. No access list is needed; the existence of the profile enables the function.

GENERAL FACILITY CLASS BPX RESOURCES	
RESOURCE NAME	DESCRIPTION/NOTES
BPX.SERVER	READ: Allows the server to establish a thread-level security environment for its clients. Access control decisions are based on the server's userid and the client's userid unless the server specifies a password on the service invocation. UPDATE: Allows the server to establish a thread-level security environment for its clients. Access control decisions are based only on the client's userid. The pthread_security_np (create/delete security environment) and the auth_check_resource_np (resource authorization checking) services are used. Also see the BPX.SRV.*userid* profile description.
BPX.SMF	Allows a user to write an SMF record or test if an SMF type or subtype is being recorded.
BPX.STOR.SWAP	Allows a user to make address spaces non-swappable or swappable.
BPX.SUPERUSER	Allows a user to switch to superuser authority (i.e., effective UID of **0**).
BPX.WLMSERVER	Allows a user to access Work Load Manager (WLM) server functions and C language WLM interfaces. These functions and interfaces are commonly used by server applications. Also see the BPX.SERVER profile description.

The default access for each of these resources must be no access. Access can be permitted only to users with a requirement for the resource that is documented to the IAO. A generic resource (e.g., BPX.*) must also be set to a default access of none to cover future additions. Because they convey especially powerful privileges, the settings for BPX.DAEMON, BPX.SAFFASTPATH, BPX.SERVER, and BPX.SUPERUSER require special attention.

- *(ZUSS0021: CAT II) The Systems Programmer and IAO will ensure that BPX. Resources are properly protected and access is restricted to appropriate system tasks or systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

Access to BPX.DAEMON must be restricted to the OS/390 UNIX kernel userid, OS/390 UNIX daemons (e.g., inetd, syslogd, ftpd), and other system software daemons (e.g., web servers).

- *(ZUSS0021: CAT II) The Systems Programmer and IAO will ensure that BPX. DAEMON resources are properly protected and access is restricted to appropriate system tasks or systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

As noted above, the BPX.SAFFASTPATH definition can cause successful security checks not to be audited. Because auditing of all accesses is required for some system files, BPX.SAFFASTPATH must not be used. Additionally for ACF2 and TOP SECRET environments, when BPX.SAFFASTPATH is defined, calls to the ACP are not performed for file accesses and there is no audit trail of access failures. This configuration is unacceptable. Therefore BPX.SAFFASTPATH must not be used on any system.

- *(ZUSS0021: CAT II) The Systems Programmer will ensure that BPX.SAFFASTPATH resource is not specified.*

Access to BPX.SERVER must be restricted to system software processes that act as servers under OS/390 UNIX (e.g., web servers).

- *(ZUSS0021: CAT II) The Systems Programmer and IAO will ensure that BPX. SERVER resources are properly protected and access is restricted to appropriate system tasks, unless a letter justifying additional access is filed.*

Access to BPX.SUPERUSER must be restricted to Security Administrators and individual systems programming personnel. It is not appropriate for all systems programming personnel, only for those with responsibilities for components or products that use OS/390 UNIX and that require superuser capability for maintenance.

- *(ZUSS0021: CAT II) The Systems Programmer and IAO will ensure that BPX. SUPERUSER is properly protected and access is restricted to Security Administrators and appropriate systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

2.5.2.3.2 FACILITY Class Resources for RTLS

Run-Time library services (RTLS) can be used in place of link list entries or STEPLIBs that might otherwise be required for OS/390 UNIX to access the appropriate version of IBM Language Environment (LE) run-time libraries. If RTLS is being used (see the BPXPRMxx SYS1.PARMLIB requirements), *read* access or access checking deactivation must also be established for each logical library.

The following table shows the FACILITY class resources that can be defined if RTLS is being used in conjunction with OS/390 UNIX:

Table A-11. RTLS FACILITY CLASS RESOURCES (2.5.2.3.2)

RTLS FACILITY CLASS RESOURCES	
RESOURCE NAME	DESCRIPTION/NOTES
CSVRTLS.LIBRARY.*library.version*	Allows the user access to the modules specified by *library.version.*
CSVRTLS.NOSECCONNECT.*library.version*	Deactivates security checking for access to the modules specified by *library.version*. This means that any user has access.
CSVRTLS.CONNECT	UPDATE: Overrides the limit of 32 connections by unauthorized callers in a single address space.

Please refer to IBM's *OS/390 MVS Initialization and Tuning Reference* document for detailed information on using RTLS.

- *(ZUSS0021: CAT II) The Systems Programmer and IAO will ensure that CSVRTLS is properly protected and access is restricted to appropriate systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

2.5.2.3.3 SURROGAT Class BPX Resources

SURROGAT class BPX resources are used in conjunction with server applications that are performing tasks on behalf of client users that may not supply an authenticator to the server. This can be the case when clients are otherwise validated or when the requested service is performed from userids representing groups.

The following table describes the format of SURROGAT class BPX resources:

Table A-12. SURROGAT CLASS BPX RESOURCES (2.5.2.3.3)

SURROGAT CLASS BPX RESOURCES	
RESOURCE NAME	DESCRIPTION/NOTES
BPX.SRV.*userid* – *userid* is the OS/390 userid of the user the server is acting as a surrogate of.	Allows the server application to act as a surrogate of a client. The server processes user requests without a password. Also see the BPX.SERVER profile description.

The default access for each BPX.SRV.*userid* resource must be no access. Access can be permitted only to system software processes that act as servers under OS/390 UNIX (e.g., web servers).

- *(ZUSS0022: CAT II) The Systems Programmer and IAO will ensure that BPX. SRV.userid resources are properly protected and access is restricted to appropriate system tasks or systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

2.5.2.3.4 FACILITY Class Resources for CA SAF HFS (ACF2/TOP SECRET)

BPX.CAHFS FACILITY class resources are unique to the ACF2 and the TOP SECRET ACP implementations. When CA SAF HFS Security is enabled, the BPX.CAHFS FACILITY class resources control security for two defined resource types—system resources and file resources.

NOTE: When enabled, CA SAF HFS takes precedence over IBM's superuser granularity support, and all UNIXPRIV class resources are ignored.

The following table describes resource names for the CA SAF HFS system functions:

Table A-13. CA SAF HFS SYSTEM FUNCTION RESOURCES (2.5.2.3.4 a)

CA SAF HFS SYSTEM FUNCTION RESOURCES	
RESOURCE NAME	DESCRIPTION/NOTES
BPX.CAHFS.CHANGE.PRIORITY	Allows the user to change the scheduling priority of a process, process group, or user.
BPX.CAHFS.SET.PRIORITY	Allows the user to set the scheduling priority of a process, process group, or user.
BPX.CAHFS.SET.RLIMIT	Allows the user to set the resource limit for the calling process.
BPX.CAHFS.MOUNT	Allows the user to mount file systems.
BPX.CAHFS.UNMOUNT	Allows the user to remove a virtual file system.
BPX.CAHFS.PTRACE	Allows the user to control and debug another process. Access is denied if the user attempts to debug a program running setuid or setgid.
BPX.CAHFS.CREATE.LINK	Allows the user to create a hard link to an existing file. The user must have *alter* permission for both the original and link file names.
BPX.CAHFS.CREATE.EXTERNAL.LINK	Allows the user to create an external link to an object outside of the file system, e.g., an MVS data set. An external link is a file that contains the name of an external object.
BPX.CAHFS.CREATE.SYMBOLIC.LINK	Allows the user to create a symbolic link to an existing file. The user must have *alter* permission for both the original and link file names.

The default access for each BPX.CAHFS FACILITY class system function resource must be no access. Access can be permitted only to users with a requirement for the resource that is documented to the IAO.

- *(ZUSS0021: CAT II) The IAO will ensure that BPX.CAHFS is properly protected and access is restricted to users with a requirement to the resource with a letter justifying access is filed with the IAO.*

The following tables describe the access rule keywords and resource names for the CA SAF HFS file functions. The SERVICE (ACF2) or ACCESS (TOP SECRET) keyword of the access rule impacts the extent of the permission as follows:

Table A-14. CA SAF HFS FILE RESOURCE KEYWORDS (2.5.2.3.4 b)

CA SAF HFS FILE RESOURCE KEYWORDS	
SERVICE/ACCESS	***RESULT***
ADD/ALL, ALTER	Allows the user to perform the function against all files.
DELETE/CONTROL	Allows the user to perform the function if the user also has DELETE/CONTROL access to the HFS file resource.
UPDATE/UPDATE	Allows the user to perform the function if the user also has DELETE/CONTROL access to the HFS file resource.
READ/READ	Allows the user to perform the function if the user also has DELETE/CONTROL access to the HFS file resource or if the user is the owner of the file as defined by CA SAF HFS security.
EXECUTE/EXEC	Allows the user to perform the function if the user also has DELETE/CONTROL access to the HFS file resource.
NONE/NONE	The user cannot perform the function.

Table A-15. CA SAF HFS FILE RESOURCES (2.5.2.3.4 c)

CA SAF HFS FILE RESOURCES	
RESOURCE NAME	***DESCRIPTION/NOTES***
BPX.CAHFS.CHANGE.FILE.ATTRIBUTES	Allows the user to change extended file attributes including APF authorization and program control.
BPX.CAHFS.CHANGE.FILE.AUDIT.FLAGS	Allows the user to change the user-audit flags in a file.
BPX.CAHFS.CHANGE.FILE.FORMAT	Allows the user to change the format of a file.
BPX.CAHFS.CHANGE.FILE.MODE	Allows the user to change any file mode information, including file permission settings, execution UID or GID indicators, and the sticky bit.
BPX.CAHFS.CHANGE.FILE.MODE.STICKY	Allows the user to change the sticky bit in the file mode information.

CA SAF HFS FILE RESOURCES	
RESOURCE NAME	***DESCRIPTION/NOTES***
BPX.CAHFS.CHANGE.FILE.MODE.EUID	Allows the user to set the execution-UID indicator in the file mode information.
BPX.CAHFS.CHANGE.FILE.MODE.EGID	Allows the user to set the execution-GID indicator in the file mode information.
BPX.CAHFS.CHANGE.FILE.OWNER	Allows the user to change the file owner UID setting.
BPX.CAHFS.CHANGE.FILE.GROUP	Allows the user to change the file owner GID setting.
BPX.CAHFS.CHANGE.FILE.TIME	Allows the user to change the last access or modification time.

NOTE: Actions taken for resource rules for the CA SAF HFS file resources are impacted by the resource rules for the HFS files.

The default access for each BPX.CAHFS FACILITY class file resource must be no access. Access can be permitted only to users with a requirement for the resource that is documented to the IAO.

- *(ZUSS0021: CAT II) The IAO will ensure that BPX.CAHFS is properly protected and access is restricted to users with a requirement to the resource with a letter justifying access is filed with the IAO.*

For the ACF2 implementation of CA SAF HFS security, BPX.CAHFS.SECURITY FACILITY class resources are defined to control the execution of the SAFHFMOD utility. This utility reports on and controls the status of CA SAF HFS security on ACF2 systems. The following table shows the resource names to be defined:

Table A-16. ACF2 SAF HFS SECURITY RESOURCES (2.5.2.3.4 d)

ACF2 SAF HFS SECURITY RESOURCES	
RESOURCE NAME	***DESCRIPTION/NOTES***
BPX.CAHFS.SECURITY.STATUS	Allows the user to execute the SAFHFMOD utility with the STATUS parameter.
BPX.CAHFS.SECURITY.ENABLE	Allows the user to execute the SAFHFMOD utility with the ENABLE parameter.
BPX.CAHFS.SECURITY.DISABLE	Allows the user to execute the SAFHFMOD utility with the DISABLE parameter.

The nature of the SAFHFMOD utility requires that these resources be strictly controlled. The BPX.CAHFS.SECURITY resources must only be accessible to systems programming personnel and Security Administrators.

- *(ZUSS0021: CAT II) The IAO will ensure that BPX.CAHFS.SECURITY is properly protected and access is restricted to Security Administrators and appropriate systems programming personnel, unless a letter justifying access is filed with the IAO.*

2.5.2.3.5 FACILITY Class Resources for Default User Values in IBM Security Server (RACF)

The BPX.DEFAULT.USER FACILITY class resource is unique to the IBM Security Server (RACF) ACP implementation. The BPX.DEFAULT.USER resource can be used to supply default assignments for userid or userid and group ID. The application data field of the BPX.DEFAULT.USER resource can contain a RACF userid (e.g., OMVSLTU) or RACF userid/RACF group ID (e.g., OMVSLTU/OMVSLTG) that has been defined with the security attributes to be used as defaults.

Table A-17. RACF DEFAULT USER RESOURCES (2.5.2.3.5)

RACF DEFAULT USER RESOURCES	
RESOURCE NAME	***DESCRIPTIO/NOTES***
BPX.DEFAULT.USER	A profile containing the userid or the userid/group ID of the default profiles to be used for a user without an OS/390 UNIX profile

This resource will only be permitted for FTP socket applications on non classified systems. When coding these options, be sure that the restrictions specified in *Section 2.5.6.2, Paragraph 7* of this STIG are followed.

- *(ZUSSR050: CAT II) The IAO will ensure that the BPX.DEFAULT.USER for the FACILITY resource class is only defined for FTP socket applications on non-classified systems.*

2.5.2.3.6 UNIXPRIV Class Profiles

UNIXPRIV class profiles are used to manage certain system privileges that are typically associated with OS/390 UNIX superuser authority. By defining UNIXPRIV class profiles, certain individual superuser privileges can be granted to users who do not have superuser authority. This reduces the security risks associated with assigning full superuser authority to users.

The following table shows the available UNIXPRIV resource names with a description of their usage:

Table A-18. UNIXPRIV CLASS RESOURCES (2.5.2.3.6)

UNIXPRIV CLASS RESOURCES	
RESOURCE NAME	***DESCRIPTION/NOTES***
CHOWN.UNRESTRICTED[3]	Allows all OS/390 UNIX users to transfer ownership for files they own to any UID or GID on the system. No access list is needed; the existence of the profile enables the function.
SUPERUSER.FILESYS	READ: Allows the user to read any HFS file and to read or search any HFS directory. UPDATE: Allows the user to write to any HFS file and includes *read* access. CONTROL: Allows user to write to any HFS directory and includes *update* access. *NOTE:* Allows access only to local HFS files, not to NFS files.
SUPERUSER.FILESYS.CHOWN	READ: Allows the user to change the ownership of any file.
SUPERUSER.FILESYS.MOUNT	READ: Allows the user to mount a file system with the nosetuid option and to unmount a file system mounted with the nosetuid option. UPDATE: Allows the user to mount a file system with the setuid option and to unmount a file system mounted with the setuid option.
SUPERUSER.FILESYS.QUIESCE	READ: Allows the user to quiesce and unquiesce a file system mounted with the nosetuid option. UPDATE: Allows the user to quiesce and unquiesce a file system mounted with the setuid option.
SUPERUSER.FILESYS.PFSCTL	READ: Allows the user to use the pfsctl() (physical file system control) callable service.
SUPERUSER.FILESYS.VREGISTER	READ: Allows a server to use the v_reg() callable service to register as a virtual file system (VFS) file server.
SUPERUSER.IPC.RMID	READ: Allows the user to issue the ipcrm command to release IPC (Interprocess Communication) resources.

[3] The CHOWN.UNRESTRICTED profile defeats a basic file ownership protection, and must not be defined unless justified and documented to the IAO.

UNIXPRIV CLASS RESOURCES	
RESOURCE NAME	***DESCRIPTION/NOTES***
SUPERUSER.PROCESS.GETPSENT	READ: Allows the user to use the w_getpsent callable service to receive process status data for any process.
SUPERUSER.PROCESS.KILL	READ: Allows the user to use the kill() callable service to send signals to any process.
SUPERUSER.PROCESS.PTRACE	READ: Allows the user to use the ptrace() function through the dbx debugger to trace any process. Also allows users of the ps command to output information on all processes. *NOTE*: Authorization to FACILITY class resource BPX.DEBUG is required to trace processes that run with APF authority or BPX.SERVER authority.
SUPERUSER.SETPRIORITY	READ: Allows the user to increase that user's own priority.

- *(ZUSS0023: CAT II) The IAO will ensure that the CHOWN.UNRESTRICTED resource is not defined, unless a letter justifying access is filed with the IAO.*

- *(ZUSS0023: CAT II) The IAO will ensure that all SUPERUSER resources for the UNIXPRIV resource class are restricted to appropriate system tasks and/or system programming personnel, unless a letter justifying access is filed with the IAO.*

The default access for each UNIXPRIV class SUPERUSER resource must be **no** access. Access can be permitted only to users with a requirement for the resource that is documented to the IAO.

- *(ZUSS0023: CAT II) The IAO will ensure that all SUPERUSER resources for the UNIXPRIV resource class have default access of none.*

NOTE: When enabled, CA SAF HFS takes precedence over IBM's superuser granularity support and all UNIXPRIV class resources are ignored.

2.5.2.4 OS/390 UNIX MVS Data Sets

Security rules must be defined to prevent unauthorized changes to the OS/390 UNIX components in MVS data sets. Because OS/390 UNIX is integrated with the OS/390 base control program, many of the OS/390 UNIX components reside in data sets that are protected by security definitions specified elsewhere. The data set names (or masks) unique to OS/390 UNIX that may require additional definitions are listed in this section. Data sets in conventional MVS formats (e.g., PDS) and those in HFS format are listed. There is also a note on security for user MVS data sets in HFS format.

- *(ZUSS0031: CAT II) The Systems Programmer and IAO will ensure that data set rules for the data sets referenced in the ROOT and MOUNT statements in BPXPRMxx member in*

PARMLIB are properly restricted and access is restricted to appropriate system tasks or systems programming personnel, unless a letter justifying additional access is filed with the IAO.

2.5.2.4.1 MVS Data Sets for OS/390 UNIX Components

The following MVS format data sets are unique to OS/390 UNIX and may require additional security definitions:

Table A-19. MVS DATA SETS WITH OS/390 UNIX COMPONENTS (2.5.2.4.1)

MVS DATA SETS WITH OS/390 UNIX COMPONENTS		
DATA SET NAME/MASK	***MAINTENANCE TYPE***	***FUNCTION***
SYS1.ABPX*	Distribution	IBM OS/390 UNIX ISPF panels, messages, tables, clists
SYS1.AFOM*	Distribution	IBM OS/390 UNIX Application Services
SYS1.BPA.ABPA*	Distribution	IBM OS/390 UNIX Connection Scaling Process Mgr.
SYS1.CMX.ACMX*	Distribution	IBM OS/390 UNIX Connection Scaling Connection Mgr.
SYS1.SBPX*	Target	IBM OS/390 UNIX ISPF panels, messages, tables, clists
SYS1.SFOM*	Target	IBM OS/390 UNIX Application Services
SYS1.CMX.SCMX*	Target	IBM OS/390 UNIX Connection Scaling Connection Mgr.

The data sets designated as distribution data sets should have all access restricted to systems programming personnel. TSO/E users who also use OS/390 UNIX should have *read* access to the SYS1.SBPX* data sets. *Read* access for all users to the remaining target data sets is at the site's discretion. All other access must be restricted to systems programming personnel.

- *(ZUSS0032: CAT II) The Systems Programmer and IAO will ensure that data set rules for the data sets referenced in the above table are properly restricted and UPDATE and/or ALLOCATE/ALTER access is restricted to systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

2.5.2.4.2 MVS Data Sets Containing OS/390 Hierarchical File Systems

The following HFS format data sets are unique to OS/390 UNIX and require security definitions:

Table A-20. MVS DATA SETS CONTAINING HFS DATA (2.5.2.4.2)

MVS DATA SETS CONTAINING HFS DATA		
DATA SET NAME/MASK	***MAINTENANCE TYPE***	***FUNCTION***
SYS1.OE.ROOT	Target	Root File System – IBM OS/390 directories and files
SYS3.OE.ETCFILES	Target	Local site directories and files

These data sets should have all access restricted to systems programming personnel and to the OS/390 UNIX kernel userid OMVS. The site may choose different names for these data sets, but the access restrictions must be maintained.

There may be additional data sets that contain system HFS data. Any data set that specifies a file system that is at the root level (e.g., /tmp, /u) must also have all access restricted to systems programming personnel and to the OS/390 UNIX kernel userid.

- *(ZUSS0031: CAT II) The Systems Programmer and IAO will ensure that data set rules for the data sets referenced in the ROOT and MOUNT statements in BPXPRMxx member in PARMLIB are properly restricted and access is restricted to appropriate system tasks or systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

2.5.2.4.3 User MVS Data Sets Containing Hierarchical File Systems

Depending on the number of users defined in a given OS/390 UNIX image, there may be a need to define individual MVS data sets to hold their personal HFS format data. These data sets must be protected in accordance with the existing security guidelines for user data. However, there is a need for special additions to those rules. The OS/390 UNIX kernel userid OMVS must have update access to all user HFS data sets. Also, users must not have *update* access to the MVS data sets so that HFS permission controls cannot be altered outside of the OS/390 UNIX environment.

- *(ZUSS0031: CAT II) The Systems Programmer and IAO will ensure that data set rules for the data sets referenced in the ROOT and MOUNT statements in BPXPRMxx member in PARMLIB are properly restricted and access is restricted to appropriate system tasks or systems programming personnel, unless a letter justifying additional access is filed with the IAO.*

2.5.2.5 OS/390 UNIX HFS Directories and Files

Appropriate UNIX permission bit settings must be maintained to prevent unauthorized changes to the OS/390 UNIX components in HFS directories and files. Some directories and files must also have specific audit bit settings so that critical changes are logged. In this section, the settings that must be maintained on the OS/390 elements are listed. Settings for user HFS elements are also documented.

NOTE: As mentioned earlier in *Section 2.5.1.2, Data Storage – HFS Directories and Files*, the ACF2 and TOP SECRET ACPs offer the CA SAF HFS security option. When this option is enabled, OS/390 UNIX file permission bit security is bypassed. File access security is maintained through the use of HFS file resource rules. However, path names are truncated at 255 characters and translated before the rules are checked. The SAFHFUSR exit can be coded to modify this behavior and to facilitate de-centralized control for user data. Even when CA SAF HFS is enabled, the permission bit and audit bit settings documented in this section must be maintained. Doing so ensures security if CA SAF HFS is disabled or if certain directories or files are shared with other sites that are not using the option.

2.5.2.5.1 OS/390 System HFS Directories

There are a number of directories that must be secured to protect system functions in OS/390 UNIX. For directories not specified in this section or other product-specific sections throughout this STIG, a permission bit setting of 755 or 775 will be specified. The 775 setting may be used at the site’s discretion to help reduce the need for assignment of superuser privileges. The following table identifies permission bit and audit bit settings that are required for these specific directories. More restrictive permission settings may be used at the site’s discretion or as specific environments dictate.

Table A-21. SYSTEM DIRECTORY SECURITY SETTINGS (2.5.2.5.1)

SYSTEM DIRECTORY SECURITY SETTINGS			
DIRECTORY	***PERMISSION BITS***	***USER AUDIT BITS***	***FUNCTION***
/ [root]	755	faf	Root level of all file systems. Holds critical mount points.
/bin	755	fff	Shell scripts and executables for basic functions
/dev	755	fff	Character-special files used when logging into the OMVS shell and during C language program compilation. Files are created during system IPL and on a per-demand basis.
/etc	755	faf	Configuration programs and files (usually with locally customized data) used by OS/390 UNIX and other product initialization processes
/lib	755	fff	System libraries including dynamic link libraries and files for static linking
/samples	755	fff	Sample configuration and other files
/tmp	1777	fff	Temporary data used by daemons, servers, and users. *NOTE*: /tmp must have the sticky bit on to restrict file renames and deletions.
/u	755	fff	Mount point for user home directories and optionally for third-party software and other local site files
/usr	755	fff	Shell scripts, executables, help (man) files and other data. Contains sub-directories (e.g., lpp) and mount points used by program products that may be in separate file systems.
/var	1775	fff	Dynamic data used internally by products and by elements and features of OS/390 UNIX ***NOTE***: /var must have the sticky bit on to restrict file renames and deletions.

In addition, the following guidelines must be followed:

All directories (such as /tmp) with the *write* permission set for the other group must also have the sticky bit set.

Any directory (such as /tmp) with the *write* permission set for the other group must not contain any files with the following bits set:

- set-user-ID permission
- set-group-ID permission
- APF-authorized extended attribute
- Program control extended attribute

- *(ZUSS0034: CAT II) The Systems Programmer will ensure that the HFS permission bits for each directory match or are more restrictive than the specified settings listed in the table A-21 entitled System Directory Security Settings in Section 2.5.2.5.1.*

- *(ZUSS0034: CAT II) The Systems Programmer will ensure that the HFS user audit bits for each directory match settings listed in the table A-21 entitled System Directory Security Settings in Section 2.5.2.5.1.*

- *(ZUSS0036: CAT II) The Systems Programmer will ensure that the HFS directory(ies) with the "other" write permission bit set is (are) not properly defined.*

2.5.2.5.2 OS/390 System HFS Files

There are a number of files that must be secured to protect system functions in OS/390 UNIX. Where not otherwise specified, these files must receive a permission setting of 744 or 774. The 774 setting may be used at the site's discretion to help to reduce the need for assignment of superuser privileges. The following table identifies permission bit and audit bit settings that are required for these specific files. More restrictive permission settings may be used at the site's discretion or as specific environments dictate.

Table A-22. SYSTEM FILE SECURITY SETTINGS (2.5.2.5.2)

SYSTEM FILE SECURITY SETTINGS			
FILE	***PERMISSION BITS***	***USER AUDIT BITS***	***FUNCTION***
/bin/sh	1755	Faf	OS/390 UNIX shell ***NOTE***: /bin/sh has the sticky bit on to improve performance.
/dev/console	740	Fff	The system console file receives messages that may require System Administrator (SA) attention.
/dev/null	666	Fff	A null file; data written to it is discarded.
/etc/auto.master and any *mapname* files	740	Faf	Configuration files for automount facility
/etc/inetd.conf	740	Faf	Configuration file for network services
/etc/init.options	740	Faf	Kernel initialization options file for OS/390 UNIX environment

SYSTEM FILE SECURITY SETTINGS			
FILE	***PERMISSION BITS***	***USER AUDIT BITS***	***FUNCTION***
/etc/log	744	Fff	Kernel initialization output file
/etc/profile	755	Faf	Environment setup script executed for each user
/etc/rc	744	Faf	Kernel initialization script for OS/390 UNIX environment
/etc/steplib	740	Faf	List of MVS data sets valid for set-user-ID and set-group-ID executables
/etc/tablename	740	Faf	List of OS/390 userids and group names with corresponding alias names
/usr/lib/cron/at.allow /usr/lib/cron/at.deny	700	Faf	Configuration files for the at and batch commands
/usr/lib/cron/cron.allow /usr/lib/cron/cron.deny	700	Faf	Configuration files for the crontab command

Some of the files listed above (e.g., /etc/steplib) are not used in every configuration. While the absence of a file is generally not a security issue, the existence of a file that has not been properly customized can often be an issue. Therefore, all directories and files that do exist must have the specified permission and audit bit settings.

- *(ZUSS0035: CAT II) The Systems Programmer will ensure that the HFS permission bits for each file match or are more restrictive than the specified settings listed in the table A-22 entitled System File Security Settings in Section 2.5.2.5.2.*

- *(ZUSS0035: CAT II) The Systems Programmer will ensure that the HFS user audit bits for each file match settings listed in the table A-22 entitled System File Security Settings in Section 2.5.2.5.2.*

2.5.2.5.3 Individual User HFS Directories and Files

When a user’s home directory and other setup files (such as .profile) are initially created by the IAO, their permissions must be set to 700. Permissions to the home directory will never be more permissive than 750 unless justified and documented to the IAO. With the required umask setting of 077 that is established by the execution of /etc/profile at user logon, most new files created by a user will automatically receive the 700 permission setting. This setting allows full[4] control to the file owner (i.e., logged on user), no access to the members of the user’s group, and no access to others.

[4] The at, batch, and crontab commands are used to manipulate the functions of the cron daemon. The default specified environment disables cron; the information is included here for the sake of completeness.

- *(ZUSS0015: CAT II) The Systems Programmer will ensure that the umask command is executed with a value of 077 for the /etc/profile file, exceptions are documented with the IAO.*

2.5.2.6 Users and Groups

A number of identification-related items must be defined to the ACP to enable OS/390 UNIX. This section discusses requirements for the following:

- Specific userids and groups that must be defined for MVS started tasks, UNIX daemons, and UNIX servers

- UID and GID information as referenced earlier in *Section 2.5.1.1*, *User Identity – UID and GID Assignment*.

- Specific parameters such as home directory and startup shell program that must be added to OS/390 users' userid definitions to support access to the OS/390 UNIX shell and the ability to run programs under OS/390 UNIX

While this section discusses requirements, the specific commands for those definitions are discussed in the ACP implementation sections later in this document.

- *(ZUSS0042: CAT II) The IAO will ensure each group has a unique GID number.*

2.5.2.6.1 Privileged Users and Special Groups

A userid that has access to certain special capabilities within the system is considered privileged. These capabilities include superuser status by being assigned UID(0) or access to the BPX.SUPERUSER profile or access to services permitted through the BPX.DAEMON or BPX.SERVER profiles.

The privileged userids shown in the following table must be defined for OS/390 UNIX:

Table A-23. PRIVILEGED USERIDS (2.5.2.6.1 a)

PRIVILEGED USERIDS			
ID	***UID***	***GROUP***	***FUNCTION***
OMVS	0	STCOMVS[5]	OS/390 UNIX kernel userid. Owning user for the OMVS and BPXOINIT address spaces.
BPXROOT	0	OMVSGRP	ID used by daemons that need to invoke setuid() for superusers. Specified in the SUPERUSER parameter in the BPXPRMxx member of **SYS1.PARMLIB**.

The OMVS userid must have access to the BPX.DAEMON resource because daemons are (ordinarily) started by the OS/390 UNIX kernel at system initialization. The BPXROOT account must **not** have access to BPX.DAEMON because it would allow su command users to bypass intended security for that resource. Neither OMVS nor BPXROOT should have access to TSO.

- *(ZUSS0043: CAT II) The Systems Programmer and IAO will ensure that the user account for the OS/390 UNIX kernel (OMVS) is properly defined to the security database.*

- *(ZUSS0044: CAT II) The Systems Programmer and IAO will ensure that BPXROOT user account is properly defined to the security database.*

Any OS/390 started task that requires the use of OS/390 UNIX must have a userid with valid UID and group values assigned. The following existed prior to OS/390 UNIX, but now requires OS/390 UNIX facilities to run:

RMFGAT – RMFGAT is the userid for the Resource Measurement Facility (RMF) Monitor III Gatherer. It requires access to OS/390 UNIX data. It must be assigned a unique UID and group and assigned the root directory (“/”) as its home directory.

- *(ZUSS0045: CAT II) The Systems Programmer and IAO will ensure that the RMFGAT user account is properly defined in the security database.*

[5] OMVSGRP is the name suggested by IBM for all the required userids. STCOMVS is the standard name used at some sites for the userids that are associated with OS/390 UNIX started tasks and daemons. These groups can be combined at the site’s discretion.

The groups listed in the following table must be defined or updated with group ID (GID) information:

Table A-24. PRIVILEGED GROUPS (2.5.2.6.1 b)

PRIVILEGED GROUPS		
GROUP	***GID***	***FUNCTION***
OMVSGRP	3	Owning group for the BPXROOT userid
STCOMVS	4	OS/390 UNIX kernel group ID. Owning group for the OMSVKERN/OMVS userid.
TTY	1	Assigned to pseudo terminals (PTYs) and remote terminals (RTYs) that are used by the talk, write, and mesg utilities. Specified in the TTYGROUP parameter in the BPXPRMxx member of SYS1.PARMLIB.
SYS1	0	IBM software group

- *(ZUSS0041: CAT II) The Systems Programmer will ensure that the OMVSGRP group and / or the STCOMVS group are each defined to the security database with a unique GID in the range of 1-99.*

In order to manage the OS/390 UNIX environment, it is necessary to assign UID 0 to the following types of user accounts:

- Maintenance accounts (e.g., accounts dedicated for SMP/E) for OS/390 UNIX or other components that have HFS-based elements

- Security Administrators who create or maintain userid definitions

- Userids associated with UNIX daemons

- There are two preferable alternatives that can enhance security compared to the assignment of UID 0:

- Assign users the BPX.SUPERUSER FACILITY class profile so that they can switch from unprivileged to superuser status when necessary.

- Assign users individual privileges via UNIXPRIV class profiles (or optionally CA SAF HFS FACILITY class profiles in ACF2 or TOP SECRET environments).

The following guidelines apply to the definitions of privileged userids and special groups:

- UID numbers for privileged userids associated with OS/390 UNIX tasks (MVS started tasks, UNIX daemons, and UNIX servers) must be between 1 and 99. Unless the exception is documented to the IAO, each userid with a non-zero UID must receive a unique UID.

- GID numbers for groups associated with OS/390 UNIX tasks (MVS started tasks, UNIX daemons, and UNIX servers) must be between 1 and 99.

- All userids that are assigned UID 0 must be documented to the IAO.

- *(ZUSS0046: CAT II) The Systems Programmer and IAO will ensure that UID(0) is assigned only to system tasks such as the OS/390 UNIX kernel (i.e., OMVS or OMVSKERN), OS/390 UNIX daemons (e.g., inetd, syslogd, ftpd), and other system software daemons; to security administrators who create or maintain user account definitions; and to systems programming accounts dedicated to maintenance (e.g., SMP/E) of HFS-based components.*

2.5.2.6.2 Unprivileged Users and Groups

The following recommendations apply to the definitions of unprivileged userids and ordinary groups:

(1) Each user must be assigned a UID that is unique within the ACP database. If NFS-based file sharing is used between systems, UIDs within the sharing group must be unique. Ideally, each user should have a UID that is unique.

(2) UID numbers for unprivileged userids should be between 100 and 16,777,215. (OS/390 allows values up to 2,147,483,647, but there are issues with the tar command.)

(3) Each group should be assigned a GID that is unique within the ACP database. If NFS-based file sharing is used between systems, GIDs within the sharing group must be unique. Ideally, each group should have a GID that is unique.

(4) GID numbers for unprivileged groups should be between 100 and 16,777,215. (OS/390 allows values up to 2,147,483,647, but there are issues with the tar command.)

(5) Each user of the OS/390 UNIX shell should be assigned that user’s own unique home directory. If this were not done, the default file permission setting 740 would have to be changed for correct operation of the shell and this would degrade security for those users.

(6) Each user of the OS/390 UNIX shell should be assigned the default shell /bin/sh or (optionally in OS/390, Version 2, Release 9 and beyond) /bin/tcsh. Exceptions to this must be justified and documented to the IAO.

(7) Users of non-shell OS/390 UNIX services, must be assigned a unique UID (as described above).At the discretion of the IAO, an exception to this rule is the use of FTP socket applications with the following restrictions.

  - Use of the OMVS default UID will not be allowed on any classified system.

  - The definition of the OMVS default user will be restricted to a non-0 UID, a non-writable home directory, such as "\" root, and a non-executable, but existing, binary file, "/bin/false" or “/bin/echo.”

- Application of the APAR PQ63326 to control FTP access to UNIX files is required.

- Collection of SMF type 80 records to track user access to OMVS default UID.

- *(ZUSS0048: CAT II) The Systems Programmer and IAO will ensure that the above bulleted options are enforced for FTP socket applications using shared OMVS segments.*

- *(ZUSS0047: CAT II) The Systems Programmer and IAO will ensure that each user account is defined with a unique UID number (except for UID(0) users), a unique HOME directory (except for UID(0) and other system task accounts), and shell program specified as "/bin/sh", "/bin/tcsh", or "/bin/false."*

2.5.2.7 OS/390 UNIX Started Tasks

The following MVS started tasks must be defined to the ACP to support OS/390 UNIX:

OMVS – The OMVS task initializes the OS/390 UNIX kernel. The name is specified in the STARTUP_PROC parameter in the BPXPRMxx member of SYS1.PARMLIB.

BPXOINIT – The BPXOINIT task runs the initialization process (as specified in /etc/rc), acts as the parent process of some OS/390 UNIX processes, and provides support for some kernel calls.

BPXAS – The BPXAS task is used to create a new address space when a program issues a fork or spawn.

BPXSTOP – The BPXSTOP task stops any USS tasks, inetd, dfscm, syslogd.

NOTE: RACF (OS/390 Security Server) does not require a started task definition for BPXAS, but ACF2 and TOP SECRET do require one.

2.5.2.8 OS/390 UNIX Daemons and Servers

As discussed in *Section 2.5.1.4, Background Processes – Daemons and Servers,* earlier in this document, there are facilities in OS/390 UNIX that are provided through processes called daemons. In addition, third-party software packages may require the definition of daemons or servers. This section provides specific guidance for daemons provided with OS/390 UNIX and general guidance for all daemons and servers.

The inetd daemon manages connections to some network services. It uses parameters defined in the /etc/inetd.conf file. The inetd daemon is started in the /etc/rc script and inherits the initialization userid OMVSKERN or OMVS. The _BPX_JOBNAME variable must be set to INETD or another unique value immediately before the inetd command in /etc/rc.

The cron, lm, rlogind, and uucp (uucico, uucpd, and uuxqt) daemons must not be enabled unless justified and documented to the IAO. This policy improves system security by reducing the number of common targets of system attacks. If these daemons are enabled, distinct userids for them are recommended.

The following general guidelines apply to the definition of daemons and servers:

(1) For daemons, access to the BPX.DAEMON FACILITY class profile should be defined for their userids in the ACP. In accordance with this, the OS/390 UNIX kernel userid (usually OMVSKERN or OMVS) must be assigned this access.

(2) For servers, appropriate access to the BPX.SERVER FACILITY class profile must be defined for their userids in the ACP. Access to specific SURROGAT class profiles may also be required.

(3) Daemon processes that issue the setuid() for superusers use the userid (usually BPXROOT) specified in the SUPERUSER parameter in SYS1.PARMLIB. This userid must not be given access to BPX.DAEMON. Refer to IBM's *OS/390 UNIX System Services Planning* document for details.

(4) Programs loaded into an address space that requires daemon or server authority must be defined to the ACP's facility for program control, and the extended attribute for program control must be set on the HFS file. If this is not done, the address space is marked *dirty* and daemon or server services are not available. Use the ACP-specific procedures to define the libraries and/or programs. The BPX.FILEATTR.PROGCTL FACILITY class profile may be assigned to users who need to use the extattr command to set the program-controlled extended attribute on the HFS files for a daemon or server.

2.5.2.9 Operator Commands

OS/390 UNIX is designed to be an operational element of OS/390 at all times. To support availability and allow operating flexibility, there are operator commands that permit execution parameters to be altered dynamically, individual threads and processes to be terminated, and partial and full shutdowns to be performed. To preserve system integrity, these commands must be secured. This section very briefly describes the relevant operator commands for OS/390 UNIX. Please refer to IBM's *OS/390 UNIX System Services Planning* and *OS/390 MVS System Commands* documents for details.

F BPXOINIT,TERM=*pid.tid* and F BPXOINIT,FORCE= *pid.tid*
The F BPXOINIT,TERM=*pid.tid* and F BPXOINIT,FORCE=*pid.tid* commands terminate an OS/390 UNIX process thread (*pid.tid*) or entire process (*pid*). The TERM option allows the process's signal interface routine to receive control before termination; the FORCE option does not.

F BPXOINIT,SHUTDOWN=FORKINIT
The F BPXOINIT,SHUTDOWN=FORKINIT command shuts down the OS/390 UNIX initiators in preparation for a complete shutdown of OS/390. A system IPL is required to resume normal processing.

F BPXOINIT,SHUTDOWN=FORKS and F BPXOINIT,RESTART=FORKS
The F BPXOINIT,SHUTDOWN=FORKS command terminates all forked and non-local spawned address spaces and prevents new ones from starting. This is done in preparation for JES2 maintenance. The F BPXOINIT,RESTART=FORKS command enables normal processing to resume.

SET OMVS=*xx*
The SET OMVS=*xx* command dynamically reconfigures OS/390 UNIX by specifying one or more new BPXPRM*xx* members of SYS1.PARMLIB to replace the one(s) currently in effect.

SETOMVS *xxx*=*yyy*
The SETOMVS *xxx*=*yyy* command dynamically reconfigures OS/390 UNIX by changing the value (*yyy*) of one or more of the individual parameters (*xxx*) that were originally set at IPL from the BPXPRMxx member of SYS1.PARMLIB.

- *(ACP00282: CAT II) The IAO will ensure that OS/390 Sensitive System Commands are defined to the OPERCMDS resource class. Only limited number of authorized people are able to issue these commands. All access is logged.*

The operator commands described in this section must be secured through definitions in the ACP. Refer to *Section 3.1.5.6, OS/390 System Command Controls*, of this document for specifications on securing MODIFY (F), SET OMVS, and SETOMVS.

2.5.2.10 Sensitive TSO/E and Shell Commands and Environment Variable Settings

The TSO/E and OS/390 shell environments provide interactive access to facilities of OS/390 UNIX. Some of these facilities include commands that can impact security definitions and operational elements. This section describes the security controls for sensitive commands in TSO/E and the OS/390 shell and guidance for environment variables in the OS/390 shell.

2.5.2.10.1 Sensitive User Commands – TSO/E Environment

This section discusses those sensitive TSO/E commands that have the capability to impact the security and operational status of the OS/390 UNIX environment.

ishell
The TSO/E ishell command is the *ISPF shell* for OS/390 UNIX. It provides a full screen interface within TSO/E for performing many tasks, especially those related to file systems and files. In addition to functions for unprivileged users, ishell provides access to a number of functions that System Administrators might perform. These functions include file system mounts and unmounts, changing user attributes, setting up OS/390 UNIX users and groups, and switching to and from superuser status. Access to sensitive functions in ishell requires superuser authority. There is no explicit action to secure these functions other than limiting access to superuser authority. Refer to IBM’s *OS/390 UNIX System Services User’s Guide* document for details on the ishell command.

mount and unmount

The TSO/E mount command makes a file system available to OS/390 UNIX users. The TSO/E unmount command performs the reverse function. The setuid|nosetuid and security|nosecurity parameters of the mount command have important security implications. Refer to the discussion of the MOUNT parameter in *Section 2.5.2.1.1*, *SYS1.PARMLIB – BPXPRMxx*, in this document for details on those parameters. Both mount and unmount can only be performed by users with UID 0, access to the BPX.SUPERUSER profile, access to the SUPERUSER.FILESYS.MOUNT profile, or (for CA SAF HFS environments) access to the BPX.CAHFS.MOUNT and/or BPX.CAHFS.UNMOUNT profiles.

2.5.2.10.2 Sensitive User Commands – UNIX Shell Environment

This section discusses those sensitive OS/390 shell commands have the capability to impact the security and operational status of the OS/390 UNIX environment. The functions of these commands were discussed in *Section 2.5.1.3, Interactive Environment – The UNIX Shell*, of this document. The following table indicates the security controls that are used to limit access to those sensitive shell commands:

Table A-25. SHELL COMMAND SECURITY CONTROLS (2.5.2.10.2)

SHELL COMMAND SECURITY CONTROLS	
COMMANDS	***SECURITY CONTROLS***
at[6], automount, batch, chaudit, chgrp, chmod, chown, chroot, crontab, extattr, su	UID 0 BPX.SUPERUSER profile
at, batch	/usr/lib/cron/at.allow file /usr/lib/cron/at.deny file
Chaudit	BPX.CAHFS.CHANGE.FILE.AUDIT.FLAGS profile
Chgrp	BPX.CAHFS.CHANGE.FILE.GROUP[7] profile
Chmod	BPX.CAHFS.CHANGE.FILE.MODE profile BPX.CAHFS.CHANGE.FILE.MODE.STICKY profile BPX.CAHFS.CHANGE.FILE.MODE.EUID profile BPX.CAHFS.CHANGE.FILE.MODE.EGID profile
Chown	CHOWN.UNRESTRICTED[8] profile SUPERUSER.FILESYS.CHOWN profile BPX.CAHFS.CHANGE.FILE.OWNER profile
crontab	/usr/lib/cron/cron.allow file /usr/lib/cron/cron.deny file

[6] The at, batch, and crontab commands are used to manipulate the functions of the cron daemon. The default specified environment disables cron; the information is included here for the sake of completeness.
[7] The BPX.CAHFS profiles are only available when enabled in ACF2 or TOP SECRET environments.
[8] The CHOWN.UNRESTRICTED profile defeats a basic file ownership protection and must not be defined unless justified and documented to the IAO.

SHELL COMMAND SECURITY CONTROLS	
COMMANDS	***SECURITY CONTROLS***
Extattr	BPX.FILEATTR.APF profile BPX.FILEATTR.PROGCTL profile BPX.CAHFS.CHANGE.FILE.ATTRIBUTES profile
Su	BPX.SUPERUSER for superuser userid BPX.SRV.*userid* profile for userid *userid* Password for target userid

- *(ZUSS0023: CAT II) The Systems Programmer and IAO will ensure that the CHOWN.UNRESTRICTED resource is not defined, unless a letter justifying access is filed with the IAO.*

2.5.2.10.3 Sensitive User Environment Variable Settings

As discussed in *Section 2.5.1.3, Interactive Environment – The UNIX Shell*, of this document, OS/390 UNIX supports variables to control the environment of each user. There are a number of these variables, but only a few have significant potential security impacts. This section provides guidance on how a Security Administrator sets values for those variables.

HOME
The HOME variable holds the fully qualified path to an individual user's home directory. The value of HOME is set from data in the ACP when the user logs on. A user's home directory usually contains files that are required for normal processing. Because the default value for HOME is the root directory (i.e., "/"), it is important that the ACP's userid definition for each user of OS/390 UNIX have a valid value and that each unprivileged userid has a unique value.

LOGNAME
The LOGNAME variable holds the userid. The value of LOGNAME is set from data in the ACP when the user logs in. Because the value is often used in scripts and by system components, LOGNAME will be set to read-only status in the /etc/profile script.

- *(ZUSS0015: CAT II) The Systems Programmer will ensure that the LOGNAME variable is marked read-only for the /etc/profile file, exceptions are documented with the IAO.*

SHELL
The SHELL variable holds the fully qualified path of the shell program. The value of SHELL is set from data in the ACP when the user logs on. The value of SHELL is used by various commands to invoke the shell. Although there is only one shell supported for Version 2, Release 8 of OS/390 UNIX, this changes in subsequent releases. It is important that there be a valid value in the ACP's userid definition for each OS/390 user who uses the OS/390 UNIX shell. For OS/390 UNIX users who do not use the shell, a non-working value (i.e., /bin/false) should be specified so that access to the shell is disabled.

PATH
The PATH variable holds the list of HFS directories that the system searches for an executable file. Because omissions can cause a program not to be found and errors can cause the wrong version of a program to be executed, the value of PATH should be explicitly set in the /etc/profile script. Using a dot (.) or null string in the list causes the working directory to be searched. This working directory function should not be used in the /etc/profile script and can only be specified as the last directory in the sequence for user PATH values.

STEPLIB
The STEPLIB variable holds the specification that identifies the MVS libraries to be searched for executables stored in MVS data sets. STEPLIB can have a value of none, current, or a list of MVS data sets. The use of STEPLIB can cause errors for set-user-ID or set-group-ID programs if not coordinated with the values of the STEPLIBLIST parameter of BPXPRMxx in SYS1.PARMLIB and the /etc/steplib file. Unless required for another function (such as use of the RTLS function), STEPLIB=none should be specified in the /etc/profile script.

_BPX_ACCT_DATA
The _BPX_ACCT_DATA variable holds account data to be used for a new process that is being started. There is no explicit security control for the _BPX_ACCT_DATA variable. However, if account code data is secured through ACP rules and checked by system exits, specification of invalid data in _BPX_ACCT_DATA could cause processes to fail with apparent security violations.

_BPX_JOBNAME
The _BPX_JOBNAME variable holds the name to be used for a new process that is being started. To use the _BPX_JOBNAME variable, the user should have superuser authority or access to the BPX.JOBNAME profile. Any scripts that start daemons (such as /etc/rc) should use _BPX_JOBNAME to ensure that the daemons can be easily identified for operational purposes.

_BPX_USERID
The _BPX_USERID variable holds a userid that is to be used for the security identity for a new process that is being started. To use the _BPX_USERID variable, the user should have superuser authority or access to the BPX.DAEMON profile. The use of _BPX_USERID in scripts that start daemons (such as /etc/rc) is recommended.

2.5.3 ACF2 Implementation for OS/390 UNIX

This section describes the commands needed to implement the security guidelines for OS/390 UNIX under the ACF2 ACP. The following task categories are described:

- Resource profiles and program control
- Users and groups
- Started tasks
- Data
- Sensitive commands and utilities

Please note that when the optional CA SAF HFS security is enabled, the use of the SAFHFUSR installation exit can significantly alter the behavior of ACF2 with regards to security decisions for OS/390 UNIX. The use of this exit should conform to the procedures documented in *Section 2.1.2.7, Access Control Product Exits*, of this document. Please refer to the *Implementing CA SAF HFS Security* section of the *CA-ACF2 OS/390 Administrator's Guide* document for detailed information on implementing the exit.

Throughout this section assumptions are made about the naming convention for some objects. These assumptions are used for ease of discussion and do not reflect absolute standards. These assumptions are as follows:

- OMVS is the userid to be used for the OS/390 UNIX kernel.
- User-personal MVS files are prefixed *USERNN*. where *USERNN* is the userid.
- User-personal HFS files are mounted at /u/*usernn* where *usernn* is the userid.

2.5.3.1 Defining Resource Profiles and Program Control

This section lists the rules and commands needed to define the resource profiles required for OS/390 UNIX. These commands are grouped as primary FACILITY class profiles, optional RTLS FACILITY class profiles, SURROGAT class profiles, UNIXPRIV class profiles, optional CA SAF HFS FACILITY class profiles, and program control.

- *(ZUSSA060: CAT II) The IAO will ensure that the CLASMAP DEFINITIONS list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

2.5.3.1.1 Primary FACILITY Class Profiles

(1) The following rules form a base to be used to secure the primary FACILITY class profiles:

```
$KEY(BPX) TYPE(FAC)
- UID(*) PREVENT
DAEMON  UID(OMVS-uid) SERVICE(READ) ALLOW

DEBUG UID(*) PREVENT
FILEATTR.APF UID(*) PREVENT
FILEATTR.PROGCTL UID(*) PREVENT
JOBNAME UID(*) PREVENT
SAFFASTPATH UID(*) PREVENT
SERVER UID(*) PREVENT
SMF UID(*) PREVENT
STOR.SWAP UID(*) PREVENT
SUPERUSER UID(*) PREVENT
WLMSERVER UID(*) PREVENT
```

If CA SAF HFS security is to be enabled and CAIENF is the started task:

```
DAEMON UID(CAIENF-uid) SERVICE(READ) ALLOW
```

If CA SAF HFS security is to be enabled and CAIENF is the started task:

```
   SUPERUSER UID(CAIENF-uid) SERVICE(READ) ALLOW
```

(2) The following operator command is required to complete the update:

```
F ACF2,REBUILD(FAC)
```

NOTE: The OS/390 UNIX kernel userid (OMVS) must have access to the DAEMON resource. At system initialization, some daemons (e.g., inetd) inherit this userid.

NOTE: The SAFFASTPATH PREVENT rule must be coded on ACF2 systems. If access to BPX.SAFFASTPATH were allowed, OS/390 UNIX would perform permission bit checking internally instead of calling the ACP. On ACF2 systems this would bypass any audit trail of violations. If the OS/390 UNIX kernel userid (OMVS) has the ACF2 NON-CNCL privilege, the following ACF command is required:

```
INSERT SAFDEF.OEFSTAUT FUNCRET(8) ID(OEFSTAUT) -
    JOBNAME(OMVS) MODE(IGNORE) RB(BPXINIT) -
    RACROUTE(REQUEST=AUTH CLASS=FACILITY-
    ENTITY=BPX.SAFFASTPATH) -
    REP
```

- *(ZUSSA060: CAT II) The IAO will ensure that the CLASMAP DEFINITIONS list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

2.5.3.1.2 Optional RTLS FACILITY Class Profiles

If the OS/390 RTLS feature is being used for the C run-time libraries that OS/390 UNIX requires, resource profiles for RTLS may need to be added or updated. The following rule must be used to deactivate this checking if RTLS is being used and access checking is not required for any other purpose:

```
$KEY(CSVRTLS) TYPE(FAC)
NOSECCONNECT.-
```

The following operator command is required to complete the update:

```
F ACF2,REBUILD(FAC)
```

Please refer to IBM’s *OS/390 MVS Initialization and Tuning Reference* document for detailed information on using RTLS.

2.5.3.1.3 SURROGAT Class Profiles

SURROGAT class profiles are only needed if there are servers (e.g., web server) running in the OS/390 UNIX environment that must be able to act with the security context of a client and that client does not supply a password or other authenticator for the ACP. The following ACF commands and sample rule set would be required to authorize a server process (e.g., *server01*) to act as a surrogate for a client user (e.g., *user01*).

(1) If no SURROGAT class records were previously defined, use the following ACF commands:

```
SET CONTROL(GSO)
CHANGE INFODIR TYPES(R-RSUR)
```

(2) The following rules form an example to be used to secure the SURROGAT class profiles:

```
$KEY(BPX) TYPE(SUR)
- UID(*) PREVENT
SRV. UID(*) PREVENT
SRV.user01 UID(server01-uid) SERVICE(READ) ALLOW
```

(3) The following operator commands are required to complete the update:

If no SURROGAT class records were previously defined:

```
F ACF2,REFRESH(INFODIR)

F ACF2,REBUILD(SUR)
```

NOTE: The server process (e.g., server01) must also have access to the BPX.SERVER FACILITY class profile.

- *(ZUSSA060: CAT II) The IAO will ensure that the CLASMAP DEFINITIONS list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

- *(ZUSSA070: CAT II) The IAO will ensure that the INFODIR record includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

2.5.3.1.4 UNIXPRIV Class Profiles

To secure the class profiles that support superuser granularity, the following ACF commands and rules for the UNIXPRIV class must be entered:

(1) If no UNIXPRIV class records were previously defined, use the following ACF commands:

```
SET CONTROL(GSO)
CHANGE INFODIR TYPES(R-RUNI)
```

(2) The following rules form a base to be used to secure the UNIXPRIV class profiles:

```
$KEY(SUPERUSER) TYPE(UNI)
- UID(*) PREVENT
FILESYS UID(*) PREVENT
FILESYS.CHOWN UID(*) PREVENT
FILESYS.MOUNT UID(*) PREVENT
FILESYS.QUIESCE UID(*) PREVENT
FILESYS.PFSCTL UID(*) PREVENT
FILESYS.VREGISTER UID(*) PREVENT
IPC.RMID UID(*) PREVENT
PROCESS.GETPSENT UID(*) PREVENT
PROCESS.KILL UID(*) PREVENT
PROCESS.PTRACE UID(*) PREVENT
SETPRIORITY UID(*) PREVENT
```

(3) The following operator commands are required to complete the update:

If no UNIXPRIV class records were previously defined:

```
F ACF2,REFRESH(INFODIR)

F ACF2,REBUILD(UNI)
```

NOTE : A rule for the CHOWN.UNRESTRICTED UNIXPRIV profile must **not** be coded and the GSO UNIXOPTS CHOWNRES option must be selected on ACF2 systems. If the rule is coded or the NOCHOWNRES option is selected, a basic file ownership protection is defeated. Deviations from this policy must be documented and justified to the IAO.

NOTE : If CA SAF HFS security is enabled (see the next section), it takes precedence and all UNIXPRIV class resources are ignored.

- *(ZUSSA060: CAT II) The IAO will ensure that the CLASMAP DEFINITIONS list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

- *(ZUSSA070: CAT II) The IAO will ensure that the INFODIR record includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

2.5.3.1.5 Optional CA SAF HFS FACILITY Class Resources

CA SAF HFS security is an option for ACF2 that can provide alternative functions for HFS file control. One set of these functions provides superuser granularity as an alternative to the UNIXPRIV class. The following additions to the BPX rule set form a base to be used to secure the CA SAF HFS FACILITY class resources:

```
        $KEY(BPX) TYPE(FAC)
…
```

```
CAHFS.-  PREVENT
CAHFS.CHANGE.FILE.ATTRIBUTES[9] UID(*) PREVENT
CAHFS.CHANGE.FILE.AUDIT.FLAGS UID(*) PREVENT
CAHFS.CHANGE.FILE.FORMAT UID(*) PREVENT
CAHFS.CHANGE.FILE.GROUP UID(*) PREVENT
CAHFS.CHANGE.FILE.MODE UID(*) PREVENT
CAHFS.CHANGE.FILE.MODE.EGID UID(*) PREVENT
CAHFS.CHANGE.FILE.MODE.EUID UID(*) PREVENT
CAHFS.CHANGE.FILE.MODE.STICKY UID(*) PREVENT
CAHFS.CHANGE.FILE.OWNER UID(*) PREVENT
CAHFS.CHANGE.FILE.TIME UID(*) PREVENT
CAHFS.CHANGE.PRIORITY UID(*) PREVENT
CAHFS.CREATE.EXTERNAL.LINK UID(*) PREVENT
CAHFS.CREATE.LINK UID(*) PREVENT
CAHFS.CREATE.SYMBOLIC.LINK UID(*) PREVENT
CAHFS.MOUNT UID(*) PREVENT
CAHFS.PTRACE UID(*) PREVENT
CAHFS.SECURITY.DISABLE UID(*) PREVENT
CAHFS.SECURITY.ENABLE UID(*) PREVENT
CAHFS.SECURITY.STATUS UID(*) PREVENT
CAHFS.SET.PRIORITY UID(*) PREVENT
CAHFS.SET.RLIMIT UID(*) PREVENT
CAHFS.UNMOUNT UID(*) PREVENT
…
```

The following operator command is required to complete the update:

```
F ACF2,REBUILD(FAC)
```

If CA SAF HFS security is enabled, it takes precedence and all UNIXPRIV class resources are ignored.

2.5.3.1.6 Program Control

For daemon processes to be permitted to use certain privileged operations, all executables loaded into the daemon’s address space must be indicated as program control programs. These executables can come from HFS files or MVS data sets. For executables in HFS files, program control is indicated through the program control extended attribute bit. For executables in MVS data sets, program control is indicated through the ACP. ACF2 implements program control through authorized libraries. A library is authorized by being a member of the LPA list, the APF list, the MVS link list, or the ACF2 GSO LINKLST.

[9] When access is enabled to one of the CAHFS.CHANGE.FILE profiles, the action permitted varies according to the value of the SERVICE operand and the HFS file resource rules. See *Section 2.5.2.3.4, FACILITY Class Resources for CA SAF HFS (ACF2/TOP SECRET) Functions,* for a discussion of the SERVICE values. See *Section 2.5.3.4.3, Critical HFS Directories and Files for OS/390 UNIX Components,* for information on HFS file resource rules.

Some OS/390 UNIX daemons load programs from the C run-time (SYS1.LE.SCEERUN) and SYS1.LINKLIB libraries. Under the ACF2 options for program control, SYS1.LINKLIB is always indicated as a program control library because it is a member of the MVS link list. If the C run-time library (SYS1.LE.SCEERUN) is defined in the MVS link list or APF list, no further action is required. If not, the following ACF commands can be used to add it to the ACF2 GSO LINKLST:

```
SET CONTROL GSO
CHANGE LINKLST LIBRARY(SYS1.LE.SCEERUN)
```

The following operator command is required to complete the update:

```
F ACF2,REFRESH(LINKLST)
```

2.5.3.2 Defining Users and Groups

To authorize a user to access OS/390 UNIX resources, two actions are required:

(1) An OMVS GROUP Profile record with a Group ID (GID) number should be added under the group name.

(2) An OMVS USER Profile record with a UNIX Userid (UID) number, home directory (HOME), and shell (PROGRAM) should be added under the userid.

This section lists the commands to accomplish these user and group definitions.

NOTE: The GSO UNIXOPTS will specify the CHOWNRES option. DFTGROUP and DFTUSER fields can be used for FTP socket applications on non classified systems. DFTGROUP and DFTUSER fields will not be used on classified systems and systems that do not run FTP. When coding these options be sure that the restrictions specified in *Section 2.5.6.2, Paragraph 7,* of this STIG are followed.

- *(ZUSSA053: CAT II) The IAO will ensure that UINIXOPTS record specifies CHOWNRES.*

- *(ZUSSA050: CAT II) The IAO will ensure that UINIXOPTS record does not specify DFTGROUP and DFTUSER fields for classified systems and systems not using FTP.*

2.5.3.2.1 Groups

(1) The following ACF commands can be used to create the group profile records that are required for OS/390 UNIX:

If no GROUP Profile records were previously defined:

```
SET CONTROL(GSO)
CHANGE INFODIR TYPES(R-PGRP)
```

Define Privileged Groups:

```
SET PROFILE(GROUP) DIVISION(OMVS)
INSERT OMVSGRP GID(3)
INSERT STCOMVS GID(4)
INSERT TTY GID(1)
INSERT SYS1 GID(0)
```

(2) The following operator commands are required to complete the update:

If no GROUP Profile records were previously defined:

```
F ACF2,REFRESH(INFODIR)
```

(3) The following ACF commands illustrate what is required for existing or new groups when users in those groups use OS/390 UNIX:

```
SET PROFILE(GROUP) DIVISION(OMVS)
INSERT group-name GID(gid)
```

(4) The following operator commands are required to complete the update:

```
F ACF2,REBUILD(GRP),CLASS(P)
F ACF2,OMVS
```

NOTE: For privileged groups, the value of GID must be between 1 and 99, unique within the site’s ACF2 database. For unprivileged groups, the value of GID must be between 100 and 16,777,215, unique within the site’s ACF2 database.

2.5.3.2.2 Privileged Users – Started Tasks and Daemons

(1) The following ACF commands can be used to create and change the userids that are required for OS/390 UNIX:

```
SET LID
INSERT OMVS NAME(OMVS) GROUP(STCOMVS) STC
INSERT BPXOINIT NAME(BPXOINIT) GROUP(STCOMVS) STC
INSERT BPXAS NAME(BPXAS) GROUP(STCOMVS) STC
INSERT BPXSTOP NAME(BPXSTOP) GROUP(STCOMVS) STC
INSERT BPXROOT NAME(BPXROOT) GROUP(OMVSGRP) -
      PASSWORD(password)
INSERT RMFGAT NAME(RMFGAT) GROUP(STCOMVS) STC
```

(2) If CA SAF HFS security is to be enabled and CAIENF is the started task’s userid:

```
CHANGE CAIENF GROUP(STCOMVS)
```

(3) If no USER Profile records were previously defined:

```
SET CONTROL(GSO)
CHANGE INFODIR TYPES(R-PUSR)
```

Define Privileged USERs:

```
SET PROFILE(USER) DIVISION(OMVS)
INSERT OMVS UID(0) HOME(/) OMVSPGM(/bin/sh)
INSERT BPXOINIT UID(0) HOME(/) OMVSPGM(/bin/sh)
INSERT BPXAS UID(0) HOME(/) OMVSPGM(/bin/sh)
INSERT BPXSTOP UID(0) HOME(/) OMVSPGM(/bin/sh)
INSERT BPXROOT UID(0) HOME(/) OMVSPGM(/bin/sh)
INSERT RMFGAT UID(uid) HOME(/) OMVSPGM(/bin/sh)
```

NOTE: At eTrust CA-ACF2 6.3 and below, the OMVSPGM field in the user profile record was previously named PROGRAM.

(4) If CA SAF HFS security is to be enabled and CAIENF is the started task's userid:

```
INSERT CAIENF UID(uid) HOME(/) OMVSPGM(/bin/sh)
```

(5) The following operator commands are required to complete the update:

If no USER Profile records were previously defined:

```
F ACF2,REFRESH(INFODIR)

F ACF2,REBUILD(USR),CLASS(P)
F ACF2,OMVS
```

NOTE: When UID(0) is not required, the value of uid must be between 1 and 99 (unique within the site's ACF2 database).

NOTE: The INSERT RMFGAT command assumes that this userid did not already exist. If the userid already exists, only a CHANGE RMFGAT GROUP(STCOMVS) command may be required.

NOTE: The assignment of the ACF2 NON-CNCL attribute is effectively equal to assigning UID(0) to that user in the OS/390 UNIX environment.

2.5.3.2.3 Privileged Users – OS/390 UNIX System Administrators

A user is considered privileged if that user's userid has access to superuser status by being assigned UID 0, or access to the BPX.SUPERUSER resource, or access to services permitted through the BPX.DAEMON or BPX.SERVER resources.

(1) The following illustrates the minimum version of the ACF commands (CHANGE/INSERT) required for an existing user when the user needs *full-time superuser authority* for OS/390 UNIX:

```
SET LID
CHANGE existing-user GROUP(existing-group)
SET PROFILE(USER) DIVISION(OMVS)
INSERT existing-user UID(0) HOME(/u/existing-user) OMVSPGM(/bin/sh)
```

NOTE: At eTrust CA-ACF2 6.3 and below, the OMVSPGM field in the user profile record was previously named PROGRAM.

(2) The following illustrates the minimum version of the ACF commands (INSERT/INSERT) required for a new user when the user needs *full-time superuser authority* for OS/390 UNIX:

```
SET LID
INSERT new-user NAME(new-user name) GROUP(existing-group)
PASSWORD(password)
SET PROFILE(USER) DIVISION(OMVS)
INSERT new-user UID(0) HOME(/u/new-user) OMVSPGM(/bin/sh)
```

(3) After changes or inserts are performed, the following operator commands are required to complete the update:

```
F ACF2,REBUILD(USR),CLASS(P)
F ACF2,OMVS
```

NOTE: The assignment of UID(0) to any user must follow the guidelines in Section 2.5.2.6.1, Privileged Users and Special Groups, of this document. All userids with UID(0) must be documented to the IAO.

NOTE: The assignment of the ACF2 NON-CNCL attribute is effectively equal to assigning UID(0) to that user in the OS/390 UNIX environment.

(4) The following illustrates the minimum version of the ACF commands (CHANGE/INSERT) required for an existing user when the user needs *the ability to switch to superuser authority* for OS/390 UNIX:

```
SET LID
CHANGE existing-user GROUP(existing-group)
SET PROFILE(USER) DIVISION(OMVS)
INSERT existing-user UID(uid) HOME(/u/existing-user) OMVSPGM(/bin/sh)
```

(5) The following illustrates the minimum version of the ACF commands (INSERT/INSERT) required for a new user when the user needs *the ability to switch to superuser authority* for OS/390 UNIX:

```
SET LID
INSERT new-user NAME(user name) GROUP(existing-group)
PASSWORD(password)
SET PROFILE(USER) DIVISION(OMVS)
INSERT new-user UID(uid) HOME(/u/new-user) PROGRAM(/bin/sh)
```

(6) If (4) or (5) above are entered perform an update to the $KEY(BPX) TYPE(FAC) rule set to allow the user access to the SUPERUSER resource.

(7) After changes or inserts are performed, the following operator commands are required to complete the update:

```
F ACF2,REBUILD(USR),CLASS(P)
F ACF2,OMVS
```

NOTE: The value of uid must be between 100 and 16,777,215, unique within the site's ACF2 database.

NOTE: The GROUP for the user must have a valid GID value.

2.5.3.2.4 Unprivileged Users

(1) The following illustrates the minimum version of the ACF commands (CHANGE/INSERT) required for an existing user when the user needs *unprivileged access* for OS/390 UNIX:

```
SET LID
CHANGE existing-user GROUP(existing-group)
SET PROFILE(USER) DIVISION(OMVS)
INSERT existing-user UID(uid) HOME(/u/existing-user) OMVSPGM(/bin/sh)
```

NOTE: At eTrust CA-ACF2 6.3 and below, the OMVSPGMPROGRAM field in the user profile record was previously named PROGRAM.

(2) The following illustrates the minimum version of the ACF commands (INSERT/INSERT) required for a new user when the user needs *unprivileged access* for OS/390 UNIX:

```
SET LID
INSERT new-user NAME(user name) GROUP(existing-group)
PASSWORD(password)
SET PROFILE(USER) DIVISION(OMVS)
INSERT new-user UID(uid) HOME(/u/new-user) OMVSPGM(/bin/sh)
```

(3) After changes or inserts are performed, the following operator commands are required to complete the update:

```
F ACF2,REBUILD(USR),CLASS(P)
F ACF2,OMVS
```

NOTE: The value of uid must be between 100 and 16,777,215 (unique within the site's ACF2 database).

NOTE: The GROUP for the user must have a valid GID value.

2.5.3.3 Defining Started Tasks

ACF2 does not require a separate definition for started tasks. When the userid for a started task is created, the STC attribute indicates that the userid is to be used when a started task of the same name is started.

2.5.3.4 Protecting Data

In order to preserve system integrity, data must be protected from unauthorized or inadvertent modification. This section lists the rules and commands that are used to protect the system components and data that are essential to OS/390 UNIX. Guidelines for protecting user HFS data are also discussed.

2.5.3.4.1 MVS Data Sets for OS/390 UNIX Components

The following additions to the SYS1 rule set can be used as a base to secure MVS data sets that contain OS/390 UNIX components:

```
$KEY(SYS1)
…
ABPX-UID(*) READ(P)
AFOM-UID(*) READ(P)
BPA.ABPA- UID(*) READ(P)
CMX.ACMX-UID(*) READ(P)
CMX.SCMX-UID(*) READ(P)
SBPX-UID(*)  READ(A)
[- The SYS1.SBPX- data sets are used by TSO/E users of OS/390 UNIX; a READ(A)
specification is acceptable for this resource.]
SFOM- READ(P)
…
```

2.5.3.4.2 MVS Data Sets Containing OS/390 Hierarchical File Systems

The following additions to the SYS1 and SYS3 rule sets can be used as a base to secure MVS data sets that contain Hierarchical File Systems containing OS/390 UNIX system components:

$KEY(SYS1)
…
OE.- UID(*OMVS-uid*) READ(A) WRITE(L) ALLOC(L) EXEC(A)
…
$KEY(SYS3)
…
OE.- UID(*OMVS-uid*) READ(A) WRITE(L) ALLOC(L) EXEC(A)
…

NOTE: The OS/390 UNIX kernel userid (OMVS) must have appropriate access to these MVS data sets for HFS data to be accessible.

NOTE: This example assumes that system HFS data sets are named with SYS1 or SYS3 as the high-level node, followed by OE as the second node. This may not be the convention at all sites.

If individual user HFS data sets are being allocated, rules must be established to allow OS/390 UNIX to access the data sets. The following illustrates the commands that must be executed to protect each MVS data set that contains users' Hierarchical File Systems:

> $KEY(*user*)
> …
> OE.- UID(*user-uid*) READ(A) EXEC(A)
> OE.- UID(*OMVS-uid*) READ(A) WRITE(L) EXEC(A)
> …

NOTE: The OS/390 UNIX kernel userid (OMVS) must have appropriate access to these MVS data sets for HFS data to be accessible.

NOTE: This example assumes that user HFS data sets are named with the user's userid as the high-level node, followed by OE as the second node. This may not be the convention at all sites. The choice of naming convention impacts the use of the automount facility. Please refer to IBM's OS/390 UNIX System Services Planning document for detailed information on automount and MapName files.

NOTE: If users are given access authority that allows them to update MVS data sets that contain HFS files, a potential system integrity or security exposure exists. Specifically, permission and extended attribute bit settings might be altered outside of OS/390 UNIX control. Therefore users must not be given update authority to MVS data sets containing HFS files.

2.5.3.4.3 Critical HFS Directories and Files for OS/390 UNIX Components

As stated in *Section 2.5.2.5, OS/390 UNIX HFS Directories and Files*, of this document, appropriate UNIX permissions must be maintained on the specified directories and files to maintain system integrity. These permission settings are not maintained in ACF2, but must be checked and maintained within the OS/390 UNIX environment.

(1) If the optional CA SAF HFS security is enabled, UNIX permission bit checking is bypassed and access rules must be written.

(2) If no HFS class records were previously defined, use the following ACF commands:

```
SET CONTROL(GSO)
CHANGE INFODIR TYPES(R-RHFS)
```

(3) The following rule sets form a base to be used to secure the critical HFS directories and files and to allow for separate control of user HFS directories and files:

```
$KEY(/) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW

$KEY(/BIN) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW
- UID(-) SERVICE(READ,EXEC) ALLOW

$KEY(/DEV) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW
- UID(-) SERVICE(READ,EXEC) ALLOW
CONSOLE  PREVENT
NULL SERVICE(READ,UPDATE,EXEC) ALLOW

$KEY(/ETC) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW
- UID(-) SERVICE(READ,EXEC) ALLOW
AUTO$MASTER  PREVENT
INETD$CONF  PREVENT
INIT$OPTIONS  PREVENT
RESOLV$CONF  PREVENT
SERVICES  PREVENT
STEPLIB  PREVENT
TABLENAME  PREVENT

$KEY(/LIB) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW
- UID(-) SERVICE(READ,EXEC) ALLOW

$KEY(/SAMPLES) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW
- UID(-) SERVICE(READ,EXEC) ALLOW

$KEY(/TMP) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW
```

```
- UID(-) SERVICE(READ,UPDATE,ADD,EXEC) ALLOW

$KEY(/U) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW
- UID(-) PREVENT
user01.-  NEXTKEY($$user01)
…
usernn.-  NEXTKEY($$usernn)

$KEY(/USR) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW
- UID(-) SERVICE(READ,EXEC) ALLOW
LIB.CRON.AT$ALLOW  PREVENT
LIB.CRON.AT$DENY  PREVENT
LIB.CRON.CRON$ALLOW  PREVENT
LIB.CRON.CRON$DENY  PREVENT

$KEY(/VAR) TYPE(HFS)
UID(-) SERVICE(READ) ALLOW
- UID(-) SERVICE(READ,EXEC) ALLOW
```

- *(ZUSS0080: CAT II) The IAO will ensure that the HFSSEC resource access is restricted to appropriate personnel with appropriate logging based on bit and audit settings.*

(4) The following operator commands are required to complete the update:

If no HFS class records were previously defined:

```
F ACF2,REFRESH(INFODIR)

F ACF2,REBUILD(HFS)
```

NOTE: Even when CA SAF HFS security is enabled, UNIX permission bits must be maintained on the critical HFS directories and files.

NOTE: Because of path translation (noted next), all paths are represented in uppercase in rules.

NOTE: HFS paths are folded to upper case and, if necessary, truncated to 255 characters. All slash characters after the first are translated to periods and other special characters (including periods, asterisks, dashes, plus signs, blanks, and quote marks) are translated to the dollar sign ($) character.

NOTE: To protect user data and to permit either de-coupling of user data rules from operating system data rules or de-centralized access control (via GSO RULEOPTS NOCENTRAL), the ACF2 NEXTKEY facility can be used.

NOTE: The $KEY(/U) rule assumes that user directories are mounted at the /u mount point and use the userid as the next level. This may not be the convention at all sites. The choice of naming convention impacts ACF2 rule writing and the use of the automount facility. Please refer to IBM's OS/390 UNIX System Services Planning document for detailed information on automount and MapName files.

NOTE: The SAFHFUSR exit has the ability to alter path translation, indicate user path processing, and indicate no validation for user owned files. Please refer to the Implementing CA SAF HFS Security section of the CA-ACF2 OS/390 Administrator's Guide for detailed information on implementing the exit.

(5) If the optional CA SAF HFS security is enabled, rules for individual user directories are also necessary. If the ACF2 NEXTKEY facility is used at the /u mount point as shown above, the following sample rules form a base to be used to secure user HFS directories and files:

```
$KEY($$user01) TYPE(HFS)
$PREFIX(/u.user01)
UID(user01-uid) SERVICE(READ) ALLOW
- UID(user01-uid) SERVICE(READ,UPDATE,ADD,EXEC) ALLOW
```

NOTE: The HFS path translation and SAFHFUSR exit notes above also apply to user directories and files.

NOTE: The example assumes that user directories are mounted at the /u mount point and use the userid as the next level. This may not be the convention at all sites. The choice of naming convention impacts ACF2 rule writing and the use of the automount facility. Please refer to IBM's OS/390 UNIX System Services Planning document for detailed information on automount and MapName files.

2.5.3.5 Protecting Sensitive Commands, Utilities, and Language Interfaces

There are various commands, utilities, and language interfaces for OS/390 UNIX that could cause significant damage if used with malicious intent or used incorrectly by unauthorized personnel. This section provides guidelines for managing the security of those items.

2.5.3.5.1 Interactive Commands, Utilities, and Language Interfaces

Section 2.5.2.3, Resource Profiles, and Section 2.5.2.10, Sensitive TSO/E and Shell Commands and Environment Variable Settings, of this document provide lists of the general resources and commands that can be protected. If the rules in Section 2.5.3.1, Defining Resource Profiles and Program Control, have been implemented, the system has protections in place. The following examples illustrate rules to provide access to protected resources:

(1) Allow a user to use the OS/390 shell extattr command to set the program control attribute on a file:

```
$KEY(BPX) TYPE(FAC)
…
FILEATTR.PROGCTL UID(privileged-user-uid) SERVICE(READ) ALLOW
…
```

(2) Allow a user to use the TSO/E mount command to mount file systems and use the setuid option:

When CA SAF HFS security is not enabled:

```
$KEY(SUPERUSER) TYPE(UNI)
…
FILESYS.MOUNT UID(privileged-user-uid) SERVICE(READ,UPDATE)
ALLOW
```

When CA SAF HFS security is enabled:

```
$KEY(BPX) TYPE(FAC)
…
CAHFS.MOUNT UID(privileged-user-uid) ALLOW
CAHFS.UNMOUNT UID(privileged-user-uid) ALLOW
…
```

(3) Allow a user to use daemon privileges and to start daemon processes:

```
$KEY(BPX) TYPE(FAC)
…
DAEMON UID(privileged-user-uid) SERVICE(READ) ALLOW
…
```

(4) Allow a user to read and write any HFS file and read, search, and write any HFS directory:

When CA SAF HFS security is not enabled:

```
$KEY(SUPERUSER) TYPE(UNI)
…
FILESYS UID(privileged-user-uid) SERVICE(DELETE) ALLOW
…
```

When CA SAF HFS security is enabled:

```
$KEY(BPX) TYPE(FAC)
…
```

```
SUPERUSER UID(privileged-user-uid) SERVICE(READ) ALLOW
```

(There is no CA SAF HFS privilege that maps to the UNIXPRIV FILESYS privilege. In this case the user receives full superuser authority.)

(5) Allow a user to use the SAFHFMOD utility to enable or disable CA SAF HFS security:

```
$KEY(BPX) TYPE(FAC)
…
CAHFS.SECURITY.DISABLE UID(privileged-user-uid) ALLOW
CAHFS.SECURITY.ENABLE UID(privileged-user-uid) ALLOW
…
```

It must be kept in mind that users with UID(0), users with permission to the BPX.SUPERUSER profile, and users with the ACF2 NON-CNCL attribute bypass any of the security protection implemented by resource profiles.

2.5.3.5.2 OS/390 System Commands

Section 3.2.5.6, OS/390 System Command Controls, of this document specifies protections that cover the OS/390 system commands that can specifically impact OS/390 UNIX operations. These commands are F BPXOINIT, SET OMVS, and SETOMVS. It is worth confirming that the following resource rules, appropriate equivalents, or more restrictive rules have been defined:

```
$KEY(MVS) TYPE(OPR)
…
MODIFY.STC.* UID(operator-group-uid) SERVICE(UPDATE) LOG
SET.* UID(operator-group-uid) SERVICE(UPDATE) LOG
SETOMVS.OMVS UID(operator-group-uid) SERVICE(UPDATE) LOG
…
```

2.5.4 RACF (OS/390 Security Server) Implementation for OS/390 UNIX

This section describes the commands needed to implement the security guidelines for OS/390 UNIX under the RACF (OS/390 Security Server) ACP. The following task categories are described:

- Resource profiles and controls
- Users and groups
- Started tasks
- Data
- Sensitive commands and utilities

Please note that the use of the IRRSXT00 installation exit can significantly alter the behavior of RACF with regard to security decisions for OS/390 UNIX. The use of this exit must conform to the procedures documented in *Section 2.1.2.7, Access Control Product Exits*, of this document.

Please refer to IBM's *OS/390 Security Server (RACF) Callable Services* document for detailed information on implementing the exit.

Throughout this section assumptions are made about the naming convention for some objects. These assumptions are used for ease of discussion and do not reflect absolute standards. These assumptions are as follows:

- ADMIN is an existing group to be used for the ownership of system resources excluding data.
- SYS1 is an existing group to be used for the ownership of system data.
- OMVSKERN is the userid to be used for the OS/390 UNIX kernel.
- User-personal MVS files are prefixed *USERNN*. where *USERNN* is the userid.
- User-personal HFS files are mounted at "/u/*usernn*" where *usernn* is the userid.

2.5.4.1 Defining Resource Profiles and Program Control

This section lists the commands needed to define the resource profiles required for OS/390 UNIX. These commands are grouped as primary FACILITY class profiles, optional RTLS FACILITY class profiles, SURROGAT class profiles, UNIXPRIV class profiles, and program control.

2.5.4.1.1 Primary FACILITY Class Profiles

Use the following commands to define the primary FACILITY class profiles:

```
RDEFINE FACILITY BPX.*  UACC(NONE) OWNER(ADMIN)
RDEFINE FACILITY BPX.DAEMON  UACC(NONE) OWNER(ADMIN)
RDEFINE FACILITY BPX.DEBUG  UACC(NONE) OWNER(ADMIN)
RDEFINE FACILITY BPX.FILEATTR.APF  UACC(NONE) OWNER(ADMIN)
RDEFINE FACILITY BPX.FILEATTR.PROGCTL  UACC(NONE) -
    OWNER(ADMIN)
RDEFINE FACILITY BPX.JOBNAME  UACC(NONE) OWNER(ADMIN)
RDEFINE FACILITY BPX.SERVER  UACC(NONE) OWNER(ADMIN)
RDEFINE FACILITY BPX.SMF  UACC(NONE) OWNER(ADMIN)
RDEFINE FACILITY BPX.STOR.SWAP  UACC(NONE) OWNER(ADMIN)
RDEFINE FACILITY BPX.SUPERUSER  UACC(NONE) OWNER(ADMIN)
RDEFINE FACILITY BPX.WLMSERVER  UACC(NONE) OWNER(ADMIN)
SETROPTS RACLIST(FACILITY) REFRESH
```

The BPX.SAFFASTPATH FACILITY profile must not be defined. When BPX.SAFFASTPATH is defined, successful security checks are not audited. Because of audit requirements for some critical files, a configuration with auditing disabled is not permitted.

- *(ZUSSR060: CAT II) The IAO will ensure that the ACTIVE CLASSES list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

- *(ZUSSR070: CAT II) The IAO will ensure that the SETR RACLIST CLASSES list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

2.5.4.1.2 Optional RTLS FACILITY Class Profiles

If the OS/390 RTLS feature is being used for the C run-time libraries that OS/390 UNIX requires, resource profiles for RTLS may need to be added or updated. The following command must be used to deactivate this checking if RTLS is being used and access checking is not required for any other purpose:

```
RDEFINE FACILITY CSVRTLS.NOSECCONNECT.*  OWNER(ADMIN)
SETROPTS RACLIST(FACILITY) REFRESH
```

Please refer to IBM's *OS/390 MVS Initialization and Tuning Reference* document for detailed information on using RTLS.

- *(ZUSSR060: CAT II) The IAO will ensure that the ACTIVE CLASSES list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

- *(ZUSSR070: CAT II) The IAO will ensure that the SETR RACLIST CLASSES list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

2.5.4.1.3 SURROGAT Class Profiles

SURROGAT class profiles are only needed if there are servers (e.g., web server) running in the OS/390 UNIX environment that must be able to act with the security context of a client and that client does not supply a password or other authenticator for the ACP. The following commands are required to authorize a server process (e.g., *server01*) to act as a surrogate for a client user (e.g., *user01*):

If SURROGAT was not previously defined: SETROPTS CLASSACT(SURROGAT)
If SURROGAT was not previously defined: SETROPTS RACLIST(SURROGAT)
RDEFINE SURROGAT BPX.SRV.*user01* OWNER(ADMIN)
PERMIT BPX.SRV.*user01* CLASS(SURROGAT) ACCESS(READ) ID(*server01*)
SETROPTS RACLIST(SURROGAT) REFRESH

NOTE: The server process (e.g., server01) must also have access to the BPX.SERVER FACILITY class profile.

- *(ZUSSR060: CAT II) The IAO will ensure that the ACTIVE CLASSES list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

- *(ZUSSR070: CAT II) The IAO will ensure that the SETR RACLIST CLASSES list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

2.5.4.1.4 UNIXPRIV Class Profiles

Use the following commands to define the UNIXPRIV class profiles for superuser granularity support:

```
If UNIXPRIV was not previously defined:  SETROPTS CLASSACT(UNIXPRIV)
If UNIXPRIV was not previously defined:  SETROPTS RACLIST(UNIXPRIV)
RDEFINE UNIXPRIV SUPERUSER.* -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.FILESYS -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.FILESYS.CHOWN -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.FILESYS.MOUNT -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.FILESYS.QUIESCE -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.FILESYS.PFSCTL -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.FILESYS.VREGISTER -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.IPC.RMID -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.PROCESS.GETPSENT -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.PROCESS.KILL -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.PROCESS.PTRACE -
        UACC(NONE) OWNER(ADMIN)
RDEFINE UNIXPRIV SUPERUSER.SETPRIORITY -
        UACC(NONE) OWNER(ADMIN)
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

Do not define the CHOWN.UNRESTRICTED UNIXPRIV profile. If the profile is defined, a basic file ownership protection is defeated. Deviations from this policy must be documented and justified to the IAO.

- *(ZUSSR060: CAT II) The IAO will ensure that the ACTIVE CLASSES list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

- *(ZUSSR070: CAT II) The IAO will ensure that the SETR RACLIST CLASSES list includes entries for the FACILITY, SURROGAT, and UNIXPRIV resource classes.*

2.5.4.1.5 Program Control

For daemon processes to be permitted to use certain privileged operations, all executables loaded into the daemon’s address space must be indicated as program control programs. These executables can come from HFS files or MVS data sets. For executables in HFS files, program control is indicated through the program control extended attribute bit. For executables in MVS data sets, program control is indicated through the ACP. RACF implements program control through the PROGRAM resource profile.

Some OS/390 UNIX daemons load programs from the C run-time (SYS1.LE.SCEERUN) and SYS1.LINKLIB libraries. The following commands can be used to indicate that all programs from those libraries are marked for program control:

```
RALTER PROGRAM ** ADDMEM('SYS1.LE.SCEERUN'/volser/NOPADCHK)
RALTER PROGRAM ** ADDMEM('SYS1.LINKLIB'/******/NOPADCHK)
SETROPTS WHEN(PROGRAM) REFRESH
```

NOTE: If the PROGRAM ** profile has not been defined previously, change the RALTER to RDEFINE and add the appropriate UACC operand.

NOTE: If the site desires not to add the entire C run-time and SYS1.LINKLIB libraries for program control, individual program entries can be made. Please refer to IBM's OS/390 UNIX System Services Planning document for detailed information.

2.5.4.2 Defining Users and Groups

To authorize a user to access OS/390 UNIX resources, two actions are required:

A Group ID (GID) number must be added to the OMVS segment of the RACF group profile for the user's default group.

A UNIX Userid (UID) number, home directory (HOME), and shell (PROGRAM) must be added to the OMVS segment of the user's RACF user profile.

This section lists the commands to accomplish these user and group definitions.

The BPX.DEFAULT.USER profile will only be used for FTP socket applications. When coding this resource be sure that the restrictions specified in *Section 2.5.6.2 paragraph 7* of this STIG are followed.

- *(ZUSSR050: CAT II) The IAO will ensure that the BPX.DEFAULT.USER for the FACILITY resource class is only used for FTP socket applications on non classified systems.*

2.5.4.2.1 Groups

The following commands can be used to create new groups (OMVSGRP, STCOMVS, TTY) and to update an existing group (SYS1) that are required for OS/390 UNIX:

```
    ADDGROUP OMVSGRP OMVS(GID(3)) OWNER(ADMIN)
    ADDGROUP STCOMVS OMVS(GID(4)) OWNER(ADMIN)
    ADDGROUP TTY OMVS(GID(1)) OWNER(ADMIN)
```

```
ALTGROUP SYS1 OMVS(GID(0))
```

The following commands illustrate the minimum versions of the command (ALTGROUP) required for existing groups and the command (ADDGROUP) required for new groups when users of those groups use OS/390 UNIX:

```
ALTGROUP existing-group OMVS(GID(gid))
ADDGROUP new-group OMVS(GID(gid)) OWNER(ADMIN)
```

NOTE: For privileged groups, the value of GID must be between 1 and 99, unique within the site’s RACF database. For unprivileged groups, the value of GID must be between 100 and 16,777,215, unique within the site’s RACF database.

2.5.4.2.2 Privileged Users – Started Tasks and Daemons

The following commands can be used to create the userids that are required for OS/390 UNIX:

```
ADDUSER OMVS DFLTGRP(STCOMVS) OWNER(ADMIN) -
    NOPASSWORD OMVS(UID(0) HOME(‘/’) PROGRAM(‘/bin/sh’))
PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(OMVS)
ADDUSER BPXROOT DFLTGRP(OMVSGRP) OWNER(ADMIN) -
    NOPASSWORD OMVS(UID(0) HOME(‘/’) PROGRAM(‘/bin/sh’))
ADDUSER RMFGAT DFLTGRP(STCOMVS) OWNER(ADMIN) -
    NOPASSWORD OMVS(UID(uid) HOME(‘/’) PROGRAM(‘/bin/sh’))
ADDUSER BPXSTOP DFLTGRP(STCOMVS) OWNER(ADMIN) -
    NOPASSWORD OMVS(UID(0) HOME(‘/’) PROGRAM(‘/bin/sh’))
```

NOTE: When UID(0) is not required, the value of uid must be between 1 and 99, unique within the site’s RACF database.

NOTE: When a userid (such as OMVS) is created with the NOPASSWORD and (default) NOOIDCARD parameters, it is a protected userid. This type of userid cannot be used in any instance where a password is required and it cannot be revoked due to failed password attempts. This is valuable for started tasks or OS/390 UNIX daemons that might otherwise be disabled by a user’s attempt to compromise the system. (This can only be done where all systems accessing the RACF database are at the Version 2, Release 8 level or higher of OS/390.)

NOTE: The assignment of the RACF PRIVILEGED or TRUSTED attributes is effectively equal to assigning UID(0) to that user in the OS/390 UNIX environment.

2.5.4.2.3 Privileged Users – OS/390 UNIX System Administrators

A user is considered privileged if that user’s userid has access to superuser status by being assigned UID 0, or access to the BPX.SUPERUSER profile, or access to services permitted through the BPX.DAEMON or BPX.SERVER profiles.

(1) The following illustrates the minimum version of the command (ALTUSER) required for a user when the user needs *full-time superuser authority* for OS/390 UNIX:

Existing USER:

```
ALTUSER existing-user -
OMVS(UID(0) HOME('/u/existing-user') PROGRAM('/bin/sh'))
```

New User:

```
ADDUSER new-user DFLTGRP(existing-group) OWNER(ADMIN) -
PASSWORD(password) -
OMVS(UID(0) HOME('/u/new-user') PROGRAM('/bin/sh'))
```

NOTE: The assignment of UID(0) to any user must follow the guidelines in *Section 2.5.2.6.1, Privileged Users and Special Groups*, of this document. All userids with UID(0) must be documented to the IAO.

NOTE: The assignment of the RACF PRIVILEGED or TRUSTED attributes is effectively equal to assigning UID(0) to that user in the OS/390 UNIX environment.

(2) The following illustrates the minimum version of the command required for a user when the user needs *the ability to switch to superuser authority* for OS/390 UNIX:

Existing User:

```
ALTUSER existing-user -
OMVS(UID(uid) HOME('/u/existing-user') PROGRAM('/bin/sh'))
PERMIT BPX.SUPERUSER CLASS(FACILITY) ACCESS(READ) -
ID(existing-user)
```

New User:

```
ADDUSER new-user DFLTGRP(existing-group) OWNER(ADMIN) -
PASSWORD(password) -
OMVS(UID(uid) HOME('/u/new-user') PROGRAM('/bin/sh'))
PERMIT BPX.SUPERUSER CLASS(FACILITY) ACCESS(READ) ID(new-user)
```

NOTE: The value of must be between 100 and 16,777,215 (unique within the site's RACF database).

NOTE: The DFLTGRP for the user must have a valid GID value.

2.5.4.2.4 Unprivileged Users

The following illustrates the minimum version of the command required for a user when the user needs *unprivileged access* for OS/390 UNIX:

Existing User:

```
ALTUSER existing-user -
OMVS(UID(uid) HOME('/u/existing-user') PROGRAM('/bin/sh'))
```

New User:

```
ADDUSER new-user DFLTGRP(existing-group-name) OWNER(ADMIN) -
PASSWORD(password) -
OMVS(UID(uid) HOME(‘/u/new-user’) PROGRAM(‘/bin/sh’))
```

NOTE: The value of uid must be between 100 and 16,777,215 (unique within the site’s RACF database).

NOTE: The DFLTGRP for the user must have a valid GID value.

2.5.4.3 Defining Started Tasks

The following commands can be used to define the started tasks that are required for OS/390 UNIX:

```
SETROPTS GENERIC(STARTED)
RDEFINE STARTED OMVS.*  UACC(NONE) OWNER(ADMIN) -
      STDATA(USER(OMVS) GROUP(STCOMVS) TRUSTED(NO))
RDEFINE STARTED BPXOINIT.* UACC(NONE) OWNER(ADMIN) -
      STDATA(USER(OMVS) GROUP(STCOMVS) TRUSTED(NO))
RDEFINE STARTED BPXAS.* UACC(NONE) OWNER(ADMIN) -
      STDATA(USER(OMVS) GROUP(STCOMVS) TRUSTED(NO))
RDEFINE STARTED BPXSTOP.* UACC(NONE) OWNER(ADMIN) -
      STDATA(USER(BPXSTOP) GROUP(STCOMVS) TRUSTED(NO))
RDEFINE STARTED RMFGAT.* UACC(NONE) OWNER(ADMIN) -
      STDATA(USER(RMFGAT) GROUP(STCOMVS) TRUSTED(NO))
SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
SETROPTS RACLIST(STARTED) REFRESH
```

NOTE: If the RMFGAT started task had been defined previously, the RDEFINE is not required but the definition must be checked to ensure that it uses the RMFGAT userid. This is required so that the appropriate UID is used when the RMFGAT task accesses OS/390 UNIX data.

2.5.4.4 Protecting Data

In order to preserve system integrity, data must be protected from unauthorized or inadvertent modification. This section lists the commands that are used to protect the system components and data that are essential to OS/390 UNIX. Guidelines for protecting user HFS data are also discussed.

2.5.4.4.1 MVS Data Sets for OS/390 UNIX Components

The following commands can be used to provide the required access control for the MVS data sets that contain OS/390 UNIX components:

```
    ADDSD ‘SYS1.ABPX*.**’ OWNER(SYS1) UACC(NONE)
    ADDSD ‘SYS1.AFOM*.**’ OWNER(SYS1) UACC(NONE)
    ADDSD ‘SYS1.BPA.ABPA*.**’ OWNER(SYS1) UACC(NONE)
    ADDSD ‘SYS1.CMX.ACMX*.**’ OWNER(SYS1) UACC(NONE)
    ADDSD ‘SYS1.SBPX*.**’ OWNER(SYS1) UACC(NONE)
    PERMIT ‘SYS1.SBPX*.**’ ACCESS(READ) ID(*)
```

[- The SYS1.SBPX* data sets are used by TSO/E users of OS/390 UNIX; an ACCESS(READ) specification for all users is acceptable for this resource.]

```
    ADDSD ‘SYS1.SFOM*.**’ OWNER(SYS1) UACC(NONE)
    ADDSD ‘SYS1.CMX.SCMX*.**’ OWNER(SYS1) UACC(NONE)
```

2.5.4.4.2 MVS Data Sets Containing OS/390 Hierarchical File Systems

The following commands can be used to provide the required access control for the MVS data sets that contain Hierarchical File Systems containing OS/390 UNIX components:

```
ADDSD ‘SYS1.OE.**’ OWNER(SYS1) UACC(NONE) AUDIT(ALL(UPDATE))
PERMIT ‘SYS1.OE.**’ ACCESS(UPDATE) ID(OMVS)
ADDSD ‘SYS3.OE.**’ OWNER(SYS1) UACC(NONE) AUDIT(ALL(UPDATE))
PERMIT ‘SYS3.OE.**’ ACCESS(UPDATE) ID(OMVS)
```

NOTE: The OS/390 UNIX kernel userid (OMVS) must have appropriate access to these MVS data sets for HFS data to be accessible.

NOTE: This example assumes that system HFS data sets are named with SYS1 or SYS3 as the high-level node, followed by OE as the second node. This may not be the convention at all sites.

If individual user HFS data sets are being allocated, rules must be established to allow OS/390 UNIX to access the data sets. The following illustrates the commands that can be used to provide the required access control for each MVS data set that contains users’ Hierarchical File Systems:

```
    ADDSD ‘user.OE.**’ OWNER(user) UACC(NONE) AUDIT(ALL(UPDATE))
    PERMIT ‘user.OE.**’ ACCESS(READ) ID(user)
    PERMIT ‘user.OE.**’ ACCESS(UPDATE) ID(OMVS)
```

NOTE: The OS/390 UNIX kernel userid (OMVS) must have appropriate access to these MVS data sets for HFS data to be accessible.

NOTE: This example assumes that user HFS data sets are named with the user’s userid as the high-level node, followed by OE as the second node. This may not be the convention at all sites. The choice of naming convention impacts the use of the automount facility. Please refer to IBM’s OS/390 UNIX System Services Planning document for detailed information on automount and MapName files.

NOTE: If users are given access authority that allows them to update MVS data sets that contain HFS files, a potential system integrity or security exposure exists. Specifically, permission and extended attribute bit settings might be altered outside of OS/390 UNIX control. Therefore users must not be given update authority to MVS data sets containing HFS files.

2.5.4.4.3 Critical HFS Directories and Files for OS/390 UNIX Components

As stated in *Section 2.5.2.5, OS/390 UNIX HFS Directories and Files* of this document, appropriate UNIX permissions must be maintained on the specified directories and files to maintain system integrity. These permission settings are not maintained in RACF, but must be checked and maintained within the OS/390 UNIX environment.

2.5.4.5 Protecting Sensitive Commands, Utilities, and Language Interfaces

There are various commands, utilities, and language interfaces for OS/390 UNIX that could cause significant damage if used with malicious intent or used incorrectly by unauthorized personnel. This section provides guidelines for managing the security of those items.

2.5.4.5.1 Interactive Commands, Utilities, and Language Interfaces

Section 2.5.2.3, Resource Profiles, and *Section 2.5.2.10, Sensitive TSO/E and Shell Commands and Environment Variable Settings,* of this document provide lists of the general resources and commands that can be protected. If the rules in *Section 2.5.4.1, Defining Resource Profiles and Program Control,* have been implemented, the system has protections in place. The following examples illustrate commands to provide access to the protected resources.

(1) Allow a user to use the OS/390 shell extattr command to set the program control attribute on a file:

```
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) -
ACCESS(READ) ID(privileged-user)
```

(2) Allow a user to use the TSO/E mount command to mount file systems and use the setuid option:

```
PERMIT SUPERUSER.FILESYS.MOUNT CLASS(UNIXPRIV) -
ACCESS(UPDATE) ID(privileged-user)
```

(3) Allow a user to use daemon privileges and to start daemon processes:

```
PERMIT BPX.DAEMON CLASS(FACILITY) -
ACCESS(READ) ID(privileged-user)
```

(4) Allow a user to read and write any HFS file and read, search, and write any HFS directory:

```
PERMIT SUPERUSER.FILESYS CLASS(UNIXPRIV) -
```

```
ACCESS(CONTROL) ID(privileged-user)
```

It must be kept in mind that users with UID(0), users with permission to the BPX.SUPERUSER profile, and users with the RACF TRUSTED or PRIVILEGED attribute bypass any of the security protection implemented by resource profiles.

2.5.4.5.2 OS/390 System Commands

Section 3.3.5.6, OS/390 *System Command Controls*, of this document specifies protections that cover the OS/390 system commands that can specifically impact OS/390 UNIX operations. These commands are F BPXOINIT, SET OMVS, and SETOMVS. It is worth confirming that the following resource protections, appropriate equivalents, or more restrictive controls have been defined:

```
PERMIT MVS.MODIFY.STC.* CLASS(OPERCMDS) ID(operator-group) -
    ACCESS(UPDATE)
PERMIT MVS.SET.* CLASS(OPERCMDS) ID(operator-group) -
    ACCESS(UPDATE)
PERMIT MVS.SETOMVS.OMVS CLASS(OPERCMDS) ID(operator-group) -
    ACCESS(UPDATE)
```

2.5.5 TOP SECRET Implementation for OS/390 UNIX

This section describes the commands needed to implement the security guidelines for OS/390 UNIX under the TOP SECRET ACP. The following task categories are described:

- Resource profiles and program control
- Users and groups
- Started tasks
- Data
- Sensitive commands and utilities

Please note that when the optional CA SAF HFS security is enabled, the use of the SAFHFUSR installation exit can significantly alter the behavior of TOP SECRET with regards to security decisions for OS/390 UNIX. The use of this exit must conform to the procedures documented in *Section 2.1.2.7*, *Access Control Product Exits*, of this document. Please refer to the *Controlling Access to the Hierarchical File System* section of the *CA-TOP SECRET OS/390 Security Cookbook* document for detailed information on implementing the exit.

Throughout this section assumptions are made about the naming convention for some objects. These assumptions are used for ease of discussion and do not reflect absolute standards. These assumptions are as follows:

- ADMIN is an existing department ACID to be used for the ownership of system resources.
- OMVS is the ACID to be used for the OS/390 UNIX kernel.

- User-personal MVS files are prefixed *USERNN*. where *USERNN* is the userid.
- User-personal HFS files are mounted at /u/*usernn* where *usernn* is the userid.

2.5.5.1 Defining Resource Profiles and Program Control

This section lists the commands needed to define the resource profiles required for OS/390 UNIX. These commands are grouped as primary FACILITY class profiles, optional RTLS FACILITY class profiles, SURROGAT class profiles, UNIXPRIV class profiles, optional CA SAF HFS FACILITY class profiles, and program control.

2.5.5.1.1 Primary FACILITY Class Profiles

The following IBMFAC resources need to be protected:

BPX.DAEMON
BPX.DEBUG
BPX.FILEATTR.APF
BPX.FILEATTR.PROGCTL
BPX.JOBNAME
BPX.SAFFASTPATH
BPX.SERVER
BPX.SMF
BPX.STOR.SWAP
BPX.SUPERUSER
BPX.WLMSERVER

The following commands can be used to provide the required protection:

```
TSS ADD(ADMIN) IBMFAC(BPX.)
TSS PERMIT(ALL) IBMFAC(BPX.SAFFASTPATH) ACCESS(NONE)
```

NOTE: The PERMIT command for BPX.SAFFASTPATH must be executed on TOP SECRET systems. If access to BPX.SAFFSTPATH were allowed, OS/390 UNIX would perform permission bit checking internally instead of calling the ACP. On TOP SECRET systems this would bypass any audit trail of violations. In addition, the OS/390 UNIX kernel userid (OMVS is the example in this section) must not have the TOP SECRET NORESCHK privilege. Having that privilege would allow access to BPX.SAFFASTPATH even though the access restriction was in place.

2.5.5.1.2 Optional RTLS FACILITY Class Profiles

If the OS/390 RTLS feature is being used for the C run-time libraries that OS/390 UNIX requires, resource profiles for RTLS may need to be added or updated. The following command must be used to deactivate this checking if RTLS is being used and access checking is not required for any other purpose:

```
TSS ADD(ADMIN) IBMFAC(CSVRTLS.NOSECCONNECT.)
```

Please refer to IBM's *OS/390 MVS Initialization and Tuning Reference* document for detailed information on using RTLS.

2.5.5.1.3 SURROGAT Class Profiles

SURROGAT class profiles are only needed if there are servers (e.g., web server) running in the OS/390 UNIX environment that must be able to act with the security context of a client and that client does not supply a password or other authenticator for the ACP. The following commands are required to authorize a server process (e.g., *server01*) to act as a surrogate for a client user (e.g., *user01*):

If SURROGAT was not previously defined following command can be used to provide the required protection:

```
TSS ADD(ADMIN) SURROGAT(BPX.)
```

Permit access using the following sample:

TSS PERMIT(*server01*) SURROGAT(BPX.SRV.*user01*) ACCESS(READ)

NOTE: The server process (e.g., server01) must also have access to the BPX.SERVER FACILITY class profile.

2.5.5.1.4 UNIXPRIV Class Profiles

The following UNIXPRIV resources need to be protected for superuser granularity support:

```
SUPERUSER.FILESYS
SUPERUSER.FILESYS.CHOWN
SUPERUSER.FILESYS.MOUNT
SUPERUSER.FILESYS.QUIESCE
SUPERUSER.FILESYS.PFSCTL
SUPERUSER.FILESYS.VREGISTER
SUPERUSER.IPC.RMID
SUPERUSER.PROCESS.GETPSENT
SUPERUSER.PROCESS.KILL
SUPERUSER.PROCESS.PTRACE
SUPERUSER.SETPRIORITY
```

The following command can be used to provide the required protection:

```
TSS ADD(ADMIN) UNIXPRIV(SUPERUSE)
```

NOTE: Do not define the CHOWN.UNRESTRICTED UNIXPRIV profile and do **not** select the CHOWNURS(ON) control option. If the profile is defined or the CHOWNURS(ON) option is selected, a basic file ownership protection is defeated. Deviations from this policy must be documented and justified to the IAO.

NOTE: If CA SAF HFS security is enabled (see next section), it takes precedence and all UNIXPRIV class resources are ignored.

- *(ZUSST052: CAT II) The IAO will ensure that the CHOWNURS control option is set to CHOWNURS(OFF).*

2.5.5.1.5 Optional CA SAF HFS FACILITY Class Resources

CA SAF HFS security is an option for TOP SECRET that can provide alternative functions for HFS file control. One set of these functions provides superuser granularity as an alternative to the UNIXPRIV class. The following IBMFAC resources need to be protected:

```
BPX.CAHFS.CHANGE.FILE.ATTRIBUTES
BPX.CAHFS.CHANGE.FILE.AUDIT.FLAGS
BPX.CAHFS.CHANGE.FILE.FORMAT
BPX.CAHFS.CHANGE.FILE.GROUP
BPX.CAHFS.CHANGE.FILE.MODE
BPX.CAHFS.CHANGE.FILE.MODE.EGID
BPX.CAHFS.CHANGE.FILE.MODE.EUID
BPX.CAHFS.CHANGE.FILE.MODE.STICKY
BPX.CAHFS.CHANGE.FILE.OWNER
BPX.CAHFS.CHANGE.FILE.TIME
BPX.CAHFS.CHANGE.PRIORITY
BPX.CAHFS.CREATE.EXTERNAL.LINK
BPX.CAHFS.CREATE.LINK
BPX.CAHFS.CREATE.SYMBOLIC.LINK
BPX.CAHFS.MOUNT
BPX.CAHFS.PTRACE
BPX.CAHFS.SET.PRIORITY
BPX.CAHFS.SET.RLIMIT
BPX.CAHFS.UNMOUNT
```

After the TSS ADD command in *Section 2.5.5.1.1*, *Primary FACILITY Class Profiles*, has been executed, these resources are protected.

NOTE: For these resources to be available, CA SAF HFS security must be enabled by setting the HFSSEC Control Option to ON. This is done permanently through the TOP SECRET Parameter File.

NOTE: When access is enabled to one of the BPX.CAHFS.CHANGE.FILE profiles, the action permitted varies according to the value of the ACCESS operand and the HFSSEC file resource rules. See *Section 2.5.2.3.4, FACILITY Class Resources for CA SAF HFS (ACF2/TOP SECRET)*, for a discussion of the ACCESS values. See *Section 2.5.5.4.3, Critical HFS Directories and Files for OS/390 UNIX Components*, for information on HFSSEC resource rules.

NOTE: If CA SAF HFS security is enabled, it takes precedence and all UNIXPRIV class resources are ignored.

2.5.5.1.6 Program Control

For daemon processes to be permitted to use certain privileged operations, all executables loaded into the daemon's address space must be indicated as program control programs. These executables can come from HFS files or MVS data sets. For executables in HFS files, program control is indicated through the program control extended attribute bit. For executables in MVS data sets, program control is indicated through the ACP. TOP SECRET implements program control through authorized libraries. A library is authorized by being a member of the LPA list, the APF list, or the MVS link list.

Some OS/390 UNIX daemons load programs from the C run-time (SYS1.LE.SCEERUN) and SYS1.LINKLIB libraries. Under the TOP SECRET options for program control, SYS1.LINKLIB is always indicated as a program control library because it is a member of the MVS link list. If the C run-time library (SYS1.LE.SCEERUN) is not defined in the MVS link list, it must be in the APF list.

2.5.5.2 Defining Users and Groups

To authorize a user to access OS/390 UNIX resources, two actions are required:

- A Group ID (GID) number must be added to the OMVS segment of the TOP SECRET GROUP ACID for the user's default group.

- A UNIX Userid (UID) number, home directory (HOME), and shell (PROGRAM) must be added to the OMVS segment of the user's TOP SECRET ACID with the exception of FTP socket applications.

This section lists the commands to accomplish these user and group definitions.

The OMVSUSR and OMVSGRP control options will only be used for FTP socket applications. When coding these options be sure that the restrictions specified in *Section 2.5.6.2, Paragraph 7* of this STIG are followed.

- *(ZUSST050: CAT II) The IAO will ensure that the OMVSUSR/OMVSGRP control options having a value set to OMVSUSR() or OMVSUSR(*NONE*)/ OMVSGRP() or OMVSGRP(*NONE*) are specified for FTP socket applications on non classified systems.*

2.5.5.2.1 Groups

The following commands can be used to create new groups (OMVSGRP, STCOMVS, TTY) and to update an existing group (SYS1) that are required for OS/390 UNIX:

```
    TSS CREATE(OMVSGRP) TYPE(GROUP) NAME(OMVSGRP) -
        DEPT(existing-dept)
    TSS ADD(OMVSGRP) GID(3)
    TSS CREATE(STCOMVS) TYPE(GROUP) NAME(STCOMVS) DEPT(existing-dept)
    TSS ADD(STCOMVS) GID(4)
    TSS CREATE(TTY) TYPE(GROUP) NAME(TTY) DEPT(existing-dept)
    TSS ADD(TTY) GID(1)
    TSS ADD(SYS1) GID(0)
    TSS MODIFY(OMVSTABS)
```

The following commands illustrate the minimum versions of the command (ADD) required for existing groups and the commands (CREATE/ADD) required for new groups when users of those groups use OS/390 UNIX:

```
    TSS ADD(existing-group) GID(gid)
    TSS CREATE(new-group) TYPE(GROUP) NAME(new-group-name) -
        DEPT(existing-dept)
    TSS ADD(new-group) GID(gid)
    TSS MODIFY(OMVSTABS)
```

NOTE: The TSS WHOOWNS GID(*) command lists all existing GIDs.

NOTE: For privileged groups, the value of gid must be between 1 and 99, unique within the site's TOP SECRET database. For unprivileged groups, the value of gid must be between 100 and 16,777,215, unique within the site's TOP SECRET database.

2.5.5.2.2 Privileged Users – Started Tasks and Daemons

The following commands can be used to create the userids that are required for OS/390 UNIX:

```
TSS CREATE(OMVS) TYPE(USER) NAME(OMVS) -
        DEPT(existing-dept) FACILITY(STC) PASSWORD(password,0)
TSS ADD(OMVS) DFLTGRP(STCOMVS) GROUP(STCOMVS,TTY)
TSS ADD(OMVS) UID(0) HOME(/) OMVSPGM(/bin/sh)
TSS PERMIT(OMVS) IBMFAC(BPX.DAEMON) ACCESS(READ)
TSS CREATE(BPXROOT) TYPE(USER) NAME(BPXROOT) -
        DEPT(existing-dept) PASSWORD(password,0)
TSS ADD(BPXROOT) DFLTGRP(OMVSGRP) GROUP(OMVSGRP)
TSS ADD(BPXROOT) UID(0) HOME(/) OMVSPGM(/bin/sh)
TSS CREATE(RMFGAT) TYPE(USER) NAME(RMFGAT) -
        DEPT(existing-dept) FACILITY(STC) PASSWORD(password,0)
TSS ADD(RMFGAT) DFLTGRP(STCOMVS) GROUP(STCOMVS)
TSS ADD(RMFGAT) UID(uid) HOME(/) OMVSPGM(/bin/sh)
TSS CREATE(BPXSTOP) TYPE(USER) NAME(BPXSTOP) -
        DEPT(existing-dept) FACILITY(STC) PASSWORD(password,0)
TSS ADD(BPXSTOP) DFLTGRP(STCOMVS) GROUP(STCOMVS,TTY)
TSS ADD(BPXSTOP) UID(0) HOME(/) OMVSPGM(/bin/sh)
```

If CA SAF HFS security is to be enabled and CAIENF is the started task's userid:

```
TSS ADD(CAIENF) DFLTGRP(STCOMVS) GROUP(STCOMVS)
TSS ADD(CAIENF) UID(uid) HOME(/) OMVSPGM(/bin/sh)
TSS PERMIT(CAIENF) IBMFAC(BPX.SUPERUSER) ACCESS(READ)
TSS PERMIT(CAIENF) IBMFAC(BPX.DAEMON) ACCESS(READ)

TSS MODIFY(OMVSTABS)
```

NOTE: ACIDs with UID(0) must be assigned a password to avoid a potential security exposure through telnet, rlogin, or ftp. All started task ACIDs must have a password and a source of internal reader. Please refer to *Section 3.4.2.3, Started Task Control (STC) Users*, for guidelines on defining started tasks.

NOTE: When UID(0) is not required, the value of uid must be between 1 and 99, unique within the site's TOP SECRET database. The TSS WHOOWNS UID(*) command lists all existing UIDs.

NOTE: The CREATE(RMFGAT) command assumes that this ACID did not already exist. If the ACID already exists, only the ADD(RMFGAT) commands may be required.

NOTE: The assignment of the TOP SECRET NORESCHK attribute is effectively equal to assigning UID(0) to that user in the OS/390 UNIX environment.

2.5.5.2.3 Privileged Users – OS/390 UNIX System Administrators

A user is considered privileged if his userid has access to superuser status by being assigned UID 0, or access to the BPX.SUPERUSER profile, or access to services permitted through the BPX.DAEMON or BPX.SERVER profiles.

The following illustrates the minimum version of the command required for a user when the user needs *full-time superuser authority* for OS/390 UNIX:

```
Existing User:
TSS ADD(existing-user) UID(0) HOME(/u/existing-user) OMVSPGM(/bin/sh)
TSS MODIFY(OMVSTABS)
```

```
New User:
TSS CREATE(new-user) TYPE(USER) NAME(new-user-name) -
DEPT(existing-dept) PASSWORD(password,90,EXP)
TSS ADD(new-user) DFLTGRP(existing-group) GROUP(existing-group)
TSS ADD(new-user) UID(0) HOME(/u/new-user) OMVSPGM(/bin/sh)
TSS MODIFY(OMVSTABS)
```

NOTE: The assignment of UID(0) to any user must follow the guidelines in *Section 2.5.2.6.1, Privileged Users and Special Groups*, of this document. All userids with UID(0) must be documented to the IAO.

NOTE: ACIDs with UID(0) must be assigned a password to avoid a potential security exposure through telnet, rlogin, or ftp.

NOTE: The assignment of the TOP SECRET NORESCHK attribute is effectively equal to assigning UID(0) to that user in the OS/390 UNIX environment.

The following illustrates the minimum version of the commands (CREATE/ADD/PERMIT) required for a user when the user needs *the ability to switch to superuser authority* for OS/390 UNIX:

```
New User:
TSS CREATE(new-user) TYPE(USER) NAME(new-user-name) -
    DEPT(existing-dept) PASSWORD(password,90,EXP)

New and Existing User:
TSS ADD(new-user/existing-user) DFLTGRP(existing-group) GROUP(existing-group)
TSS ADD(new-user/existing-user) UID(uid) HOME(/u/new-user or /u/existing-user) –
    OMVSPGM(/bin/sh)
TSS PERMIT(new-user/existing-user) IBMFAC(BPX.SUPERUSER) ACCESS(READ)
TSS MODIFY(OMVSTABS)
```

NOTE: The value of UID must be between 100 and 16,777,215, unique within the site's TOP SECRET database. The TSS WHOOWNS UID(*) command lists all existing UIDs.

NOTE: The DFLTGRP for the user must have a valid GID value.

2.5.5.2.4 Unprivileged Users

The following illustrates the minimum version of the commands (CREATE / ADD) required for a user when the user needs *unprivileged access* for OS/390 UNIX:

```
New User:
TSS CREATE(new-user) TYPE(USER) NAME(new-user-name) -
    DEPT(existing-dept) PASSWORD(password,90,EXP)

New and Existing User:
```

```
TSS ADD(new-user/existing-user) DFLTGRP(existing-group) GROUP(existing-group)
TSS ADD(new-user/existing-user) UID(uid) HOME(/u/new-user or /u/existing-user) –
    OMVSPGM(/bin/sh)
TSS MODIFY(OMVSTABS)
```

NOTE: The value of UID must be between 100 and 16,777,215, unique within the site's TOP SECRET database. The TSS WHOOWNS UID(*) command lists all existing UIDs.

NOTE: The DFLTGRP for the user must have a valid GID value.

2.5.5.3 Defining Started Tasks

The following commands can be used to define the started tasks that are required for OS/390 UNIX:

```
TSS ADD(STC) PROCNAME(OMVS) ACID(OMVS)
TSS ADD(STC) PROCNAME(BPXOINIT) ACID(OMVS)
TSS ADD(STC) PROCNAME(BPXAS) ACID(OMVS)
TSS ADD(STC) PROCNAME(BPXSTOP) ACID(BPXSTOP)
TSS ADD(STC) PROCNAME(RMFGAT) ACID(RMFGAT)
```

NOTE: If the RMFGAT started task had been defined previously, the definition must be checked to ensure that it uses the RMFGAT ACID. This is required so that the appropriate UID is used when the RMFGAT task accesses OS/390 UNIX data.

2.5.5.4 Protecting Data

In order to preserve system integrity, data must be protected from unauthorized or inadvertent modification. This section lists the commands that are used to protect the system components and data that are essential to OS/390 UNIX. Guidelines for protecting user HFS data are also discussed.

2.5.5.4.1 MVS Data Sets for OS/390 UNIX Components

The following command can be used to provide the required access control for the MVS data sets that contain OS/390 UNIX components:

```
TSS PERMIT(ALL) DSN(SYS1.SBPX) FAC(TSO) ACCESS(READ)
```

[- The SYS1.SBPX data sets are used by TSO/E users of OS/390 UNIX; an ACCESS(READ) PERMIT is acceptable for this resource.]

2.5.5.4.2 MVS Data Sets Containing OS/390 Hierarchical File Systems

The following commands can be used to provide the required access control for the MVS data sets that contain Hierarchical File Systems containing OS/390 UNIX components:

```
TSS PERMIT(OMVS) DSN(SYS1.OE.) ACCESS(UPDATE)
TSS PERMIT(OMVS) DSN(SYS3.OE.) ACCESS(UPDATE)
```

NOTE: The OS/390 UNIX kernel userid (OMVS) must have appropriate access to these MVS data sets for HFS data to be accessible.

NOTE: This example assumes that system HFS data sets are named with SYS1 or SYS3 as the high-level node, followed by OE as the second node. This may not be the convention at all sites.

If individual user HFS data sets are being allocated, rules must be established to allow OS/390 UNIX to access the data sets. The following illustrates the commands that can be used to provide the required access control for each MVS data set that contains users' Hierarchical File Systems:

TSS PERMIT(*user*) DSN(*user*.OE.) ACCESS(READ)
TSS PERMIT(OMVS) DSN(*user*.OE.) ACCESS(UPDATE)

NOTE: The OS/390 UNIX kernel userid (OMVS) must have appropriate access to these MVS data sets for HFS data to be accessible.

NOTE: This example assumes that user HFS data sets are named with the user's userid as the high-level node, followed by OE as the second node. This may not be the convention at all sites. The choice of naming convention impacts the use of the automount facility. Please refer to IBM's OS/390 UNIX System Services Planning document for detailed information on automount and MapName files.

NOTE: If users are given access authority that allows them to update MVS data sets that contain HFS files, a potential system integrity or security exposure exists. Specifically, permission and extended attribute bit settings might be altered outside of OS/390 UNIX control. Therefore users must not be given update authority to MVS data sets containing HFS files.

2.5.5.4.3 Critical HFS Directories and Files for OS/390 UNIX Components

As stated in the *OS/390 UNIX HFS Directories and Files* section of this document, appropriate UNIX permissions must be maintained on the specified directories and files to maintain system integrity. These permission settings are not maintained in TOP SECRET, but must be checked and maintained within the OS/390 UNIX environment.

If the optional CA SAF HFS security is enabled, UNIX permission bit checking is bypassed and access profiles must be created. As distributed by Computer Associates, TOP SECRET does not have default protection enabled for the HFSSEC resource class. As a result, unauthorized users are able to create directories and/or files at the root (/) level. It will be required that default protection be specified for the HFSSEC resource class to prevent unauthorized access to objects at the root level. The following command can be used to enable default protection for the HFSSEC resource class by adding the DEFPROT attribute to the RDT entry:

```
TSS REPLACE(RDT) RESCLASS(HFSSEC) ATTR(DEFPROT)
```

- *(ZUSST060: CAT I) The IAO will ensure that the HFSSEC resource class has the attribute DEFPROT.*

The following commands form a base to be used to secure the critical HFS directories and files and to allow for separate control of user HFS directories and files:

```
TSS ADD(ADMIN) HFSSEC(ROOT)
TSS PERMIT(ALL) HFSSEC(ROOT) ACCESS(READ)

TSS ADD(ADMIN) HFSSEC(/BIN)
TSS PERMIT(ALL) HFSSEC(/BIN) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/BIN.) ACCESS(READ,EXEC)

TSS ADD(ADMIN) HFSSEC(/DEV)
TSS PERMIT(ALL) HFSSEC(/DEV) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/DEV.) ACCESS(READ,EXEC)
TSS PERMIT(ALL) HFSSEC(/DEV.CONSOLE) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/DEV.NULL) ACCESS(READ,UPDATE,EXEC)

TSS ADD(ADMIN) HFSSEC(/ETC)
TSS PERMIT(ALL) HFSSEC(/ETC) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/ETC.) ACCESS(READ,EXEC)
TSS PERMIT(ALL) HFSSEC(/ETC.AUTO$MASTER) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/ETC.INETD$CONF) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/ETC.INIT$OPTIONS) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/ETC.RESOLV$CONF) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/ETC.SERVICES) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/ETC.STEPLIB) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/ETC.TABLENAME) ACCESS(NONE)

TSS ADD(ADMIN) HFSSEC(/LIB)
TSS PERMIT(ALL) HFSSEC(/LIB) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/LIB.) ACCESS(READ,EXEC)

TSS ADD(ADMIN) HFSSEC(/SAMPLES)
TSS PERMIT(ALL) HFSSEC(/SAMPLES) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/SAMPLES.) ACCESS(READ,EXEC)
TSS ADD(ADMIN) HFSSEC(/TMP)
TSS PERMIT(ALL) HFSSEC(/TMP) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/TMP.) ACCESS(ALL)

TSS ADD(ADMIN) HFSSEC(/U)
TSS PERMIT(ALL) HFSSEC(/U) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/U.%) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/U.%.) ACCESS(ALL)

TSS ADD(ADMIN) HFSSEC(/USR)
TSS PERMIT(ALL) HFSSEC(/USR) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/USR.) ACCESS(READ,EXEC)
TSS PERMIT(ALL) HFSSEC(/USR.LIB.CRON.AT$ALLOW) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/USR.LIB.CRON.AT$DENY) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/USR.LIB.CRON.CRON$ALLOW) ACCESS(NONE)
TSS PERMIT(ALL) HFSSEC(/USR.LIB.CRON.CRON$DENY) ACCESS(NONE)

TSS ADD(ADMIN) HFSSEC(/VAR)
TSS PERMIT(ALL) HFSSEC(/VAR) ACCESS(READ)
TSS PERMIT(ALL) HFSSEC(/VAR.) ACCESS(READ,EXEC)
```

- *(ZUSS0080: CAT II) The IAO will ensure that the HFSSEC resource access is restricted to appropriate personnel with appropriate logging based on bit and audit settings.*

NOTE: For these profiles to be effective, CA SAF HFS security must be enabled by setting the HFSSEC Control Option to ON. This is done permanently through the TOP SECRET Parameter File.

NOTE: Even when CA SAF HFS security is enabled, UNIX permission bits must be maintained on the critical HFS directories and files.

NOTE: Because of path translation (noted next), all paths are represented in uppercase in rules.

NOTE: HFS paths are folded to uppercase and, if necessary, truncated to 255 characters. All slash characters after the first are translated to periods and other special characters (including periods, asterisks, dashes, plus signs, blanks, and quote marks) are translated to the dollar sign ($) character.

NOTE: The /U.%. commands assume that user directories are mounted at the /u mount point and use the userid as the next level. This may not be the convention at all sites. The choice of naming convention impacts TOP SECRET command coding and the use of the automount facility. Please refer to IBM's OS/390 UNIX System Services Planning document for detailed information on automount and MapName files.

NOTE: The SAFHFUSR exit has the ability to alter path translation, indicate user path processing, and indicate no validation for user owned files. Please refer to the

Controlling Access to the Hierarchical File System section of the CA-TOP SECRET OS/390 Security Cookbook document for detailed information on implementing the exit.

2.5.5.5 Protecting Sensitive Commands, Utilities, and Language Interfaces

There are various commands, utilities, and language interfaces for OS/390 UNIX that could cause significant damage if used with malicious intent or used incorrectly by unauthorized personnel. This section provides guidelines for managing the security of those items.

2.5.5.5.1 Interactive Commands, Utilities, and Language Interfaces

Section 2.5.2.3, Resource Profiles, and *Section 2.5.2.10, Sensitive TSO/E and Shell Commands and Environment Variable Settings,* of this document provide lists of the general resources and commands that can be protected. If the rules in *Section 2.5.5.1, Defining Resource Profiles and Program Control,* have been implemented, the system has protections in place. The following examples illustrate commands to provide access to the protected resources:

(1) Allow a user to use the OS/390 shell extattr command to set the program control attribute on a file:

TSS PERMIT(*privileged-user*) IBMFAC(BPX.FILEATTR.PROGCTL) -
ACCESS(READ)

(2) Allow a user to use the TSO/E mount command to mount file systems and use the setuid option:

When CA SAF HFS security is not enabled:

TSS PERMIT(*privileged-user*) UNIXPRIV(SUPERUSER.FILESYS.MOUNT) -
ACCESS(READ,UPDATE)

When CA SAF HFS security is enabled:

TSS PERMIT(*privileged-user*) IBMFAC(BPX.CAHFS.MOUNT) -
ACCESS(READ)
TSS PERMIT(*privileged-user*) IBMFAC(BPX.CAHFS.UNMOUNT) -
ACCESS(READ)

(3) Allow a user to use daemon privileges and to start daemon processes:

TSS PERMIT(*privileged-user*) IBMFAC(BPX.DAEMON) -
ACCESS(READ)

(4) Allow a user to read and write any HFS file and read, search, and write any HFS directory:

When CA SAF HFS security is not enabled:

```
TSS PERMIT(privileged-user) UNIXPRIV(SUPERUSER.FILESYS) -
ACCESS(CONTROL)
```

When CA SAF HFS security is enabled:

```
TSS PERMIT(privileged-user) IBMFAC(BPX.SUPERUSER) -
ACCESS(READ)
```

(There is no CA SAF HFS privilege that maps to the UNIXPRIV FILESYS privilege. In this case the user receives full superuser authority.)

It must be kept in mind that users with UID(0), users with permission to the BPX.SUPERUSER profile, and users with the TOP SECRET NORESCHK attribute bypass any of the security protection implemented by resource profiles.

2.5.5.5.2 OS/390 System Commands

Section 3.4.5.6, OS/390 System Command Controls, of this document specifies protections that cover the OS/390 system commands that can specifically impact OS/390 UNIX operations. These commands are F BPXOINIT, SET OMVS, and SETOMVS. It is worth confirming that the following resource protections, appropriate equivalents, or more restrictive controls have been defined:

```
TSS PERMIT(operator-group) OPERCMDS(MVS.MODIFY.STC) -
    ACCESS(UPDATE) ACTION(AUDIT)
TSS PERMIT(operator-group) OPERCMDS(MVS.SET) -
    ACCESS(UPDATE) ACTION(AUDIT)
TSS PERMIT(operator-group) OPERCMDS(MVS.SETOMVS.OMVS) -
    ACCESS(UPDATE) ACTION(AUDIT)
```

3. ACCESS CONTROL PRODUCT IMPLEMENTATION

3.1 General Considerations

The ACP is the primary mechanism that controls access to data and resources in OS/390 systems. Each ACP in use on the DOD platforms provides the flexibility to tailor the implementation to meet the needs of the local installation.

Many different implementations of the various ACPs exist. These different implementations meet the needs of each local installation, but make it difficult to coordinate and control the DOD Enterprise.

The installation and implementation of each ACP should be standardized across all DOD processing environments. STIG recommended implementation criteria are specified in the individual ACP installation sections of this document.

All deviations are to be specifically noted, with justification and approval documentation, in the system security plan and the accreditation package submitted to the Designated Approving Authority (DAA).

To provide full compliance with the security support required by *DOD Directive 8500.1*, control all products within the operating system using the ACP. Use the following guidance in the acquisition of products to ensure that security-related issues are adequately addressed:

(1) Products are to be on the National Information Assurance Partnership (NIAP) - Common Criteria Evaluation and Validation Scheme (CCEVS) Validated Products List before procurement and implementation.

(2) At a minimum, evaluate products for sensitive functions and implement controls to protect these functions.

(3) Restrict all data sets associated with a product to the access levels necessary for support and operation based upon the requirements. Only those authorized personnel who require the authority to modify or maintain the product are to have *update* and *alter* access.

Many products require special security considerations. Enforce the following considerations relating to compatibility and interfacing with the IBM System Authorization Facility (SAF):

(1) Protect Commercial-Off-The-Shelf (COTS) products and associated data sets within the operating system using the ACP. Ensure that all COTS products being procured have, and utilize, the SAF interface to the ACP.

(2) Secure Government-Off-The-Shelf (GOTS) products and newly developed applications, along with associated data sets, using the ACP. Whenever possible, develop applications using the SAF interface. Safeguards enforced by the ACP are not to be duplicated by security mechanisms implemented within an application. Limit developed internal security mechanisms to those functions that augment the safeguards present in the ACP.

(3) Internal Product Security Controls (IPSCs) are security mechanisms internal to COTS products and GOTS applications. Only use IPSCs when existing products or applications do not interface to the ACP through SAF, or to augment the protections provided by the existing interface. Reconfigure products using IPSCs, which are capable of taking advantage of the SAF interface, to take proper advantage of the SAF interface.

Whenever IPSCs are being used, develop and maintain security documentation. The documentation is to include descriptions of the IPSCs, the configuration, and the policy being enforced. The IAO is to maintain the documentation and perform the administration of IPSCs where practical.

(4) Modify all GOTS products and applications (if using ACP-specific interfaces) to interface with the ACP via standard SAF calls.

All applications are to eventually migrate from IPSCs to using the ACP. If this is unreasonable for any given application, the application is to be eventually phased out.

3.1.1 Standard Global Options

Each ACP provides the capability for customization using global ACP configuration and processing options. These global options provide the flexibility to tailor the configuration and processing of the ACP to the needs of the local operating environment. These options also can pose the danger of compromising the operational environment when misused or when not properly applied.

In an organization as large as DOD, the additional complication of diversity exists. Many different applications of the global options exist. These different applications meet the needs of each local installation, but make it difficult to manage the organizational computing base as a whole. The task of optimizing the processing load of the enterprise across the myriad platforms becomes virtually impossible.

For the above reasons, and to mitigate the above risks and difficulties, all DOD processing environments are to implement the STIG required global options for each ACP installed. The STIG required options are specified in the individual Access Control Product installation sections of this document. The options specified are STIG requirements and each site can choose to be more restrictive.

3.1.2 Userid Controls

Requires that each system user is uniquely identified to the operating environment, and that access to resources is limited to those needed to perform the function. In this case, a user is defined as either an individual accessing a computer resource, or as a task executing on the system that requires access to a resource. On OS/390 systems a user is identified by means of a unique userid. This STIG requires that audit data record the identity of the user, time of access, interaction with the system, and sensitive functions that might permit a user or program to modify, bypass, or negate security safeguards.

It then follows that any userid (user) on the system must be associated with only one individual. It also follows, however, that any given individual may be assigned responsibility for multiple userids on a given system, depending on functional responsibilities, to ensure task segregation. The following sections discuss the requirements for each type of user and the elements that are to be considered in defining user access.

In addition to associating userids with individuals and tasks, MVS can associate userids with system facilities (e.g., consoles and remote workstations). Such userids are to be defined with only the access rights required for their intended use.

3.1.2.1 Interactive Users

To achieve compliance with the criteria mandated for MAC II Sensitive, all personnel are required to identify themselves to the system before access to resources is granted. A means to authenticate the user's identity (e.g., passwords) are to be used. Each user of resources is to be defined using ACP facilities to control I&A.

The recommended process used in this STIG consists of a userid and a password, which together enable the user to access the system. The IAO controls access to computing resources and adds additional functions to a userid.

DOD, in conjunction with the National Security Agency, is implementing the integrated use of PKI-based technology to perform I&A validation and other secure services. The technology is being implemented through the development of the secure web server that relies on DES-based encryption services. Until an approved PKI-based authentication solution is deployed, it is the requirement of this STIG that an extended authentication process for highly privileged users using the SecurID card along with Authenticator software be implemented. Refer to *Section 6.3, NC-PASS Authenticator*, for further information on the NC-PASS controls.

3.1.2.2 Batch Users

A major user of system resources is a batch job. A batch job is essentially a stand-alone task submitted to JES2/JES3 for processing. Batch jobs can be submitted in several ways. A user accessing the system through TSO or a similar facility may have the capability to submit batch jobs for execution. Jobs may be submitted by a scheduling system to process an orderly work flow without user intervention. Each of these batch jobs should be identified to the system to designate the resources that should be available to the job. Each ACP allows the association of a userid with the job stream (e.g., a USER= parameter on the job card or a special JCL card placed in the job stream).

Batch jobs submitted to the operating system by a user (e.g., TSO) should inherit the userid of the submitter. This identifies the batch job with the user for accessing resources. An alternative is to allow users to add userid and password information to the job stream. This allows users to submit special batch jobs that require access to system resources beyond those authorized for their normal userids.

Userid and password inheritance is the STIG required method for batch jobs submitted by an interactive user. If this is not possible, then the alternative of placing the userid and password in the JCL may be used. However, the JCL will be tightly controlled and access to it severely limited. Such use will be fully documented. The IAO will maintain the written request, justification, and authorization.

- *(ACP00245: CAT II) The IAO will ensure that the data sets containing userids and passwords are restricted to authorized users only and documentation justifying access authorization is maintained.*

Jobs submitted to the operating system via a scheduling facility or via other automated means should be identified to the system. This may be accomplished by using several different alternatives.

The first alternative is to associate a userid to a job stream without a password. This is commonly called a restricted or protected userid process. For any userid that does not require a password, additional enforcement of the SUBMIT process is required. This can be accomplished by identifying or limiting the source of submission for the job without a password. For example, creating an ACP definition stating that the userid can be used without a password, if program **X** is the submitting program.

The second alternative involves the use of a password card in the job stream, as well as the userid. Since the password cannot be stored in a clear text form, an alternate means would need to be devised for password management, making this process less desirable.

Under no circumstance are user jobs to inherit the authority associated with a system-level address space (e.g., job scheduler, job submission started task, etc.). Batch jobs submitted via such processes are only allowed to access required resources.

3.1.2.3 Started Task Control (STC) Users

Several jobs run as started tasks within the system. Due to the conceptual design of MVS, started tasks do not necessarily have a userid or password associated with them. A started task is a procedure started from the operating system console with the MVS START command. To qualify as a started task, the procedure should be found in either of the following locations:

- A procedure library concatenated to the appropriate PROC*xx* Data Definition (DD) statement in the JCL for the JES2 started task, or

- SYS1.PROCLIB for started tasks initiated through the Master JCL (or other data sets concatenated to the PROCLIB DD statement)

The actual JES2 PROC*xx* to be used for STCs, normally PROC00, is specified in the JES2 initialization parameters. Each ACP offers facilities to assign a userid to a started task so that resources can be limited for these tasks. Without controls over this process, started tasks have the authority to access any information in the operating system.

Use the following guidance to ensure protection of started tasks:

(1) Every started task should be uniquely identified to the ACP by the IAO. Software Support personnel should notify the IAO so that a unique userid can be assigned to any new started task added to the system. No default userids are to be assigned to started tasks otherwise not identified.

(2) Members that are not started tasks may exist in the concatenated libraries. These typically are procedures intended for general use as batch processes, or for use by TSO users. To prevent their improper execution, these members should **not** be defined to the ACP as started tasks.

(3) The IAO should administer resource-level protection (i.e., data set controls) so that access is available only for those resources required by the started task.

(4) The IAO should maintain data set controls for all proclib data sets concatenated to the JES PROC00 DD statement. Only authorized users should have the ability to update libraries in the concatenation.

(5) Certain started tasks perform critical operating system-related functions. The site can secure these started tasks in one of two ways:

(a) By analyzing an STC's access requirements and granting the requisite accesses.

(b) By considering these started tasks as trusted for the purpose of data set and resource access requests.

For trusted STCs, all access requests may be honored, regardless of the permissions granted in access rules and profiles.

While the actual list may vary based on local site requirements and software configuration, the following is an approved list of started tasks that may be considered trusted started procedures:

The Job Entry Subsystem
(JES2 or JES3)
The MOUNT procedure
(IEEVMPCR)
The PS procedure(s)
APPC
ASCH
CATALOG

CONSOLE DFHSM
DUMPSRV
LLA
SMF
SMS
SVWTCPIP
VLF
VTAM

The site may exclude any of the above from the local list of **trusted** started tasks based on local requirements. However, the addition of other started tasks to the list requires the approval of the site DAA.

3.1.2.4 Network Users

This section addresses jobs being submitted using network controls. Two distinctive processes are currently being used – Remote Job Entry (RJE) and Network Job Entry (NJE).

RJE is designed for job streams being submitted from a workstation. Generally this involves submitting jobs via card decks or JCL stored on a hard disk device. Two means are available to identify the job streams being submitted. One is to submit batch jobs with USER and PASSWORD cards to identify the source of submission. The other involves restricting the use of certain batch userids, only allowing submission from a particular origin (such as specific RJE devices). Where possible, the second is the preferred method of processing since passwords do not require any individual maintenance and the work can be restricted only from that particular origin.

NJE is designed to provide data flow between multiple OS/390 platforms. A remote NJE connection has multiple users who can submit work via an established NJE connection. Thus, the requirement exists for each user to be properly identified. Default userids for NJE connections are not be used since individual user access would not be accounted for and properly controlled.

3.1.2.5 Special Storage Management Users

Every data center uses special job streams for maintenance processing that require more authority than any one user may be granted. Examples include job streams performing volume backups, data set archiving, and restoration processes. Each ACP allows special privileges that may be assigned to a userid. Userids used for batch jobs that perform such functions are processed using special privileged userids to avoid the assignment of all resources to a single userid.

Refer to *Section 3.1.4, Special Privilege Access,* for further information. Also, refer to the specific Access Control Product section on the methodology to be employed for this process.

3.1.2.6 Emergency Userids

In emergencies, the access necessary to perform a function may not be available to recovery personnel. To handle such emergencies, super IDs or firecall procedures are to be made available.

Use the following rules and conditions to handle these userids:

(1) One class of userids are to exist to perform all operating system functions except ACP administration. These super IDs may be released according to STIG recommended policy to effect repairs of the operating system in emergencies.

(2) A second class of super IDs is to be maintained to allow the functions associated only with ACP administration. These IDs are to only be released at the direction of the IAO.

(3) Normally both super IDs are not be released to the same individual concurrently, although approved exceptions to this rule can be made. This constraint effects a check and balance process for recovery situations requiring both forms of authorization.

(4) The super IDs are to be implemented with logging to provide an audit trail of their activities.

(5) Both classes of super IDs are to be maintained in both the ACP and SYS1.UADS to ensure they are available in the event that the ACP is not functional.

(6) Each super ID is to have distinct, different passwords in SYS1.UADS and in the ACP, and the site is to establish procedures to ensure that the passwords differ. The password for any ID in SYS1.UADS is never to match the password for the same ID in the ACP.

(7) Documented procedures are to be established to provide a mechanism for the use of the IDs. Their release for use is to be logged, and the log is to be maintained by the IAO. When a super ID is released for use, its password is to be reset by the IAO within 12 hours after it is no longer needed for problem resolution.

3.1.2.7 FTP Userids

Many production applications in the site environment are structured to use File Transfer Protocol (FTP) mechanisms to transfer data between sites. These FTP mechanisms require the use of an userid. However, in many cases, it is impractical to change the password associated with an FTP userid. It is acceptable to establish FTP userids and the associated logon mechanisms using non-expiring passwords, as long as mitigating controls are in place and the customer has been apprised of, and has officially accepted, the risks. The following sections discuss the potential inherent risks and the required mitigating controls when using FTP with userids having non-expiring passwords.

3.1.2.7.1 Risks

The following potential risks are inherent in the use of FTP userids:

(1) Maintaining the MVS userid and password on a remote system (e.g., another OS/390 host or a remote UNIX system) increases the potential for their compromise.

(2) The use of FTP, by its nature, requires that the userid and password be transmitted to the remote host system in clear text across unsecured, non-SNA communications lines. This increases the potential for their compromise by various means (e.g., SNIFFER programs).

(3) The compromise of the userid and password could remain undetected for a long period of time.

(4) The use of non-expiring passwords increases the window of exposure to the OS/390 system in the event that the userid and password are compromised. Because the password never expires, the exploitation could persist for an unlimited amount of time.

(5) The use of FTP, by its nature, involves the transmission of application data to and from the OS/390 host system in clear text across unsecured, non-SNA communications lines. This increases the potential for their compromise by various means (e.g., SNIFFER programs).

(6) The compromise of application data could remain undetected for a long period of time.

3.1.2.7.2 Mitigating Controls

To reduce the risk of compromising integrity as a result of using non-expiring passwords for FTP connectivity to OS/390 systems, implement the following mitigating controls. For the sites running IBM Communications Server FTP, refer to *Section 4.4.4, File Transfer Protocol (FTP) Server*, and *Section 4.4.5, File Transfer Protocol (FTP) Client*, for further information.

(1) Do not use Anonymous FTP on DOD systems. It is to be disabled.

(2) Secure all scripts and/or data files located on the remote system(s) that contain the MVS FTP userid and/or password (e.g., another OS/390 host or a remote UNIX system). Restrict access to these files to those individuals responsible for the application connectivity and who have a legitimate requirement to know the userid and password.

(3) Restrict access to the OS/390 host FTP software files and data sets to the MVS staff responsible for installation and maintenance, in accordance with the requirements for system software security specified in *Section 3.1.5.1, Data Set Controls*.

(4) Restrict access to the OS/390 host FTP software log to the staff responsible for the daily processing of FTP transactions.

(5) Dedicate all MVS userids with non-expiring passwords used for FTP connectivity solely to that purpose. Do not use them for any other processing (e.g., TSO sessions, CICS, batch) on the subject OS/390 platform or on any other platform (e.g., another OS/390 host or a remote UNIX system).

(6) Dedicate all MVS userids used for FTP connectivity for a given application solely to that application. Do not use them for any other application processing (e.g., another CDA's FTP application). The userid associated with an FTP is not to be the userid logged on to the MVS system (e.g., the TSO session).

(7) Limit data set accesses for each FTP userid to the absolutely minimum level necessary for the accomplishment of the required functions.

(a) Limit *write* access only to those data sets into which data is to be transferred.

(b) Limit *read* access only to those data sets from which data is to be transferred.

(c) Limit *execute* access only to those execution libraries required for the accomplishment of the data transfers.

(8) For each MVS FTP application, ensure that all MVS userids used for FTP connectivity are fully audited. Both the IAO and the owner of the data residing on the OS/390 platform are to review the reports of their use on a daily basis.

(9) Each customer having an application that requires the use of FTP with non-expiring passwords is to be fully informed of the risks. Each customer is to officially acknowledge accepting the risks using the *Acknowledgement of Risk* document. AORL has been removed from this STIG and will be maintained in a separate document to be disclosed at a later date. The IAO/IAM is to retain and maintain each *Acknowledgement of Risk* document for reference and auditing purposes.

(10) A procedure is to be developed and implemented, in coordination with the customer, to manually change the password of FTP application userids at least twice a year or when an administrator with knowledge of the password leaves.

(11) FTP and telnet from outside the enclave into the enclave is not permitted, unless encrypted and the following conditions apply (IA Control EBRU-1 & ECCT-1): FTP and telnet are acceptable from outside the enclave through a remote access Virtual Private Network (VPN). The connection will terminate outside the firewall as to not bypass the security architecture. The connection will be proxied at the firewall or via an FTP/telnet proxy. FTP and telnet are acceptable via a site-to-site VPN between trusted enclaves; however, an Acceptance of Risk letter (AORL) must already be in place for the tunnel. FTP and telnet are acceptable within distributed enclaves, if required, as long as the traffic is physically or logically segregated from normal traffic using a method supported by the network technology to create a virtual connection (e.g., VLAN, VPN, LANE, MPLS, IPSec tunnels).

(12) For FTP software that requires the MVS FTP userid to have access to TSO (e.g., KNET), implement the following additional mitigating controls:

(a) The TSO LOGON procedure assigned to the FTP userid is to contain only those libraries (e.g., STEPLIB, panel, clist, etc.) required for the accomplishment of the data transfer.

(b) Where possible, restrict the FTP userids only to those TSO commands required for the accomplishment of the required functions.

Some FTP products, such as KNET, operate as two-party FTP products. This means that for outbound FTP transfers from the OS/390 host, the products only require a userid and password for the remote system.

Other FTP products, such as OPENLINK from Network Solutions, are three-party FTP products. These products require the userid and password for both sending and receiving sites. For example, OPENLINK requires both sets to be imbedded as part of the SYSIN data stream, and performs security checking explicitly with calls to the ACP. For these products it is even more critical that the SYSIN input file for controlling a transfer be strictly controlled and protected.

(c) Severely restrict data set accesses for the SYSIN file as follows:

1) Limit *read* access to the owner of the JCL stream, the FTP product (e.g., OPENLINK), and the job scheduling package (only if one is used to schedule FTP transfers).

2) Limit *write* access to the owner of the JCL stream, and to the production control group responsible for the day-to-day operations of the job scheduling package (only if one is used to schedule FTP transfers).

3.1.2.8 MCS Console Userids

MCS consoles allow operators to enter MVS and JES system commands. In the original design of MVS, MVS consoles did not have userids or passwords associated with them. The extended MCS support added in MVS/ESA SP, Version 4, Release 3, allows the installation to control the use of MCS consoles through the ACP. This support allows an installation to grant minimal (even null) command privileges to the console itself and to require operators to log on prior to entering other commands.

Use the following recommendations to ensure protection of MCS consoles:

(1) Assign each MCS console a unique name in the CONSOL*xx* member of SYS1.PARMLIB. (Refer to *Section 2.1.2.12, MCS Consoles*.)

(2) Define each console to the ACP as a user. The console userids are to only be usable for the automatic logon of the MCS consoles (if permitted), and are not to include any privileges or profile segments needed only for batch or interactive use.

Refer to *Section 3.1.5.5, MCS Console Controls*, and *Section 3.1.5.6, OS/390 System Command Controls*, for the resources that are permitted to each MCS console userid.

3.1.2.9 OS/390 System Operator Userids

The extended MCS support added in MVS/ESA SP, Version 4, Release 3, allows an installation to control the use of MCS consoles through the ACP. This enables an installation to place tighter controls on operator console usage. To control entry of OS/390 system commands in MVS/ESA, Release 4.3 and later, apply the following recommendations when implementing security:

(1) Define each operator to the ACP as a user.

(2) At the discretion of the IAO, the operators' userids may be given privileges and profiles beyond those needed to log on to an MCS console (e.g., TSO segments).

(3) Refer to *Section 3.1.5.5, MCS Console Controls*, and *Section 3.1.5.6, OS/390 System Command Controls*, for the resources that are permitted to each operator userid.

3.1.3 Password Controls

Each ACP allows the specification of a password. Certain guidelines are to be followed for password settings to ensure I&A criteria are in accordance with a site DAA approved secure format.

3.1.3.1 Password Guidelines

Industry analysis has shown that people are more likely to remember their passwords and not write them down if they are allowed to create their own. The most effective means is to allow users to select their own passwords, and to force certain guidelines regarding the composition of the passwords. Users are to change their passwords regularly. This reduces the possibility of a user’s password being acquired, and possibly used, by someone else.

After three consecutive password failures, the userid is to be suspended until reset by the IAO, TASO (Terminal Area Security Officer), or other authorized personnel.

In accordance with *DODI 8500.2* for DOD information systems processing sensitive information and above, and *CJCSM 6510.01*, the following recommendations concerning password requirements are mandatory and apply equally to both classified and unclassified systems:

(1) Passwords are to be eight (8) characters in length.

(2) Passwords are to be a mix of alphabetic, numeric, and special characters, including at least one of each. Special characters include the national characters (i.e., **@**, **#**, and **$**) and other non-alphabetic and non-numeric characters typically found on a keyboard. However at this time the three ACPs only support the national characters.

The following set represents the complete list of characters currently supported by the three ACPs:

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@#$

NOTE: Lower case alphabetic characters are not supported by the three ACPs.

(3) Each character of the password is to be unique, prohibiting the use of repeating characters.

(4) Passwords are to contain no consecutive characters (e.g., 12, AB).

(5) Passwords are not to include the user’s name, telephone number, userid, or any standard dictionary word.

(6) Users are to be required to change their password every 90 days at a minimum. Users are permitted to manage and change their own passwords.

(7) Passwords are not be changed more than once every 24 hours without the intervention of the IAO.

(8) Users are not to be permitted to reuse a password assigned within the last ten password changes.

(9) The password file are to be stored in encrypted form.

Password requirements are to be enforced by standard security product controls where possible. Exits are only to be used where the requirements cannot be enforced by standard security product controls. (Refer to *Section 3.1.3.2, Password Exit Processing*, for further information.)

NOTE: Adherence is required when the software has the capability to enforce. Otherwise the password policies not enforced by the software are to be documented in the site Security Features Users Guide.

3.1.3.2 Password Exit Processing

All the ACPs provide exit facilities and/or installation options that allow customers flexibility in defining password structures. These options and exits may be used to allow users the capability to specify the same password in multiple locations, according to the guidelines in *Section 3.1.3.1,* *Password Guidelines*. This process does not change a password in multiple environments where a user has access, but allows a user to specify the same password in multiple platforms.

As stated in *Section 3.1.3.1, Password Guidelines*, the site may, at its own discretion, extend the capabilities of the ACP by way of an exit to enforce any or all of the password requirements not already enforced by the ACP. If implemented, the enforcement of these password requirements are to conform to the guidelines specified in *Section 3.1.3.1, Password Guidelines*. The site *Security Features Users Guide (SFUG)* is to document those requirements that are not enforced by the software, and provide administrative direction for adhering to them.

3.1.3.3 PassTickets

A PassTicket is a temporary, one-time-only password generated by the ACP from a request received by an application or function. It is used as an alternative to the standard ACP password and eliminates the need to send passwords across the network in clear text. End users of an application can use a PassTicket to authenticate their identity and log on to computer systems. The implementation and use of PassTickets is controlled using a unique resource class defined to the ACP.

The IAO is to ensure that read access to entities defined to the PassTicket resource class is restricted to systems programming personnel, security personnel, and authorized applications. End users are not to be granted read, or any other access level, to PassTicket entities.

The IAO is to ensure that update and alter access to entities defined to the PassTicket resource class is restricted to security personnel.

The IAO is to log all update and alter access to entities defined to the PassTicket resource class.

3.1.4 Special Privilege Access

Each ACP provides special privileges. When assigned to a userid, these special privileges allow the user to do tasks such as the following:

- Modify the security environment.
- Perform auditing tasks.
- Perform functions that circumvent the ACP.

The following sections outline recommendations that are to be used to reduce any effect on the operating environment.

3.1.4.1 Access Control Product Modification Privileges

Only the IAO is to be given any privileges that can modify the security environment, such as changing system-wide options.

Users allowed to perform security administration for application-related data are to be limited by the ACP to only change properties for which the user is responsible.

3.1.4.2 Audit Privileges

Privileges to view the contents of the security database may be granted to individuals by the IAO, provided a valid need exists. In many data centers, this access may be required for interactive system programmers to work with the user community to resolve problems.

3.1.4.3 Tape Label Bypass Privileges

Access to privileges to perform tape bypass label processing (BLP) is to be tightly controlled and only given to those authorized data center individuals (e.g., the tape librarian, Operations staff, or user) who require such access. Tape label bypass privileges allow a user to access data on a tape, using BLP processing, and, as such, to bypass any security-related controls. Therefore, authorization to perform BLP processing by the user community is to be tightly controlled. This is because a severe exposure exists in that any data on any tape can be accessed.

3.1.4.4 Other Sensitive Privileges

In addition to the special privileges specifically noted above, many other special privileges pose the danger of compromising the operational environment when misused or improperly applied. Each ACP provides the ability to control these privileges and to restrict them only to those personnel with valid requirements for their use. These special privileges include, but are not limited to, the ability to do the following tasks:

- Mount tape volumes to a TSO session.
- Access system console information.
- Issue console commands.
- Execute restricted programs.
- Access data and resources despite rule restrictions.

Restrict access to special privileges only to those individuals with an authorized need. Grant access to the minimum level necessary for the performance of job requirements.

3.1.5 Resource Controls

Resource controls are the base capabilities supplied by the ACP to control access to system-level resources. These include data set controls, volume controls, spool volume controls, and sensitive utility controls. Other resources such as CICS and database resources are addressed in the individual product sections.

3.1.5.1 Data Set Controls

The most common controls used in the ACP are for the protection of data sets. Data set protection with the ACP is a time-consuming task, but presents the best way to ensure that compromise does not occur.

When writing controls for data set access, consider the following common rules:

(1) Update data set controls as users are added/deleted from the systems, as the responsibilities of an individual change, and/or as data sets are added/deleted/renamed.

(2) On a semi-annual basis, review all system-level data sets to ensure that the authority granted to individual users is valid.

(3) Limit user access to system-level and product-level execution libraries to execution (*exec/fetch*) access. To prevent a user from copying a program and executing it from a personal library, restrict *read/update/alter* access to the systems personnel responsible for the installation and maintenance of the software.

- *(ACP00140: CAT II) The IAO will ensure that update and allocate access to all system-level product execution libraries are limited to system programmers only, unless a letter justifying access is filed with the IAO, Data set access authorization for the execution libraries is restricted to read access to systems programming personnel only.*

(4) Ensure that all data sets defined as part of a specific executive software product have established protection mechanisms. Only specific users who require authorization to make changes to these data sets should have such access.

(5) Protect all audit trail data (e.g., SMF data, TMON log data, SYSLOG) via the ACP. These audit trails are required for problem diagnosis, security investigations, and customer billing. *Update* and *alter* access needs to be tightly controlled from the point of the log/dump data sets down to the point where the final repository (e.g., MICS or a master merge file) is fed. Specific batch userids are to have the authority to manage these processes. Restrict authority for an individual to *update, alter,* or *read* the file(s) to authorized personnel. Log all personnel access for *update* or *alter* using the ACP's logging facility.

(6) Tightly restrict all access to all ACP files. *Update* and *alter* access is restricted to the systems programming personnel and Security Administrators. The IAO or IAM will document and maintain any exceptions. Updates to the security database will always be tracked by the utility programs provided with the ACP. Access to these files will be audited at all times.

- *(ACP00120: CAT II) The IAO will ensure that update and alter access to all ACP files and/or databases are limited to system programmers and/or security personnel, unless a letter justifying access is filed with the IAO, and all access is logged.*

(7) Catalog all data sets either in the Master Catalog, or in an appropriately defined and connected User Catalog. Identify and define all catalog aliases to the ACP. Assign ownership of, and responsibility for, each alias to the appropriate data owner. Establish access rules for each alias, and also for all data sets defined under each alias.

- *(ACP00260: CAT II) The IAO will ensure that all CATALOG ALIASES are defined to the ACP including ownership and responsibility assigned to the data owner with access rules defined.*

(8) Protect the System Master Catalog via the ACP to prevent update activity against it outside of, and without the approval of, the change management process. Update activity includes the addition or deletion of data sets or aliases and connection or disconnection of User Catalogs, etc. Log all *alter* and *update* data set access to the System Master Catalog using the ACP's facilities. Only systems programming personnel will be authorized *alter* and *update* data set access to the System Master Catalog. The IAO will maintain the access requirements (e.g., DD form 2875), and will maintain and review the ACP logging reports. Refer to *Section 10.3.2, FACILITY Resource Class*, for additional information regarding System Master Catalog access controls.

- *(ACP00130: CAT II) The IAO will ensure that update and allocate access to MASTER CATALOG is limited to system programmers only, unless a letter justifying access is filed with the IAO, all update and allocate access is logged.*

(9) Protect the System User Catalogs via the ACP to prevent unauthorized access. Grant users access to catalogs only as required for the performance of their assigned functions. Log all *alter* access to the System User Catalogs using the ACP's facilities. Only systems programming personnel will be authorized to alter the System User Catalogs. The IAO will maintain and review the ACP logging reports.

- *(ACP00135: CAT II) The IAO will ensure that allocate access to USER CATALOGS are limited to system programmers only, unless a letter justifying access is filed with the IAO, all allocate access is logged.*

(10) Under normal circumstances, data sets are not to contain hard-coded passwords in clear text. On rare occasions a business requirement may exist for passwords to be hard-coded in a file (e.g., in production JCL). To prevent compromise in such cases, ensure that the data set is adequately protected. The IAO is to review and approve all such requirements.

- *(ACP00245: CAT II) The IAO will ensure that the data sets containing userids and passwords are restricted to authorized users only and documentation justifying access authorization is maintained.*

(11) Restrict *update* and *alter* access to all proclibs referenced in the JES2 or JES3 procedure for started tasks (STCs) and TSO logons only to systems programming personnel.

- *(ACP00250: CAT II) The IAO will ensure that update and alter access to all system proclib datasets are limited to system programmers only, unless a letter justifying access is filed with the IAO.*

3.1.5.2 Volume Controls

The capability exists within the products to protect data at the volume level, rather than at the data set level. Protect all data at the specific data set name level, rather than at the volume level. Unique cases may exist for stand-alone systems and distribution volumes, which may require security at a volume level. These volume-level types of access are to be tightly controlled with the ACP's features. The IAO is to fully document the access.

3.1.5.3 Sensitive Utility Controls

Sensitive utilities are required in a data center to support operations. However, the uncontrolled use of these utilities could result in a major system failure, loss of data, or a potential security exposure. Restrict these sensitive utilities only to the personnel who require access to them. Each ACP product offers protection of utilities at the program level. Systems personnel should evaluate utilities installed on the system to determine if the criteria mentioned above apply. They should also follow up with the IAO to ensure the utilities are protected.

The following table gives some examples of the types of utilities that are to be controlled, and the personnel to whom their use is to be restricted:

Table A-26. SENSITIVE UTILITY CONTROLS (3.1.5.3 a)

SENSITIVE UTILITY CONTROLS	
UTILITY TYPE	LEGITIMATE USERS
Tape Management	Tape Librarian
DASD Management	DASD Management staff
Job Scheduling	Production Control
Storage Alteration	Systems Programming
System Modification	Systems Programming

The following table provides a sample list of the minimal entries to be controlled:

Table A-27. SENSITIVE UTILITY CONTROLS (3.1.5.3 b)

SENSITIVE UTILITY CONTROLS		
PROGRAM	PRODUCT	FUNCTION
***GTF**	OS/390	System Activity Tracing
***IOCP	OS/390	System Configuration
*MASPZAP	OS/390	Data Management
AMAZAP	OS/390	Data Management
BLSROPTR	OS/390	Data Management
DEBE	OS/DEBE	Data Management
DITTO	OS/DITTO	Data Management
FDRZAPOP	FDR	Product Internal Modification
GIMSMP	SMP/E	Change Management Product
ICKDSF	OS/390	DASD Management
IDCSC01	OS/390	IDCAMS Set Cache Module
IEHATLAS	OS/390/DFP	Data Management
IEHD****	OS/390/DFP	DASD Management
IEHINITT	OS/390	Tape Management
IFASMFDP	OS/390	SMF Data Dump Utility
IGWSPZAP	OS/390	Data Management
IND$FILE	OS/390	PC to Mainframe File Transfer (Applicable only for classified systems)
*****SCP	OS/390	System Configuration
WHOIS	OS/390	Share MOD to identify user name from USERID. Restricted to data center personnel only.

If these or similar utilities exist on a system, they are to be controlled. Refer to the specific Access Control Product section for details on securing these utilities. Additional information about specific sensitive utility programs that require restriction is provided in the sections of this document dealing with individual software products.

3.1.5.4 Dynamic List Controls

As discussed in *Section 2.1.2*, *Software Integrity*, new releases of MVS/ESA have added new types of dynamic lists to the PROG*xx* member of SYS1.PARMLIB, and have provided security controls for both the old and new types of lists. If these facilities are made available to operators, they are to be controlled. Use the following recommendations and techniques to control the potential exposures created by these facilities:

(1) Define the following resources in the FACILITY class with a default access of *none*:

- CSVAPF.**
- CSVAPF.MVS.SETPROG.FORMAT.DYNAMIC
- CSVAPF.MVS.SETPROG.FORMAT.STATIC
- CSVDYNEX.**
- CSVDYNL.**
- CSVDYNL.UPDATE.LNKLST
-

(2) Limit authority to those resources to Systems personnel. Restrict this access to the absolutely minimum number of personnel, and log all accesses.

(3) Limit authority to the SET PROG=, SETLOAD, and SETPROG commands to Systems personnel. Restrict this access to the absolutely minimum number of personnel, and log all accesses. Refer to *Section 3.1.5.6, OS/390 System Command Controls*.

- *(ACP00270: CAT II) The IAO will ensure that Issuing of Dynamic List Commands are defined to the FACITITY resource class and protected. Only system programmers and a limited number of authorized people are able to issue these commands. All access is logged.*

3.1.5.5 MCS Console Controls

The extended MCS support added in MVS/ESA SP, Version 4, Release 3, allows the installation to control the use of MCS consoles through the ACP. It also provides the ability for authorized TSO users to enter OS/390 system commands. This allows the installation to place tighter controls on who can use the operators' consoles, and provides a secure replacement for various third-party console facilities. To control entry of OS/390 system commands in MVS/ESA 4.3 and later, apply the following recommendations when implementing security:

(1) Define each console to the ACP as a resource in the CONSOLE class with a default access of *none*.

(2) Grant the userid for each console *read* access to the corresponding resource in the CONSOLE class.

(3) Grant each operator and systems programmer *read* access to the appropriate resources in the CONSOLE class, either directly or by granting access to a connected group.

- *(ACP00293: CAT II) The IAO will ensure that all MCS consoles are defined to the CONSOLE resource class and READ access is limited to operators and system programmers, unless a letter justifying access is filed with the IAO.*

(4) Define the CONSOLE resource in the TSOAUTH class with a default access of *none*.

- *(ZTSO0030: CAT II) The IAO will ensure that the CONSOLE resource is assigned to authorized personnel.*

(5) At the discretion of the IAO, users may be allowed to issue OS/390 system commands from a TSO session. Define an OPERPARM segment for such users with AUTH(INFO), grant *read* access to the MVS.MCSOPER.userid resource in the OPERCMDS class, and grant *read* access to the CONSOLE resource in the TSOAUTH class. Do not specify an authority higher than INFO in any user's OPERPARM segment. Refer to *Section 3.1.5.6, OS/390 System Command Controls*, about granting access to commands other than DISPLAY, MONITOR, STOPMN, STOPTR, and TRACK.

- *(ACP00294: CAT II) The IAO will ensure that all users that have access to the CONSOLE resource in the TSOAUTH resource class are properly defined.*

(6) Do not use uncontrolled third-party facilities, such as the CONSOLE commands from the CBT. Modify any facility that issues operator commands to include proper access controls and review the facility prior to use.

3.1.5.6 OS/390 System Command Controls

The extended MCS support added in MVS/ESA SP, Version 4, Release 3, allows the installation to control the use of OS/390 system commands through the ACP. These commands are subject to various types of potential abuse. For this reason, it is necessary to place restrictions on the OS/390 system commands that can be entered by particular operators.

Some commands are particularly dangerous and should only be used when all less drastic options have been exhausted. Misuse of these commands can create a situation in which the only recovery is an IPL. These commands are to be referred to as sensitive commands in this section.

NOTE: Access controls for JES commands are discussed in a later section. To control access to OS/390 system commands in MVS/ESA, Release 3.1.3 and later, apply the following recommendations when implementing security:

(1) Define the **MVS.**** resource in the **OPERCMDS** class with a default access of *none*.

(2) Define categories of users to the ACP for the following:

- Network personnel
- Operations personnel (Junior)
- Operations personnel (Senior)

- Systems personnel
- Users without the above responsibilities

Where one of the above includes users with significantly different responsibilities, define as many categories as necessary to give appropriate access to resources at the category level.

(3) Associate each operator with the appropriate security categories defined above.

(4) Document in the installation SOP which users have access to which commands, whether that access is logged, and the justification for that access. The documentation of user access should be written in terms of responsibilities and roles rather than individual user names. Where this *STIG* explicitly permits access to particular commands, a reference to this *STIG* is all the justification that is required. Where this *STIG* defers the decision to the IAO, the IAO is to consult with Operations and Systems, and the decision is to be documented. However, if the IAO, in consultation with Operations and Systems, specifies a more restrictive level of access than specified in this *STIG*, no justification need be given.

As part of this documentation, specify the policies and procedures for the use of sensitive OS/390 commands. Restrict access to these commands to the absolutely minimum number of personnel, and log all access. The IAO will take into account the need to issue these commands in emergency situations and for system shutdowns and upgrades.

- *(ACP00284: CAT II) The IAO will ensure that OS/390 Sensitive System Commands are documented in the installation SOP as described above.*

(5) Collect SMF data for specific commands as shown below in *Table A-29, Controls on OS/390 System Commands*.

(6) All users may be permitted to CANCEL, MODIFY, or STOP jobs for which they are responsible, at the discretion of the IAO. However, these commands are to be logged.

NOTE: This is possible only if there is a naming convention that associates *jobnames* with groups or users.

(7) Prepare ACP controls to grant access to commands using *Table A-29, Controls on OS/390 System Commands* as a guideline for each category of users. All required resource logging of OS/390 system commands is to be performed using the ACP. In general, the commands are controlled by selectively granting access to resources in the OPERCMDS class, with names such as MVS.*command.qualifier* or MVS.*command.qualifier.object*. Tailor the permissions granted to the users' roles rather than using a wild card for the command object in every permission.

If IBM's System Display and Search Facility (SDSF) product is installed on the system, additional resource access to MVS system commands is allowed to general users when using SDSF. These resources are to be noted in the following table. Refer to *Section 14.2.1.7.8, MVS and JES2 Commands Generated by SDSF*, for additional security requirements.

Please note that the resource names in the table ending with .* are intended to show that zero or more qualifiers are possible. It is not intended to indicate a specific ACP wildcard character.

At the discretion of the IAO, access may be granted to individual users or started tasks, but document the justification.

Table A-28. REQUIRED CONTROLS ON OS/390 SYSTEM COMMANDS (3.1.5.6)

REQUIRED CONTROLS ON OS/390 SYSTEM COMMANDS				
COMMAND	RESOURCE(s)	GROUP	AUTH?	LOG. REQ'D
ACTIVATE	MVS.ACTIVATE	Operations (Sr.) Systems	I Y	Y Y
CANCEL (own jobs)	MVS.CANCEL.JOB.*	All	I	Y
CANCEL (other)	MVS.CANCEL.ATX.* MVS.CANCEL.DEV.* MVS.CANCEL.STC.* MVS.CANCEL.TSU.* NOTE: The ATX.* and TSU.* resources may be permitted to general users when using SDSF.	Operations (Jr.) Operations (Sr.)	I Y	Y Y
CHNGDUMP	MVS.CHNGDUMP	Systems	Y	Y
CONFIG	MVS.CONFIG	Operations (Sr.) Systems	Y I	Y Y
CONTROL	MVS.CONTROL.*	All Operations Systems	R Y I	N I I
DEVSERV	MVS.DEVSERV	All Operations Systems	R Y I	N I I
DISPLAY	MVS.DISPLAY.* MVS.DISPLAY.WLM	All Operations	R Y	N N
DUMP	MVS.DUMP	Operations Systems	Y I	Y Y
DUMPDS	MVS.DUMPDS	Operations Systems	Y I	I I

REQUIRED CONTROLS ON OS/390 SYSTEM COMMANDS				
COMMAND	RESOURCE(s)	GROUP	AUTH?	LOG. REQ'D
FORCE	MVS.FORCE.DEV.* MVS.FORCE.JOB.* MVS.FORCE.STC.* MVS.FORCE.TSU.* MVS.FORCEARM.DEV.* MVS.FORCEARM.JOB.* MVS.FORCEARM.STC.* MVS.FORCEARM.TSU.*	Operations (Sr.) Systems	S S	Y Y
HALT	MVS.HALT.EOD MVS.HALT.NET	Operations (Sr.) Systems	S S	Y Y
IOACTION	MVS.IOACTION	Operations (Sr.) Systems	I Y	Y Y
LIBRARY	MVS.LIBRARY	Operations	Y	Y
LOG	MVS.LOG	Operations	Y	N
MODE	MVS.MODE	Operations (Sr.) Systems	Y Y	Y Y
MODIFY (own jobs)	MVS.MODIFY.JOB.*	All	I	Y
MODIFY (GTF tasks)	MVS.MODIFY.STC.GTF	Systems	Y	Y
MODIFY (other)	MVS.MODIFY.STC.*	Operations (Jr.) Operations (Sr.)	I Y	Y Y
MONITOR	MVS.MONITOR	All Operations	R Y	N N
MOUNT	MVS.MOUNT	Operations	Y	N
MSGRT	MVS.MSGRT	Operations	Y	N
PAGEADD	MVS.PAGEADD	Operations Systems	I Y	Y Y
PAGEDEL	MVS.PAGEDEL	Operations Systems	I Y	Y Y
QUIESCE	MVS.QUIESCE	Operations (Sr.)	Y	Y
REPLY	MVS.REPLY	Network Operations Systems	Y Y Y	Y I Y
RESET	MVS.RESET MVS.RESETCN	Operations (Sr.)	Y	Y
ROUTE	MVS.ROUTE.CMD.*	Operations Systems	Y Y	N N
SEND	MVS.SEND	Operations	Y	N
SET CLOCK	MVS.SET.*	Operations	Y	N

REQUIRED CONTROLS ON OS/390 SYSTEM COMMANDS				
COMMAND	RESOURCE(s)	GROUP	AUTH?	LOG. REQ'D
SET DATE	MVS.SET.*	Operations	Y	N
SET DIAG	MVS.SET.*	Systems	Y	Y
SET OMVS	MVS.SET.*	Systems	Y	Y
SET PROD	MVS.SET.*	Systems	Y	Y
SET PROG	MVS.SET.*	Systems	I	Y
SET (other)	MVS.SET.*	Discretion of IAO	I	Y
SETDMN	MVS.SETDMN.DMN	Systems	Y	Y
SETETR	MVS.SETETR.ETR	Operations (Sr.)	Y	Y
SETGRS	MVS.SETGRS.MODE.STAR	Systems	Y	Y
SETIOS	MVS.SETIOS.IOS	Systems	Y	Y
SETLOAD	MVS.SETLOAD.*	Systems	I	Y
SETLOGRC	MVS.SETLOGRC.LOGRC	Operations (Sr.) Systems	I	Y
SETOMVS	MVS.SETOMVS.OMVS	Systems	I	Y
SETPROG	MVS.SETPROG.*	Systems	I	Y
SETSMF	MVS.SETSMF.SMF	Systems	I	Y
SETSMS	MVS.SETSMS.SMS	Systems	Y	Y
SETSSI	MVS.SETSSI.*	Systems	I	Y
SETXCF Assign separate permissions for each function needed.	MVS.SETXCF.XCF	Operations Systems	I I	Y Y
SLIP	MVS.SLIP	Operations (Sr.) Systems	I Y	Y Y
START	MVS.START.STC.*	Operations	Y	Y
START (GTF tasks)	MVS.START.STC.GTF	Systems	Y	Y
STOP	MVS.STOP.JOB.* MVS.STOP.STC.*	Operations	Y	Y
STOP (GTF tasks)	MVS.STOP.STC.GTF	Systems	Y	Y
STOPMN	MVS.STOPMN	All	Y	N
STOPTR	MVS.STOPTR	All	Y	N

REQUIRED CONTROLS ON OS/390 SYSTEM COMMANDS				
COMMAND	RESOURCE(s)	GROUP	AUTH?	LOG. REQ'D
SWAP	MVS.SWAP	Operations (Sr.)	Y	Y
SWITCH	MVS.SWITCH.*	Operations	Y	Y
TRACE	MVS.TRACE.*	Systems	Y	Y
TRACK	MVS.TRACK	All Operations	R Y	N N
Unknown commands	MVS.UNKNOWN	Operations (Sr.) Systems	I Y	Y Y
UNLOAD	MVS.UNLOAD	Operations	Y	Y
VARY CN	MVS.VARY.CN	Operations (Sr.)	Y	Y
VARY FORCE VARY GRS VARY SMS VARY WLM VARY XCF	MVS.VARYFORCE.DEV MVS.VARY.GRS MVS.VARY.SMS MVS.VARY.WLM MVS.VARY.XCF	Operations (Sr.) Systems	S S	Y Y
VARY (other)	MVS.VARY.* MVS.VARY.CONSOLE MVS.VARY.DEV MVS.VARY.HARDCPY MVS.VARY.MSTCONS MVS.VARY.NET MVS.VARY.PATH MVS.VARYAUTH.* MVS.VARYAUTH.CN MVS.VARYAUTH.CONSOLE MVS.VARYAUTH.DEV	Operations	Y	Y
WRITELOG	MVS.WRITELOG	Operations	Y	Y

In the authorized and logging columns, Y=Yes, N=No, R=Read (lowest) access only, I=IAO discretion, and S=Sensitive command.[10]

- *(ACP00282: CAT II) The IAO will ensure that OS/390 Sensitive System Commands are defined to the OPERCMDS resource class. Only limited number of authorized people are able to issue these commands. All access is logged.*

[10] As noted above, access will be limited to the minimum number of senior personnel.

3.2 ACF2

3.2.1 Standard Global Options (GSO Records)

The following table depicts the STIG required values for the ACF2 Global System Options records. The options specified are STIG requirements and each site can choose to be more restrictive.

Table A-29. REQUIRED GLOBAL OPTIONS (GSO RECORDS) - ACF2 (3.2.1)

REQUIRED GLOBAL OPTIONS (GSO RECORDS) - ACF2		
OPTION	DESCRIPTION	REQUIRED VALUE
APPLDEF	Defines site-unique structured Infostorage records when standard structured Infostorage records will not suffice. ***NOTE:*** The APPLDEF record is optional. Use of this record will be justified in writing with supporting documentation.	Site defined.

AUTHEXIT	Contains the vendor or site exit information that supports an extended authentication facility, such as operator identification (OID) card support.	GSO AUTHEXIT.001 record:
		LIDFIELD(AUTHSUP1)
		PROCPGM(AUTHXNCP)
		NOINFOSTG

AUTOERAS	Controls the automatic physical erasure of VSAM or non-VSAM data sets. CAUTION: Use of the Automatic Erase Feature can cause considerable system overhead affecting system performance.	Unclassified Systems: NONON-VSAM NOVSAM VOLS() Classified Systems: NON-VSAM VSAM VOLS(-)
BACKUP	Controls automatic Security File backup.	Site defined ***NOTE***: a time must be specified unless the database is shared and backed up on another system.
BLPPGM	Specifies those programs authorized to use tape bypass label processing (BLP).	None will be specified. ***NOTE***: BLP enforcement will be done based on LID record settings.

REQUIRED GLOBAL OPTIONS (GSO RECORDS) - ACF2		
OPTION	DESCRIPTION	REQUIRED VALUE
CLASMAP	Translates an eight-character SAF resource class into a three-character ACF2 resource type code to enable resource rules to be written to perform validation. Also it translates the resource type codes for ACF2 calls or calls made to ACF2 from CA's International Standard Security Facility (CAISSF).	Vendor defaults as specified in the internal **CLASMAP** records unless as indicated otherwise below. The following resource class to resource type translations are the STIG recommended standard: APPL maps to APL CONSOLE maps to CON FACILITY maps to FAC OPERCMDS maps to OPR TSOAUTH maps to TSO

EXITS	Specifies the module names of site-written ACF2 exit routines.
	NOTE: The DSNPOST exit is optional and is not required to be specified in the GSO EXITS record.
		DSNPOST(*module*)
		SEVPRE(SEVPRE01)
		SEVPOST(SEVPST01)
		NOTE: No other exits are authorized at this time.
		NOTE: Local changes will be justified in writing with supporting documentation.

LINKLST	Specifies one or more partitioned data sets considered part of the system link (**SYS1.LINKLIB**) during data set access validation.	Site defined. Only trusted system data sets will be listed. Application libraries will never be included.
MAINT	Specifies the logonid, program, and library combinations used for system maintenance functions. ***NOTE:*** For logonids that match environments described in records, **no** SMF logging records will be created.	Site defined. ***NOTE:*** Entries will be restricted to production storage management user accounts and programs.

REQUIRED GLOBAL OPTIONS (GSO RECORDS) - ACF2		
OPTION	DESCRIPTION	REQUIRED VALUE
NJE	Specifies ACF2 validation options that apply to jobs submitted through a network job entry subsystem (JES2, JES3, RSCS).	DFTLID() INHERIT NODEMASK(-) ENCRYPT VALIN(YES) NOVALOUT ***NOTE:*** For NJE nodes that are incompatible with the XDES algorithm, discrete NJE records will be created with NOENCRYPT. ***NOTE:*** Local changes will be justified in writing with supporting documentation.
OPTS	Defines the global options available to the system.	BLPLOG NOCACHE NOCMDREC CONSOLE(NOROLL) CPUTIME(LOCAL) DATE(MDY) NODDB DFTLID() DFTSTC() INFOLIST(SECURITY, AUDIT) JOBCHK MAXVIO(10) MODE(ABORT) NOTIFY RPTSCOPE SHRDASD STAMPSMF STC TAPEDSN NOUADS NOVTAMOPEN
PPGM	Defines protected programs that can only be executed by privileged users.	PGM-MASK(pgm-mask1, ...,pgm-mask255) Refer to the table in *Section 3.1.5.3, Sensitive Utility Controls*, for the minimal list of programs to be controlled.

REQUIRED GLOBAL OPTIONS (GSO RECORDS) - ACF2		
OPTION	DESCRIPTION	REQUIRED VALUE
PSWD	Defines various logonid password options and controls. ***NOTE:*** If NOTE 12 is installed and its function is to increase the password history to 10 entries, set the GSO PSWD option to NOPSWDHST.	MAXTRY(3) MINPSWD(8) PASSLMT(5) PSWDALT PSWDFRC PSWDHST PSWDJES PSWDLID PSWDNCH PSWDNUM PSWDREQ PSWDRSV NOPSWDXTR WRNDAYS(10)
RESRULE	Specifies data set access rules that are to be made resident at ACF2 initialization time.	None. ***NOTE***: Local changes will be justified in writing with supporting documentation.
RESVOLS	Defines the DASD and mass storage volumes for which ACF2 is to provide data set-level protection.	VOLMASK(-) ***NOTE***: Local changes will be justified in writing with supporting documentation.
RULEOPTS	Specifies the options that determine how resource and access rules are used and maintained.	CENTRAL CHANGE DECOMP(SECURITY,AUDIT) NO$NOSORT RULELONG|NORULELONG NOVOLRULE
SAFDEF	Defines System Authorization Facility (SAF) calls that each site may want to process differently than the default ACF2 process.	Vendor defaults as specified in the internal **SAFDEF** records. ***NOTE***: All vendor-modified and site-defined SAFDEF records will be justified in writing with supporting documentation.
SECVOLS	Defines those DASD, mass storage, and tape volumes for which ACF2 is to provide volume-level protection.	VOLMASK() ***NOTE***: Local changes will be justified in writing with supporting documentation.

REQUIRED GLOBAL OPTIONS (GSO RECORDS) - ACF2		
OPTION	DESCRIPTION	REQUIRED VALUE

SYNCOPTS	Defines the cache synchronization processing for a CPU running in a shared ACF2 database environment.	FILENAME(ACF2.SYNCFILE)
		POLLINTV(10)
		USECOUNT(10)
		NOACTIVATE

TSO	Specifies global usage and system parameters that define and control the TSO logon process and other system parameters.	ACCOUNT(1)
		BYPASS(#)
		CHAR(BS)
		CMDLIST()
		NOFSRETAIN
		LINE(ATTN)
		LOGONCK
		PERFORM(0)
		PROC(IKJACCNT)
		NOQLOGON
		REGION(site defined)
		SUBCLSS()
		SUBHOLD()
		SUBMSG()
		TIME(0)
		TSOSOUT(A)
		UNIT(SYSDA)
		WAITIME(60) or less

TSOCRT	Defines a clear string used to obliterate the logon to ASCII CRT devices.	STRING(A12FA11C1A270C0D)
TSOKEYS	Defines site-supplied keywords permitted by ACF2 at TSO logon time.	KEYWORDS()

TSOTWX	Defines a cross-out mask to obliterate the logon password on TWX devices.	CR(15)
		IDLE(17)
		LENGTH(8)
		M1(X)
		M2(N)
		M3(Z)
		M4(M)
		STRING()

TSO2741	Defines a cross-out string used to obliterate the logon password on 2741 devices.	BS(16)
		LENGTH(8)
		M1(X)
		M2(N)
		M3(Z)
		M4(M)
		STRING()

REQUIRED GLOBAL OPTIONS (GSO RECORDS) - ACF2		
OPTION	DESCRIPTION	REQUIRED VALUE

UNIXOPTS	Specifies global options pertinent to the UNIX System Services (OMVS) environment.
	NOTE: DFTGROUP and DFTUSER should only be used for non classified systems using FTP. Restrictions apply. Refer to *Section 2.5.2.6.2. Paragraph 7,* 'Unprivileged Users and Groups' *of this STIG for further information*.
		CHOWNRES
		DFTGROUP(*defaultgroup*)
		DFTUSER(*defaultuser*)

- *(ACF0250: CATIII) The IAO will ensure that the APPLDEF GSO record if used has supporting documentation indicating the reason it was used.*

- *(ACF0260: CAT II) The IAO will ensure that the AUTHEXIT GSO value is used to define an extended user authentication exit to is invoked at TSO logon, for Operator Identification (OID) card usage. DISA requires the use of NCPASS on all of its domains. DISA sites require the use of AUTHEXIT for other non DISA sites this value is optional.*

- *(ACF0270: CAT II) The IAO will ensure that the AUTOERASE GSO value indicates that you would like ACF2 to control the automatic physical erasure of VSAM or non-VSAM data sets. See above table for non-classified and classified values.*

- *(ACF0280: CAT II) The IAO will ensure that the BACKUP GSO value specifies a time field and Time(00:00) is not specified unless the database is shared and backed up on another system.*

- *(ACF0290: CAT II) The IAO will ensure the BLPPGM GSO value indicates that ACF2 does not control the programs authorized to use tape bypass label processing (BLP).*

- *(ACF0300: CAT II) The IAO will ensure the CLASMAP GSO value translates an eight-character SAF resource class into a three-character ACF2 resource type code.*

- *(ACF0310: CAT II) The IAO will ensure the EXITS GSO value specifies the module names of site-written ACF2 exit routines. The above table indicates DISA defaults are optional for non-DISA sites.*

- *(ACF0330: CAT II) The IAO will ensure the LINKLIST GSO value if specified only contains trusted system datasets.*

- *(ACF0350: CAT II) The IAO will ensure the MAINT GSO value if specified* will be restricted *to production storage management* user accounts and programs.

- *(ACF0360: CAT II) The IAO will ensure that the NJE GSO value indicates validation options that apply to jobs submitted through a network job entry subsystem (JES2, JES3, RSCS). See above table for values specified.*

- *(ACF0370: CAT I/II) The IAO will ensure that the OPTS GSO value is set to valid options specified in the above table. If not equal to specified settings, then it is a CAT II finding. If MODE does not indicate abort, then upgrade this finding to a CAT I finding.*

- *(ACF0380: CAT II) The IAO will ensure that the PPGM GSO value indicates protected programs that are only executed by privileged users.*

- *(ACF0390: CAT II) The IAO will ensure that the PSWD GSO values are set to the values specified in the above table.*

- *(ACF0410: CAT II) The IAO will ensure that the RESRULE GSO value is set to NONE any other setting requires documentation justifying the change.*

- *(ACF0420: CAT II) The IAO will ensure that the RESVOL GSO value is set to Volmask(-). Any other setting requires documentation justifying the change.*

- *(ACF0430: CAT II) The IAO will ensure that theRULEOPTS GSO values are set to the values specified in the above table.*

- *(ACF0480: CAT II) The IAO will ensure that the SECVOLS GSO value is set to VOLMASK(). Any local changes are justified and documented with the IAO.*

- *(ACF0490: CAT II) The IAO will ensure that the SYNCOPTS GSO values are set to the values specified in the above table.*

- *(ACF0500: CAT II) The IAO will ensure that the TSO GSO values are set to the values specified in the above table.*

- *(ACF0510: CAT II) The IAO will ensure that the TSOCRT GSO values are set to the values specified in the above table.*

- *(ACF0520: CAT II) The IAO will ensure that the TSOKEYS GSO value is set to KEYWORDS().*

- (*ACF0530: CAT II) The IAO will ensure that the TSOTWX GSO values are set to the values specified in the above table.*

- *(ACF0540: CAT II) The IAO will ensure that the TSO2741 GSO values are set to the values specified in the above table.*

3.2.2 Userid Controls

Every user will be identified to ACF2 via a unique userid. (ACF2 calls this a **logonid**.) To ACF2, a user is an individual, a started task, or a batch job.

Every user will be fully identified within ACF2. Complete the following fields for every logonid:

NAME	-	User's name
UID-String	-	All fields defined in the ACFFDR @UID macro

All fields that comprise the standard UID-string will be filled out for each user as a logonid is added.

- *(ACF0560: CAT III) The IAO will ensure that all LOGONID records have the required attributes.*

The following subsections define the requirements for defining an ACF2 user of OS/390 resources. Additional definitions are required for users accessing UNIX System Services resources. Please refer to *Section 2.5, OS/390 UNIX System Services*, for details.

3.2.2.1 Interactive Users

In addition to the standard logonid settings, individual users will be granted the minimum privileges necessary to accomplish their assigned functions. Certain default values will be specified for all interactive users as they are added to the system.

The ACF2 logonid record contains many fields. Many of these fields are reserved for use by ACF2, while others are highly site specific in content and use.

The following table provides values that will be specified for certain selected fields as user privileges and access are granted:

Table A-30. INTERACTIVE USERS - ACF2 (3.2.2.1)

INTERACTIVE USERS - ACF2		
FIELD	DESCRIPTION	REQUIRED VALUE
ALLCMDS/ NOALLCMDS	Ability to bypass ACF2 restricted command lists.	NOALLCMDS
AUTHSUP1	User Authorization Flag 1	ON for highly privileged users controlled by NC-PASS. ***NOTE***: Refer to Section 6.3.1, NC-PASS for ACF2, for further information.
CONSOLE/ NOCONSOLE	Permits access to the TSO/E CONSOLE facility.	NOCONSOLE The CONSOLE bit will not be turned on unless command-level controls are implemented.
GROUP(name)	This field is required for assigning *gids* to MVS OpenEdition users. ***NOTE***: For sites running UNIX Systems Services, see Section 2.5.3.2, Defining Users and Groups, for GROUP(name) requirements.	Will be defined for OpenEdition users.
IDLE(time)	Specifies the maximum time permitted (in minutes) between terminal transactions for this user. If exceeded, ACF2 needs the logonid and password to be revalidated before another transaction is accepted. Zero (**0**) indicates no limit is enforced. This field is available for IMS and CICS on-line processing.	IDLE(15)
INTERCOM/ NOINTERCOM	Indicates this user is willing to accept messages from other users through the TSO SEND command.	INTERCOM

INTERACTIVE USERS - ACF2		
FIELD	DESCRIPTION	REQUIRED VALUE
LGN-ACCT/ NOLGN-ACCT	Indicates permission to specify an account number at logon time. If a user has the PMT-ACCT field, ACF2 prompts the user for an account number unless an account number is specified before the prompt. If a user does not specify an account number at logon and PMT-ACCT is not specified in the user's logonid record, ACF2 uses the user's default account number (TSOACCT is the logonid field) or the system default account number. Specifies the default in the ACCOUNT field of the GSO TSO record.	LGN-ACCT
MAIL/NOMAIL	Indicates a user can receive mail messages from TSO at logon time.	MAIL
MAXDAYS(days)	Specifies the maximum number of days permitted between password changes before the password expires. Zero (**0**) indicates no limit.	MAXDAYS (90)
MINDAYS(days)	Specifies the minimum number of days that must elapse before a user can change a password. Zero (**0**) indicates no limit.	MINDAYS (1)
MOUNT/ NOMOUNT	Permission to issue mounts for devices.	NOMOUNT
MSGID/NOMSGID	Indicates this user wants TSO messages to have message IDs prefixed.	MSGID
NAME(username)	Specifies the 1- to 20-character name of the user. ACF2 displays this name on logging and security violation reports. ACF2 also uses this name as the NAME field of the job statement created for a TSO logon session, if the NOUADS field is specified in the GSO OPTS record.	Will be completed for all users.
NON-CNCL/ NONON-CNCL	ACF2 cannot cancel the user for security violations. Access is permitted but logged.	NONON-CNCL

INTERACTIVE USERS - ACF2		
FIELD	DESCRIPTION	REQUIRED VALUE
NO-STORE/ NONO-STORE	Specifies that a user cannot store or delete rule sets. This applies even if the value of the PREFIX field of the logonid record matches the $KEY of the rule of the data set, if the user has the SECURITY privilege, or if the user has change authority through a %CHANGE or %RCHANGE control statement in the rule set.	NONO-STORE ***NOTE***: The GSO RULEOPTS record must specify CENTRAL. Refer to the GSO Options in Table A-30, Standard Global Options (GSO Records) - ACF2.
NOTICES/ NONOTICES	Indicates a user can receive TSO notices at logon time.	NOTICES
OPERATOR/ NOOPERATOR	User has TSO operator privileges.	NOOPERATOR
PASSWORD	The logon password for the user.	Must be completed.
PHONE	Specifies the 1- to 12-character telephone number of a user.	Optional
PMT-ACCT/ NOPMT-ACCT	Indicates that ACF2 requires a user to specify an account at logon time and to specify the LGN-ACCT field also. ACF2 does not prompt for an account number if the FSRETAIN field is also specified. FSRETAIN obtains account values from the last session.	May be required for Fee-for-Service support.
PPGM/NOPPGM	User can execute protected programs specified in the GSO PPGM record.	NOPPGM
PREFIX	User access to the user's own data sets without rule validation.	PREFIX()
PROMPT/ NOPROMPT	Indicates that ACF2 prompts a user for missing or incorrect parameters.	PROMPT
RSRCVLD/ NORSRCVLD	Indicates that an access rule must validate any resource accesses that the user makes. Applies even if the user has ownership of the resource, or has the SECURITY attribute.	RSRCVLD
RULEVLD/ NORULEVLD	Indicates that an access rule must validate any data set accesses that the user makes. Applies even if the user has ownership of the data set, or has the SECURITY attribute.	RULEVLD
TSOACCT	Specifies the user's default TSO logon account. Used for all billing.	May be required for Fee-for-Service support.
TSOPROC	Specifies the user's default TSO logon procedure.	Will be completed for all TSO users.

INTERACTIVE USERS - ACF2		
FIELD	DESCRIPTION	REQUIRED VALUE
UID-String Fields	All fields defined in the @UID macro in the ACFFDR. UID-string fields currently are locally defined on each system. Their composition and contents will be fully documented by the IAO.	Will be completed. ***NOTE***: Only those fields necessary to restrict the user to those accesses and functions required to perform assigned tasks are required.
VLD-ACCT/ NOVLD-ACCT	Indicates that ACF2 validates the TSO account number of a user. Creates a resource rule with a type code TAC and a $KEY of the account number so that ACF2 will perform this validation.	VLD-ACCT May be required for Fee-for-Service support.
VLD-PROC/ NOVLD-PROC	Indicates that ACF2 validates the TSO logon procedure of a user. Creates a resource rule with a type code TPR and a $KEY of the logon procedure so that ACF2 will perform this validation.	VLD-PROC Will be completed for all TSO users.

- *(ACF0570: CAT III) The IAO will ensure that all LOGONID records for interactive users have the required attributes. See above table for the values for each parameter.*

3.2.2.2 Batch Users

The ability of a logonid to submit batch jobs for processing will be enabled through the setting of the JOBCHECK option on the GSO OPTS record. (Refer to *Section 3.2.1, Standard Global Options [GSO Records]*.) Logonids that can submit batch production work will have the following, in addition to the default LID (logonid) field settings:

JOB

All batch jobs scheduled via an automation process will use the **//*LOGONID *xxxxxxxx*** card in the JCL stream to identify the userid. Use **restricted logonids** with the following parameter coded:

RESTRICT

One or both of the following will also be specified:

PGM(*xxxxxxxx*) and SUBAUTH
SOURCE(*xxxxxxxx*)

The use of default IDs prevents the identification of tasks with individual users as mandated by policy, and prevents adequate accountability. Default IDs for batch processing will not be used.

- *(ACF0580: CAT II) The IAO will ensure that Logonids associated with batch jobs having the RESTRICT attribute, will have either the PGM(xxxxxxxx) and SUBAUTH attributes, and/or the SOURCE(xxxxxxxx) attribute specified.*

Refer to *Section 3.1.2.2, Batch Users*, for further information.

3.2.2.3 STC Users

All started tasks will be assigned an individual logonid. The logonid for an Started Task Control (STC) will be granted the minimum privileges necessary for the STC to function. In addition to the default LID field settings, all STC logonids will have the following field setting:

STC

- *(ACF0600: CAT II) The IAO will ensure that all logonid records assigned to started tasks have the* ***STC*** *attribute specified.*

If the STC is a Multi-User Single Address Space System (MUSASS), the STC logonid will also have the following attributes:

MUSASS
NO-SMC

- *(ACF0610: CAT II) The IAO will ensure that if the STC is a Multi-User Single Address Space System (MUSASS), the STC logonid has the MUSASS and NO-SMC attributes.*

If the Multi-User Single Address Space System (MUSASS) has the requirement to submit jobs on behalf of its users, the STC logonid will also have the following attribute:

JOBFROM

- *(ACF0620: CAT II) The IAO will ensure that if the Multi-User Single Address Space System (MUSASS) has the requirement to submit jobs on behalf of its users, the STC logonid has the JOBFROM attribute specified.*

If the Multi-User Single Address Space System (MUSASS) has the requirement to update information in the ACF2 database on behalf of its users, the STC logonid will also have the following attribute:

MUSUPDT

- *(ACF0630: CAT II) The IAO will ensure that if the Multi-User Single Address Space System (MUSASS) has the requirement to update information in the ACF2 database on behalf of its users, the STC logonid has the MUSUPDT attribute specified.*

The use of default IDs prevents the identification of tasks with individual users as mandated by policy, and prevents adequate accountability. Default IDs for STCs will not be used.

Certain started tasks performing critical operating system-related functions may be considered trusted for the purposes of data set and resource access requests. For these STCs all access requests will be honored. These STCs will be given the following attribute to facilitate access while logging any accesses they would not ordinarily be granted by the access rule sets:

NON-CNCL

- *(ACF0640: CAT II) The IAO will ensure that only STC in the trusted STC list can have the NON-CNCL attribute and any other STCs having this attribute are approved by the site DAA.*

Refer to *Section 3.1.2.3, Started Task Control (STC) Users*, for further information.

3.2.2.4 Network Users

Jobs submitted from an RJE device will have a logonid card present in the JCL. This logonid card will be defined as a restricted logonid. A source parameter will be defined that is associated with a valid source group entry. The source group entry will clearly define the associated remote number and devices allowed to submit jobs with this logonid. The logonid will have the following field settings in addition to the default LID field settings:

RESTRICT
SOURCE(*xxxxxxxx*)

Jobs submitted via NJE will be defined with a valid userid and password present in the JCL. Since JES2 and ACF2 encrypt the password before the job is written to spool, access to this information is protected. Default logonids for NJEs will not be allowed.

- *(ACF0650: CAT II) The IAO will ensure that jobs submitted from an RJE device have the RESTRICT and SOURCE(xxxxxxxx) attributes specified.*

3.2.2.5 Special Storage Management Users

Production maintenance tasks manage the backups and restoration of data for the Continuity of Operations Plan (COOP) and media maintenance. Logonids assigned to production maintenance tasks will have the following field settings in addition to the default LID field settings:

JOB
MAINT

- *(ACF0680: CAT II) The IAO will ensure that logonids assigned to production maintenance tasks have the Job and Maint field settings in addition to the default LID field settings.*

An associated GSO MAINT record will exist for each special user logonid, identifying the program(s) that it is permitted to access and the library where the program(s) resides.

- *(ACF0660: CAT III) The IAO will ensure that an associated GSO maintenance record exists for each special user logonid identifying the program(s) that it is permitted to access and the library where the program(s) resides.*

- *(ACF0670: CAT III) The IAO will ensure that an associated user logonid exists for each special GSO maintenance record identifying the program(s) that it is permitted to access and the library where the program(s) resides.*

3.2.2.6 Emergency Userids

The following sections describe the controls implemented for userids whose use is intended only for emergency and recovery purposes.

Refer to *Section 3.1.2.6, Emergency Userids*, for further information.

3.2.2.6.1 Privileged Access Emergency Userids

As defined in *Section 3.1.2.6, Emergency Userids*, two classes of emergency userids may exist. The following privileges and specifications will be used for these logonids:

(1) For emergency IDs with the ability to access and update all system data sets, but which do not have security administration privileges:

NOFSRETAIN
JCL
JOB
MONITOR

NON-CNCL (Will force logging of all activity.)
TSO
TSOPROC(*xxxxxxxx*)
TSOACCT(none)

(2) For emergency IDs with security administration privileges, but which cannot access and update system data sets:

ACCOUNT
NOFSRETAIN
JCL
JOB
MONITOR
NONON-CNCL
RULEVLD
SECURITY
TSO
TSOPROC(*xxxxxxxx*)
TSOACCT(none)

- *(ACF0690: CAT II) The IAO will ensure that Emergency Logonids use the above fields to enforce restrictions itemized in Section 3.1.2.6, Emergency Userids.*

Refer to *Section 3.1.2.6, Emergency Userids*, Paragraph (1), for further information.

3.2.2.6.2 REFRESH Userids

An additional logonid will be maintained that can effect immediate changes to ACF2 global options via the ACF2 REFRESH privilege. This REFRESH logonid will have no access to system data sets, will be authorized only for the REFRESH privilege, and will be in SUSPEND status unless actually in use. It will be established with the following logonid parameters:

REFRESH
SUSPEND

- *(ACF0710: CAT III) The IAO will ensure Logonids with the refresh privilege are only available to IAOs /IAMs.*

- *(ACF0720: CAT II) The IAO will ensure that logonids with the REFRESH privilege are in SUSPEND status unless actually in use.*

When the IAO determines it necessary to refresh the ACF2 global options, the IAO will do the following:

(1) Activate the REFRESH ID with the following setting(s):

NOSUSPEND
NOPSWD-EXP
PASSWORD(new password)

(2) Instruct Operations to perform the REFRESH.

(3) Deactivate the REFRESH ID with the following setting:

SUSPEND

- *(ACF0730: CAT III) The IAO will ensure procedures and documentation as defined above only exists for the use of Logonids with the refresh attribute.*

Refer to *Section 3.1.2.6, Emergency Userids*, Paragraph (2) for further information.

3.2.2.7 FTP Userids

The ACF2 logonid record for an FTP user will be created with all required fields as discussed in *Section 3.2.2.1, Interactive Users*. For environments running OS/390 UNIX-based FTP software (e.g., IBM Communications Server FTP), the FTP logonid should be configured as an OS/390 UNIX user as discussed in *Section 2.5.2.6.2, Unprivileged Users and Groups*.

If the FTP software requires the OS/390 FTP user to have access to TSO (e.g., KNET), the logonid will be assigned the appropriate fields for TSO users as discussed in *Section 3.2.2.1, Interactive Users*. The limiting of TSO commands will be done using a TSO command table to associate the required TSO commands with the user's logonid. The logonid will be assigned the following additional field setting to limit access to required commands only:

```
TSOCMDS(command-table)
```

For further information and assistance on the development and assignment of TSO command tables in ACF2, contact DISA FSO.

While the use of FTP logonids with non-expiring passwords is discouraged, it is the customer's decision to accept the risks as discussed in *Section 3.1.2.7.1, Risks*. If the customer is willing to officially acknowledge the risks and implement mitigating controls as discussed in *Section 3.1.2.7.2, Mitigating Controls*, the use of FTP logonids with non-expiring passwords on ACF2 systems shall be permitted. The logonid is to be assigned the following field settings to exempt the password from expiration and to force logging of all activity:

```
MAXDAYS(0)
TRACE
```

3.2.2.8 MCS Console Userids

Specify fields in the UID string that can be used to associate users with particular security categories and specific values for the categories associated with MCS consoles. If *autolog* is allowed, define a second set of values and permit the *autologged* consoles to have *read* access to the CONTROL, DISPLAY, MONITOR, STOPMN, STOPTR, and TRACK commands.

The ACF2 userid profile for an MCS console is to be created with only the fields required to allow the console to *autolog*. Do not define TSO or other segments not required for the operation of the console. Do not permit the userid to access any resources except MVS.MCSOPER.*consolename* in the OPERCMDS class.

```
SET LID

INSERT consname DATA('MCS console description') -
        PASSWORD(password) -
        GROUP(congroup)
```

3.2.2.9 OS/390 System Operator Userids

In order to control which OS/390 system operator can issue which commands, each operator is assigned a personal userid, and the console definition requires operators to log on prior to entering OS/390 commands. If an operator already has a userid, that userid may be used for the purpose. Otherwise a new userid is to be defined, as specified in *Section 3.2.2.1, Interactive Users*.

Normally several operators at a site have similar duties, responsibilities, and roles, and require the ability to enter the same OS/390 system commands. Where such a group of operators exists, define a sub-string of the UID string that can be used in rule sets instead of specifying the individual operator logonids.

3.2.3 Password Controls

3.2.3.1 Password Guidelines

The site shall utilize the capabilities of the ACF2 logonid (LID) record settings, the GSO PSWD record, and (optionally) the password validation exit (refer to *Section 3.2.3.2, Password Exit Processing*) to enforce the password requirements specified in *Section 3.1.3.1, Password Guidelines*. Several of the password requirements can be enforced through the use of standard ACF2 mechanisms. The requirements that are not already enforceable by ACF2 are as follows in the table below entitled *Password Requirements Not Enforced by ACF2*.

Table A-31. PASSWORD REQUIREMENTS NOT ENFORCED BY ACF2 (3.2.3.1)

PASSWORD REQUIREMENTS NOT ENFORCED BY ACF2
No words found in standard dictionaries will be used.
At least one alphabetic, numeric, and special character will be used.[11]
Each character of the password will be unique.
Passwords will contain no consecutive characters (e.g., 12, AB).
Passwords cannot be reused within 10 password changes.
Will not contain the user's name, userid (LID), or telephone number.[12]

[11] ACF2 can allow or disallow passwords containing all-numerics, but it cannot require them. Enforcing the DISA requirement that passwords contain at least one alphabetic, numeric, and special character requires the use of an exit.

[12] ACF2 can prevent the user's LID (userid) from being used as a password. Restricting the telephone number or the user's name requires the use of an exit.

3.2.3.2 Password Exit Processing

As indicated previously (refer to *Section 3.1.3.2, Password Exit Processing*), the site may use the ACF2 password exit to extend the capabilities of ACF2 to enforce any or all of the password requirements not already enforced by the ACP. If implemented, the following GSO EXITS record field should be used to implement these controls:

NEWPXIT(module)

3.2.4 Special Privilege Access

The special privileges discussed in this section are all of an extremely sensitive nature and will be rigidly controlled. The number of authorized users granted these privileges will be kept to an absolute minimum. Their use will be fully documented. The IAO will maintain the written request, justification, and authorization.

3.2.4.1 Access Control Product Modification Privileges

The following user privileges allow update of the ACF2 databases for administering users, data set access rules, and Infostorage records. When granted to a logonid, restrict the scope of the following privileges using an associated **SCPLIST** (scope list) record:

ACCOUNT
LEADER
SECURITY

- *(ACF0750: CAT II) The IAO will ensure logonids with the ACCOUNT, LEADER, SECURITY attributes are restricted by a SCPLIST attribute that restricts authority based on job function and area of responsibility.*

If a logonid is granted the SECURITY privilege, it is mandatory that RULEVLD and RSRCVLD attributes will also be specified for the logonid.

- *(ACF0760: CAT II) The IAO will ensure Logonids with the SECURITY attribute have the RULEVLD and RSRCVLD attributes specified.*

The following privileges cannot be scoped, and will be restricted exclusively to a site IAO:

ACCTPRIV
REFRESH

- *(ACF0770: CAT II) The IAO will ensure Logonids with the REFRESH attribute are only reserved for use by the IAO/IAM.*

- *(ZTSOA040: CAT II) The IAO will ensure Logonids with the ACCTPRIV attribute are only reserved for use by the IAO/IAM.*

3.2.4.2 Audit Privileges

The following user privileges allow viewing of the ACF2 databases for the purpose of inspecting users, data set access rules, and Infostorage records. When granted to a logonid, restrict the scope of the following privileges using an associated SCPLIST (scope list) record:

AUDIT
CONSULT

- *(ACF0780: CAT II) The IAO will ensure that logonids with the AUDIT or CONSULT attributes are restricted by a SCPLIST attribute that restricts authority based on job function and area of responsibility.*

The READALL privilege is available for actual auditing of system data. It gives the capability of looking at every data set on the system despite the data set rules. Its use is strongly discouraged. Always grant access through the use of standard data set access rules. Under no circumstances will the privilege be used as a convenience to the person maintaining the rule sets. Only use this privilege when absolutely necessary, and only give it to auditors. Remove the privilege once the audit is complete. Fully document the granting and revoking of the access.

- *(ACF0790: CAT II) The IAO will ensure that procedures are in place to control Logonids with the READALL attribute.*

3.2.4.3 Tape Label Bypass Privileges

Tape label bypass (BLP) privileges will be restricted at the user level. Specify one of the following two logonid privileges to grant a user access to BLP processing:

User LID Record: TAPE-LBL
TAPE-BLP

It is possible to grant selected programs to bypass tape label processing regardless of the BLP-related privilege of the logonid executing the program. This capability will not be used due to the requirement that accounting of BLP processing be done at the user level. Do **not** utilize the GSO **BLPPGM** record.

- *(ACF0800: CAT II) The IAO will ensure Logonids with the TAPE-LBL or TAPE-BLP are kept to a minimum and are controlled and documented.*

Refer to *Section 3.1.4, Special Privilege Access,* for further information.

3.2.4.4 Other Sensitive Privileges

For other sensitive privileges, use the following controls:

- *(ZTSOA040: CAT II) The IAO will ensure that special privilege MOUNT is assigned only on an as needed basis for LOGONIDS associated with STCs and LOGONIDS that need to execute TSO in batch.*

- *(ZTSOA040: CAT II) The IAO will ensure that access to the special privilege OPERATOR is kept to a minimum and is controlled and documented.*

- *(ZTSO0030: CAT II) The IAO will strictly control and limit access to TSOAUTH privileges. Authorization is restricted to authorized personnel; and justification for access is documented.*

- *(ACF0830: CAT II) The IAO will ensure that access to the special privilege ALLCMDS is kept to a minimum and is controlled and documented.*

- *(ACF0840: CAT II) The IAO will ensure that access to the special privilege PPGM is kept to a minimum and is controlled and documented.*

- *(ACF0640: CAT II) The IAO will ensure that only STC in the trusted STC list can have the NON-CNCL attribute and other STCs having this attribute are approved by the site DAA.*

Refer to *Section 3.1.4, Special Privilege Access,* for further information.

3.2.5 Resource Controls

Where this section talks about compiling a rule set, it is to be understood that the rules shown are to be merged with any existing rules in the rule set rather than replacing them. Refer to *Section 3.1.5. Resource Controls*, for further information.

3.2.5.1 Data Set Controls

All data sets are to be fully protected using data set rules. Rule sets are to provide discrete access to all users requiring access. They prevent any access not specifically intended by the owner of the data, since the ACF2 system is in ABORT mode.

Unless required for access to execute certain routines from system-level libraries (e.g., SYS1.LINKLIB), rule sets are not to include explicitly permissive blanket rules such as the following:

```
        - UID(*) READ(A) EXEC(A)
```

Restrict the use of global access to data sets to the minimal number of libraries. Certain data sets, such as general purpose load libraries, may use global permissions, but these should be restricted to the appropriate level of access (e.g., EXEC).

Restrict data sets that are in the Linklist only to systems, security, and audit personnel since MVS grants EXEC-level access implicitly by the program's presence in the Linklist. However, allowing EXEC-level access to all users does not create an exposure. Protect libraries that are APF-authorized, but are not in the Linklist, so that only the required users have access to these programs. Please note that CA-EXAMINE users (systems, security, and auditors) require *read*-level access to these libraries based on the way that CA-EXAMINE opens the data sets.

Great care and consideration needs to go into defining the access given to the various data sets. As an example, by giving *read* access to the CA-1 parameter/control file to all users, unauthorized personnel may be able to determine how CA-1 security is set up, thus jeopardizing the entire tape library. The Security staff, along with the Systems staff, are to work together to define the access needed and restrict the level of access appropriately.

The IAO maintains the rule sets. The IAO may delegate maintenance of a rule set to the Data Owner. The scope of this individual's privileges are to be restricted using an SCPLIST (scope list).

Refer to *Section 3.1.5.1, Data Set Controls*, for further information.

3.2.5.2 Volume Controls

Volumes requiring volume-level protection are to be controlled using a combination of the GSO SECVOLS and RESVOLS records. Identify the volume(s) to be protected in the SECVOLS record. The RESVOLS record is to be created/updated to specifically exclude these volumes. ACF2 controls both DASD and tape volumes via these mechanisms.

3.2.5.3 Sensitive Utility Controls

Access to sensitive utilities will be strictly controlled. Access to the data sets in which the utilities reside will be controlled through the use of data set rules. Execution access will be controlled and monitored through the use of the MAINT, PPGM, and LOGPGM facilities.

Maintenance utilities will be controlled as previously described. (Refer to *Section 3.1.2.5, Special Storage Management Users*.) A GSO MAINT record will exist for each library containing maintenance utilities. Each record will identify the appropriate special user logonid and the programs that it is permitted to process. The associated special user logonid will contain the MAINT attribute to allow it to execute the identified utility programs. It is imperative to note that access rule validation or SMF logging will not be performed for these utility and logonid combinations.

Sensitive utilities, which are to be available to a limited number of users, will be identified in the Protected Program List (the GSO PPGM record). Authority to execute the identified programs is identified by the logonid record PPGM attribute. Standard access rule validation and SMF logging will be performed by ACF2. To restrict access for the utilities so that only certain utilities can be executed by certain individuals, the DSNPOST exit may be used. The exit will use the following process:

(1) Programs that are to be restricted are coded in the GSO PPGM record. Users will not be given the PPGM privilege as a standard.

(2) Based upon a PPGM violation, the exit code will validate an ACF2 resource rule type of PGM to further interrogate the use of the utility. Based upon the action in the PGM resource rule, the user will be allowed or disallowed the usage of the program.

The resource rules for program validation will look like the following:

```
$KEY(pgm-name) TYPE(PGM)
UID(uid-string) ALLOW
```

Standard data set access rules are required and will be written for the library or libraries containing the utility programs.

Audit access to protected programs considered sensitive in nature. These programs will include, at a minimum, those specified in *Section 3.1.5.3, Sensitive Utility Controls.*

Sensitive utilities that are to be generally available, but whose use is to be audited, will be identified in the GSO LOGPGM record. The default values supplied by ACF2 will not be removed from the record. No special logonid record attribute is required to execute these programs. Standard access rule validation and SMF logging will be performed by ACF2. Standard data set access rules are required and will be written for the library or libraries containing the utility programs.

- *(ACF0870: CAT II) The IAO will ensure access to sensitive utilities is protected by ACF2 by using the resource rule TYPE(PGM). Only appropriate personal are to access and all access is logged.*

3.2.5.4 Dynamic List Controls

Dynamic list controls are provided via resources in the FACILITY resource class. When protecting the facilities for dynamic lists via the FACILITY class, use the following controls:

(1) Prevent access to these resources by default, and log all access. Create generic and specific resource rules as follows:

```
$KEY(CSVAPF) TYPE(FAC)
        - UID(-)
        MVS.SETPROG.FORMAT.DYNAMIC UID(-)
```

```
          MVS.SETPROG.FORMAT.STATIC UID(-)

  $KEY(CSVDYNEX) TYPE(FAC)
          - UID(-)

  $KEY(CSVDYNL) TYPE(FAC)
          - UID(-)
          UPDATE.LNKLST UID(-)
```

(2) The required access to specific resources is to be discretely granted to specific systems users. Restrict this access to the absolutely minimum number of personnel, and log all access. Sample rules are as follows:

```
  $KEY(CSVAPF) TYPE(FAC)
          SYS1.NEWLIB UID(-)
          SYS1.NEWLIB UID(sysprog) SERVICE(READ) LOG
```

3.2.5.5 MCS Console Controls

MCS console controls are provided via resources in the CONSOLE, FACILITY, OPERCMDS, and TSOAUTH resource classes. When protecting the facilities for MCS consoles via these classes, use the following controls:

(1) Prevent access to these resources by default, and log all access. Create generic and specific resource rules as follows:

```
  $KEY(consname) TYPE(CON)
          UID(-)
          UID(opermask) SERVICE(READ) LOG

  $KEY(MVS) TYPE(OPR)
          UID(-)

  $KEY(CONSOLE) TYPE(TSO)
          UID(-)
```

(2) The user profile for each real MCS console is to be granted *read* access to the corresponding console resource:

```
  $KEY(consname) TYPE(CON)
          UID(consname) ACCESS(READ) PERMIT
```

(3) The user profiles for operators and systems programmers allowed to use each real MCS console should be granted *read* access to the corresponding console resource:

```
$KEY(consname) TYPE(CON)
        UID(opermask) ACCESS(READ) PERMIT
```

(4) At the discretion of the IAO, users may be allowed to use the TSO **CONSOLE** command, subject to the restrictions in *Section 3.1.5.5, MCS Console Controls, Section 3.1.5.6, OS/390 System Command Controls,* and *Section 3.2.5.6, OS/390 System Command Controls*. To grant the access, issue the following ACF2 commands:

```
SET PROFILE(USER) DIV(OPERPARM)
INSERT userid AUTH(INFO)
```

and compile the following rule sets:

```
$KEY(MVS) TYPE(OPR)
        MCSOPER.userid UID(userid) SERVICE(READ) LOG

$KEY(CONSOLE) TYPE(TSO)
        UID(opermask) SERVICE(READ) LOG
```

3.2.5.6 OS/390 System Command Controls

OS/390 system command controls are provided via resources in the OPERCMDS resource class. When protecting the facilities for OS/390 system commands via the OPERCMDS class, use the following controls:

(1) Prevent access to the OS/390 resources by default, and log all access. Create generic and specific resource rules with logging as required using the resources defined in *Table A-29, Controls on OS/390 System Commands*. For example:

```
$KEY(MVS) TYPE(OPR)
UID(-)
ACTIVATE UID(-)
CANCEL.JOB.- UID(-) LOG
CONTROL.- UID(-) SERVICE(READ)
DISPLAY.- UID(-) SERVICE(READ)
MONITOR UID(-) SERVICE(READ)
STOPMN UID(-) SERVICE(READ)
```

(2) Only grant access to OS/390 system commands to the extent documented in the installation SOP. Additional profiles are to be defined similarly to those in *Paragraph (1)* above if the existing resource names are too specific or too generic for the controls in the SOP. The rules include the SERVICE (level) and PERMIT/LOG values specified in the SOP, or LOG if not specified.

The following is an example of granting user *userid* permission to issue commands against jobs with names beginning *pfx*, after obtaining permission from the IAO:

```
    $KEY(MVS) TYPE(OPR)
    CANCEL.JOB.pfx* UID(userid) SERVICE(UPDATE) LOG
    MODIFY.JOB.pfx* UID(userid) SERVICE(UPDATE) LOG
    STOP.JOB.pfx* UID(userid) SERVICE(UPDATE) LOG
```

The following is an example of granting to operators whose UID string matches *opergrp* permission to issue ROUTE commands to *sysid* during PRIME shift, after obtaining permission from the IAO:

```
$KEY(MVS) TYPE(OPR)
ROUTE.CMD.sysid UID(opergrp) SERVICE(READ) LOG
SHIFT(PRIME)
```

3.3 RACF

3.3.1 SETROPTS

The following table depicts the STIG required values for the RACF Standard Global Options (SETROPTS) records. The options specified are STIG requirements and each site can choose to be more restrictive.

NOTE: Values listed are deviations from product default settings. Values not listed are to be the default values for the product.

Table A-32. REQUIRED GLOBAL OPTIONS (SETROPTS) - RACF (3.3.1)

REQUIRED GLOBAL OPTIONS (SETROPTS) - RACF		
OPTION	DESCRIPTION	REQUIRED VALUE
ADSP	Automatic data set protection	NOADSP
AUDIT	Logging RACF command and RACDEF SVC activity	AUDIT(*)
CLASSACT	General resource protection	The following classes will be activated on all systems: DATASET USER GROUP The following class will be activated only if no tape management system is installed on the system: TAPEVOL
CMDVIOL	Logging of RACF command violations	CMDVIOL
EGN	Enhanced generic naming	EGN

ERASE	Erasure of scratched or released DASD data set space.
	CAUTION: Use of the ERASE feature can cause considerable system overhead affecting system performance. ERASE may be enabled in DATASET profiles which would afford more granular control.
		Unclassified Systems:
		ERASE()
		Classified Systems:
		ERASE(ALL)

REQUIRED GLOBAL OPTIONS (SETROPTS) - RACF		
OPTION	DESCRIPTION	REQUIRED VALUE
GENCMD	Generic profile creation	GENCMD(*) This option does not apply to the following resource classes: CCICSCMD GLOBAL KERBLINK PROGRAM REALM All group resource classes (e.g., GCICSTRN, GDASDVOL, etc.)
GENERIC	Generic profile checking	GENERIC(*) This option does not apply to the following resource classes: CCICSCMD GLOBAL KERBLINK PROGRAM REALM All group resource classes (e.g., GCICSTRN, GDASDVOL, etc.)
GRPLIST	List-of-Groups authority checking	GRPLIST
INACTIVE	Unused userid interval	35 days
INITSTATS	Records RACINIT statistics	INITSTATS
JES(BATCHALLRACF)	Forces batch users to identify themselves to RACF	JES(BATCHALLRACF)
JES(EARLYVERIFY)	JES userid early verification	JES(EARLYVERIFY)
JES(XBMALLRACF)	Support for execution batch monitor	JES(XBMALLRACF)
OPERAUDIT	Logging activities of users with the OPERATIONS attribute	OPERAUDIT
PASSWORD (HISTORY)	Number of previous passwords	10
PASSWORD (INTERVAL)	Maximum password change interval	90 days

REQUIRED GLOBAL OPTIONS (SETROPTS) - RACF		
OPTION	DESCRIPTION	REQUIRED VALUE
PASSWORD (REVOKE)	Consecutive password verification attempts	3

PASSWORD (RULEn)	Password syntax rules	LENGTH(8)
		ALPHANUM(1:8)

PASSWORD (WARNING)	When password expiration message is issued	10
PROTECTALL	RACF-protect all data sets	PROTECTALL
REALDSN	Places actual data set names in messages and SMF records	REALDSN
RETPD	Selects security retention period for tape data sets	99999
RVARYPW	Sets the RVARY passwords	Site defined. Must change default value. To be set in accordance with standard password guidelines.
SAUDIT	Logging of activity of users with SPECIAL attribute	SAUDIT
SECLEVELAUDIT	Auditing for security levels	NOSECLEVELAUDIT
TAPEDSN	Activates tape data set protection	TAPEDSN
TERMINAL	Universal access authority for terminals	READ
WHEN(PROGRAM)	Program control	WHEN(PROGRAM)

*Global Access Table Information:

The use of the RACF Global Access Table option (GLOBAL in SETROPTS) is optional for each site and may improve system performance.

When access is requested, the Required Global Access Table is checked first because it resides in memory. By placing resources that are frequently accessed and have a UACC of *read* in this table, no further checking is made **if** the requested access is granted. If no entry exists in the Global Access Table, or the desired access is greater than specified in the table, a search is then made of the RACF database.

While the use of the Global Access Table is a site option, the decision to use it should be carefully made and will consider the following:

(1) Only frequently accessed resources should be considered.

(2) No *RACLISTed* resources will be included because these requests bypass the table.

(3) Only widely available resources will be included (e.g., SYS1.BRODCAST, SYS1.HELP, etc.).

(4) Any resources that require logging and/or audit trails will not be included in the Global Access Table.

- *(RACF0250: CAT II) The IAO will ensure that ADSP SETROPTS value is set to NOADSP. ADSP indicates that RACF automatically creates discrete data set profiles to protect data sets created by users having this attribute.*

- *(RACF0260: CAT II) The IAO will ensure that AUDIT SETROPTS value is set to AUDIT(*) indicating that RACF sets all classes to do auditing of uses of the RACDEF SVC and all changes made to profiles by RACF commands.*

- *(RACF0270: CAT II) The IAO will ensure that CLASSACT SETROPTS value is set to values defined in the above table. See above table for the values.*

- *(RACF0280: CAT II) The IAO will ensure that CMDVIOL SETROPTS value is set to CMDVIOL to log violations to RACF commands.*

- *(RACF0290: CAT II) The IAO will ensure that EGN SETROPTS value is set to EGN this allows the generic character ** when you define dataset profiles.*

- *(RACF0300: CAT II) The IAO will ensure that ERASE SETROPTS value is set to ERASE() on unclassified systems. On classified systems it is set to ERASE(ALL) this allows DASD datasets to be erased when deleted. See above table for additional information.*

- *(RACF0310: CAT II) The IAO will ensure that GENCMD SETROPTS value is set to GENCMD(*) this activates generic profile command processing for the dataset class and all classes defined in the CDT.*

- *(RACF0320: CAT II) The IAO will ensure that GENERIC SETROPTS value is set to GENERIC(*) this activates generic profile checking for the dataset class and all classes defined in the CDT.*

- *(RACF0330: CAT II) The IAO will ensure that the TERMINAL SETROPTS value is set to READ; this sets the universal access authority (UACC) associated with undefined terminals.*

- *(RACF0350: CAT II) The IAO will ensure that GRPLIST SETROPTS value is set to GRPLIST. This sets a user's access based on the highest authority in any group to whom the IAO belongs.*

- *(RACF0360: CAT II) The IAO will ensure that INACTIVE SETROPTS value is set to 35 days this specifies the number of days that a user is inactive and still remain valid.*

- *RACF0370: CAT II) The IAO will ensure that INITSTATS SETROPTS value is set to INITSTATS this specifies that statistics available during RACINIT SVC processing are recorded.*

- *(RACF0380: CAT II) The IAO will ensure that JES(BATCHALLRACF) SETROPTS value is set to JES(BATCHALLRACF). This specifies that JES is to test for a userid and password on the job statement or for propagated RACF identification information for all batch jobs.*

- *(RACF0390: CAT II) The IAO will ensure that JES(EARLYVERIFY) SETROPTS value is set to JES(EARLYVERIFY). This specifies that JES is to invoke the system authorization facility (SAF) for jobs that do not qualify for user identification propagation.*

- *(RACF0400: CAT II) The IAO will ensure that JES(XBMALLRACF) SETROPTS value is set to JES(XBMALLRACF). This specifies that JES is set to test for a userid and password on the job statement or for propagated RACF identification information for all jobs run under the execution batch monitor.*

- *(RACF0420: CAT II) The IAO will ensure that OPERAUDIT SETROPTS value is set to OPERAUDIT. This specifies that RACF logs all actions such as accesses to resources and commands for a user who has operations or group operations attribute.*

- *(RACF0430: CAT II) The IAO will ensure that PASSWORD(HISTORY) SETROPTS value is set to 10. This specifies the number of previous passwords that RACF saves for each USERID and compares with an intended new password. If there is a match with one of the previous passwords, or with the current password, RACF rejects the intended new password.*

- *(RACF0440: CAT II) The IAO will ensure that PASSWORD(INTERVAL) SETROPTS value is set to 90 days. This specifies the maximum number of days that each user’s password is valid.*

- *(RACF0450: CAT II) The IAO will ensure that PASSWORD(REVOKE) SETROPTS value is set to 3. This specifies the number of consecutive incorrect password attempts RACF allows before it revokes the USERID on the next incorrect attempt. If you specify REVOKE, ensure INITSTATS are in effect.*

- *(RACF0460: CAT II) The IAO will ensure that PASSWORD(RULEn0) SETROPTS value is set to LENGTH(8) ALPHANUM(1:8). RULEn specifies an individual syntax rule for new passwords that users specify at logon, on the job cards, or on the PASSWORD command.*

- *(RACF0470: CAT II) The IAO will ensure that PASSWORD(WARNING) SETROPTS value is set to 10. WARNING specifies the number of days before a password expires when RACF is to issue a warning message to the user.*

- *(RACF0480: CAT II) The IAO will ensure that PROTECTALL SETROPTS value is set to PROTECTALL. PROTECTALL activates protect all processing. When protect all* *processing is active, the system automatically rejects any request to create or access a data set that is not RACF protected.*

- *(RACF0490: CAT II) The IAO will ensure that REALDSN SETROPTS value is set to REALDSN. REALDSN specifies that RACF is to record in any SMF log records and operator messages, the real data set name used on the data set commands, and in the RACHECK and RACDEF macros.*

- *(RACF0500: CAT II) The IAO will ensure that RETPD SETROPTS value is set to 99999. RETPD specifies the default RACF security retention period for tape data sets. The security retention period is the number of days that RACF protection is to remain in effect for the tape data set.*

- *(RACF0510: CAT II) The IAO will ensure that RVARYPW SETROPTS value is set to a non default value. RVARYPW specifies passwords that an operator is to use to respond with requests to approve RVARY command processing.*

- *(RACF0520: CAT II) The IAO will ensure that SAUDIT SETROPTS value is set to SAUDIT. SAUDIT specifies whether RACF is to log all RACF commands issued by users with the SPECIAL or group SPECIAL attribute.*

- *(RACF0530: CAT II) The IAO will ensure that SECLEVELAUDIT SETROPTS value is set to NOSECLEVELAUDIT. SECLEVELAUDIT specifies the SECLABEL profiles auditing options are used in addition to the auditing options specified for the resource profile. This additional auditing occurs whenever an attempt is made to access a resource protected by a profile that has a security label specified.*

- *(RACF0550: CAT II) The IAO will ensure that TAPEDSN SETROPTS value is set to TAPEDSN. TAPEDSN activates tape data set protection. When tape data set protection is in effect, RACF can protect individual tape data sets as well as tape volumes.*

- *(RACF0560: CAT II) The IAO will ensure that WHEN(PROGRAM) SETROPTS value is set to WHEN(PROGRAM). WHEN(PROGRAM) activates RACF program control, which includes both access control to load modules and program access to data sets.*

For any questions regarding the use of the Global Access Table, refer to the *RACF Security Administrator’s Guide* and the *RACF Auditor’s Guide*.

3.3.2 Userid Controls

Every user will be identified to RACF via each user's unique userid profile. To RACF, a user is an individual (user), a started task, or a batch job. Every userid will be fully identified within RACF with the following fields completed:

NAME	User's name
DFLTGRP	Default group
OWNER	User's profile owner
PASSWORD	Password

RACF will automatically assign the default group as the password if a password is not explicitly coded. Assign a unique password to every userid to prevent unauthorized access by a person who knows the default group for a new userid.

The following subsections define the requirements for defining a RACF user of OS/390 resources. Additional definitions are required for users who will be accessing UNIX System Services resources. Please refer to *Section 2.5, OS/390 UNIX System Services*, for details.

- *(RACF0570: CAT III) The IAO will ensure that Every USERID is uniquely identified to the system. Within the USERID record, the users name, default group, the owner, and the users password fields are completed.*

3.3.2.1 Interactive Users

Apply the principle of *least privilege* in the granting of all user privileges. Grant individual users the minimum resource authorizations necessary to accomplish their assigned functions. Only grant access to system resources as required.

Generate group profiles for all groups of users (e.g., general users, started tasks). These group profiles will identify the minimum privileges necessary for each group of users to accomplish its assigned functions. Associate every user's userid with at least one group profile.

Alternatively, if a user requires additional authority not granted to that user's default group, the user may be connected to one or more additional RACF groups on a permanent or a temporary basis. Because the installation is employing List-of-Groups checking, the user is given the highest level of authority allowed by any associated RACF group.

When a RACF userid initially is added, the Last Access Date (LAST-ACCESS) is set to UNKNOWN. Hence, an unused userid never expires due to inactivity. This results in non-expired, unused userids in the RACF database. Therefore the site should ensure that the local procedures for adding an interactive user include issuing the ALTUSER <userid> RESUME command. This sets the LAST-ACCESS from UNKNOWN to the current date and time and thus enforce the expiration of the userid after 35 days of inactivity, even if the userid is never used.

The following table provides values that will be specified for certain selected fields as user privileges and access are granted:

Table A-33. INTERACTIVE USERS - RACF (3.3.2.1)

INTERACTIVE USERS - RACF		
FIELD	SHORT DESCRIPTION	REQUIRED VALUE
ACCTNUM	Specifies the user's default TSO logon account. Used for all billing.	May be required for Fee-for-Service support.
DATA	Installation data field ***NOTE***: Field may be used for validation by other products (e.g., Netmaster).	Optional
DFLTGRP	User's default group	Will be completed for all users.
NAME(username)	Specifies the 1- to 20-character name of the use.	Will be completed for all users.
OWNER	User's profile owner	Will be completed for all users.
PASSWORD	Logon password for the user	Will be completed for all users.
PROC	Specifies the user's default TSO logon procedure	Will be completed for all TSO users.
SECLABEL	User's current security label	Optional for MAC II Sensitive
USERDATA	Optional user data	Site defined

NOTE: All highly privileged users controlled by NC-PASS will be connected to group **SECURID**. Refer to *Section 6.3.2, NC-PASS for RACF*.

- *(RACF0580: CAT II) The IAO will ensure that Interactive USERIDs have the values specified in the above table completed.*

- *(RACF0610: CAT II) The IAO will ensure that ALL RACF users are assigned a group profile.*

3.3.2.2 Batch Users

The following controls will be applied to production batch userids:

(1) All userids assigned to production batch jobs will be defined as PROTECTED userids. The following command shows the ALTUSER command used to assign the PROTECTED attribute to an existing userid:

```
ALTUSER batch-userid NOPASSWORD
```

- *(RACF0590: CAT II) The IAO will ensure batch jobs that are submitted to the operating system inherit the USERID of the submitter. This identifies the batch job with the user for the purpose of accessing resources. See above text to specify.*

(2) Utilize propagation control for system-level address spaces that submit jobs on behalf of users. Typical candidates include batch job schedulers (e.g., CONTROL-M) and Multiple User Single Address Space Systems (MUSASS) capable of submitting batch jobs (e.g., CICS, IMS, IDMS, ADABASE/COMPLETE). STC and batch userids associated with these tasks will use propagation control. The following command shows the CONTROL-M STC userid being defined to the PROPCNTL resource class:

RDEFINE PROPCNTL *control-m-userid*

The **PROPCNTL** resources class must be active and *RACLISTed* for this protection to be in effect. For example:

```
SETROPTS CLASSACT(PROPCNTL)
SETROPTS RACLIST(PROPCNTL)
```

(3) Define userids associated with batch production jobs using SURROGAT processing. Each individual userid used for batch submission by a scheduler will have its userid coded as an execution-userid with the production scheduler being the surrogate-userid. For example:

RDEFINE SURROGAT *batch-userid*.SUBMIT UACC(NONE)
PERMIT *batch-userid*.SUBMIT CLASS(SURROGAT)
ID(*scheduler-userid*) ACCESS(READ)

- *(RACF0600: CAT II) The IAO will ensure batch jobs that are submitted to the operating system are protected with propagation control. See above text to specify.*

Refer to *Section 3.1.2.2, Batch Users*, for further information.

3.3.2.3 STC Users

Apply the following controls to started tasks:

(1) All started tasks will be assigned a unique userid. Grant the Started Task Control (STC) the minimum access authorities necessary to perform its function.

- *(RACF0620: CAT II) The IAO will ensure that all started tasks are assigned a unique userid.*

(2) All STC userids will be defined as PROTECTED userids. The following command shows the ALTUSER command used to assign the PROTECTED attribute to an existing userid:

ALTUSER *stc-userid* NOPASSWORD

(3) Connect all started task userids to a valid STC Group ID. Only connect STC userids to STC Group IDs.

- *(RACF0650: CAT II) The IAO will ensure that All started tasks are assigned to a group ID.*

(4) All STCs not defined to RACF are run as an undefined user.

(5) All STCs will have a matching profile defined to the STARTED resource class. Matching can be defined as a profile specific to a single STC, a generic profile grouping STCs with the same access requirements, or the generic *catch all* profile of '**'. The STDATA segment of these profiles are to specify a valid RACF userid and Group ID. '=MEMBER' may be used to substitute the PDS member name of the JCL procedure as the userid. For example:

The following RACF command defines a CICS profile:

```
RDEFINE STARTED CICS*.* UACC(NONE) OWNER(admin)
STDATA(USER(=MEMBER) GROUP(STCCICS) TRUSTED(NO))
```

The following RACF command defines a TCP/IP profile:

```
RDEFINE STARTED TCPIP.* UACC(NONE) OWNER(admin)
STDATA(USER(TCPIP) GROUP(STCTCPX) TRUSTED(NO))
```

(6) Certain started tasks performing critical operating system-related functions may be considered trusted for the purpose of data set and resource access requests. For these STCs, all access requests will be honored. The STIG standard for identifying trusted procedures is to define a discrete profile for the STC to the STARTED resource class and to enable the TRUSTED flag within the profile. For example:

```
RDEFINE STARTED JES2.* UACC(NONE) OWNER(admin)
STDATA(USER(JES2) GROUP(STCTRUST) TRUSTED(YES))
```

- *(RACF0660: CAT II) The IAO will ensure that only trusted STCs have the TRUSTED flag enabled within the profile.*

NOTE: STCs identified as trusted are not to be granted the OPERATIONS attribute.

(7) To ensure RACF uses the STARTED resource class and not the ICHRIN03 started procedures table, define a matching generic *catch all* profile of '**' to the STARTED resource class. For example:

> RDEFINE STARTED ** UACC(NONE) OWNER(*admin*)
> STDATA(USER(=MEMBER) GROUP(STC) TRUSTED(NO))

The STC GROUP identified with the generic profile of '**' is not to be granted any explicit data set or resource access authorizations. All access authorizations are to be dependent on the userid.

(8) The ICHRIN03 started procedures table is to be maintained to support recovery efforts in the event the STARTED resource class is deactivated or critical STC profiles are deleted from the STARTED resource class. Ensure that STCs critical to support this recovery effort (e.g., JES2, VTAM, and any appropriate site-specific tasks) are maintained in ICHRIN03 to reflect current STARTED resource class profiles.

Refer to *Section 3.1.2.3, Started Task Control (STC) Users*, for further information.

3.3.2.4 Network Users

Control access to submit jobs from an RJE system with the JESINPUT resource class. Using the JESINPUT resource class, the userids used for batch jobs coming from RJE devices can be associated directly with the remote number defined for the RJE workstation.

Use **NODES** profiles for jobs that are being submitted from another node. These profiles will be used to define authorized NJE nodes from which authorized batch work will be accepted. Since default userids will not be allowed, all users submitting work via NJE processes will require a userid and password coded with the job stream.

- *(RACF0670: CAT II) The IAO will ensure that JESINPUT resource is used to control access to jobs submitted from an RJE and NODE profiles are used to control access to jobs submitted from a NODE.*

3.3.2.5 Special Storage Management Users

Control userids assigned to production maintenance tasks, such as DASD maintenance userids, with program protection. Data set access for backup and recovery will be handled through the DASDVOL and/or GDASDVOL resource classes. Refer to the vendor's product documentation for the specific requirements of the resident DASD management software.

Use of these resource classes will effectively and discretely restrict the privileges of these userids. The OPERATIONS attribute will not be used for such processes, since data sets cannot be accessed if authorization has been explicitly revoked.

- *(RACF0680: CAT II) The IAO will ensure that DASD management USERIDs are controlled with PGM protection through the DASDVOL and GDASDVOL resource classes. Authorization to the OPERATIONS attribute are inappropriate.*

3.3.2.6 Emergency Userids

Define the userid established for emergency access with the GROUP(NONE) specification and the OPERATIONS attribute. Full TSO access will be allowed. Also define the user with full access to all DASDVOL resource classes.

Define the userid established for security administration with the SYSTEM-SPECIAL attribute.

Implement each of these userids with logging enabled to track all activity performed by the userids.

- *(RACF0690: CAT II) The IAO will ensure that Emergency USERIDs are defined with the GROUP(NONE) specification and the OPERATIONS attribute having full access to all DASDVOL resource classes with logging enabled. The userid established for security administration has the SYSTEM-SPECIAL attribute.*

Refer to *Section 3.1.2.6, Emergency Userids*, for further information.

3.3.2.7 FTP Userids

The RACF userid for an FTP user will be created with all required fields as discussed in *Section 3.3.2.1, Interactive Users*. For environments running OS/390 UNIX-based FTP software (e.g., IBM Communications Server FTP), the FTP logonid should be configured as an OS/390 UNIX user as discussed in *Section 2.5.2.6.2, Unprivileged Users and Groups*.

If the FTP software requires the OS/390 FTP user to have access to TSO (e.g., KNET), assign the userid profile the appropriate fields for TSO users as discussed in *Section 3.3.2.1, Interactive Users*.

While the use of FTP userids with non-expiring passwords is discouraged, it is the customer's decision to accept the risks as discussed in *Section 3.1.2.7.1, Risks*. If the customer is willing to officially acknowledge the risks and implement mitigating controls as discussed in *Section 3.1.2.7.2, Mitigating Controls*, the use of FTP userids with non-expiring passwords on RACF systems will be permitted. The following commands exempts the userid’s password from expiration and forces logging of all activity:

```
PASSWORD  USER(ftp-userid)  NOINTERVAL
ALTUSER ftp-userid UAUDIT
```

3.3.2.8 MCS Console Userids

Define a RACF group profile to provide access to the resources needed by all MCS consoles. If *autolog* is allowed, define a second RACF group and permit that group to have *read* access to the CONTROL, DISPLAY, MONITOR, STOPMN, STOPTR, and TRACK commands.

```
ADDGROUP consnoautolog DATA('MCS consoles with no autolog') -
OWNER('SYS1') SUPGROUP('SYS1') NOTERMUACC

ADDGROUP consautolog DATA('MCS consoles with autolog') -
OWNER('SYS1') SUPGROUP('SYS1') NOTERMUACC
```

Create the RACF userid profile for an MCS console with only the fields required to allow the console to *autolog*. Do not define TSO or other segments not required for the operation of the console. Do not permit the userid to access any resources except MVS.MCSOPER.*consolename* in the OPERCMDS class. Use the appropriate RACF group defined above as the default group.

```
ADDUSER consname DATA('MCS console description') -
NAME('MCS console name') -
PASSWORD(password) -
DFLTGRP(congroup)
```

3.3.2.9 OS/390 System Operator Userids

In order to control which OS/390 system operator can issue which commands, each operator is to have a personal userid and the console definition requires operators to log on prior to entering OS/390 commands. If an operator already has a userid, that userid may be used for the purpose. Otherwise a new userid is to be defined, as specified in *Section 3.3.2.1, Interactive Users*.

Normally several operators at a site have similar duties, responsibilities, and roles, and require the ability to enter the same OS/390 system commands. Where such a group of operators exists, define a RACF group and connect the operators to that group, instead of issuing identical permits of individual operators to MCS consoles and OS/390 commands.

3.3.3 Password Controls

3.3.3.1 Password Guidelines

The site is to utilize the capabilities of the RACF SETROPTS PASSWORD controls, and (optionally) the password validation exit (refer to *Section 3.3.3.2, Password Exit Processing*) to enforce the password requirements specified in *Section 3.1.3.1, Password Guidelines*. Several of the password requirements can be enforced through the use of standard RACF mechanisms. The requirements that are not already enforceable by RACF are as follows:

Table A-34. PASSWORD REQUIREMENTS NOT ENFORCED BY RACF (3.3.3.1)

PASSWORD REQUIREMENTS NOT ENFORCED BY RACF
No words found in standard dictionaries are to be used.
At least one alphabetic, numeric, and special character is to be used.[13]
Each character of the password is to be unique.
Passwords are to contain no consecutive characters (e.g., 12, AB).
Are not to contain the user's name, userid, or telephone number.
Passwords cannot be changed more than once every 24 hours without IAO intervention.

3.3.3.2 Password Exit Processing

As indicated previously (refer to *Section 3.1.3.2, Password Exit Processing*), the site may use the RACF password exit to extend the capabilities of RACF to enforce any or all of the password requirements not already enforced by the ACP. If implemented, use the following exit to implement these controls:

RACF Exit: ICHPWX01

3.3.4 Special Privilege Access

The special privileges discussed in this section are all of an extremely sensitive nature, and will be rigidly controlled. Keep the number of users granted these privileges to an absolute minimum.

3.3.4.1 Access Control Product Modification Privileges

Limit the number of userids granted SPECIAL and GROUP-SPECIAL privileges to the minimum number necessary. Delegation of GROUP-SPECIAL processing to other personnel by site-defined Group Administrators is forbidden.

- *(RACF0710: CAT II) The IAO will ensure that users granted SPECIAL privileges are limited to security group and administrators, except those requiring AUDITORS attributes. Documentation providing justification for any additional users are filed.*

[13] RACF can require that passwords contain at least one alphabetic or national character (i.e., $, #, and @) and one numeric character. Enforcing the STIG requirement that passwords contain at least one alphabetic, numeric, and special character requires the use of an exit.

3.3.4.2 Audit Privileges

Limit the number of userids granted the AUDITOR privilege to the minimum number necessary. Specifics regarding the use of the AUDITOR privilege can be found in the *RACF Security Administrators Guide*.

- *(RACF0730: CAT II) The IAO will ensure that users granted AUDITOR privileges are limited to a minimum. Documentation providing justification for any additional are filed.*

3.3.4.3 Tape Label Bypass Privileges

Restrict the bypass label processing (BLP) privilege at the userid level, and grant access to the minimum number of necessary users. Implement the following controls:

(1) If a tape management system (e.g., CA-1) is installed on the system, use the facilities of the resident tape management system to control BLP. In this case, activation of the TAPEVOL class is not required.

(2) If no tape management system (e.g., CA-1) is installed on the system, use the RACF ICHBLP controls to control BLP access. In this case, activation of the TAPEVOL class is required. Grant authorized users of BLP the following authorities:

- The appropriate authority to profile ICHBLP in the FACILITY class
- The appropriate access authority to the requested tape volume(s)

- *(RACF0740: CAT II) The IAO will ensure that users granted TAPE BYPASS LABEL PROCESSING (BLP) privileges are limited to a minimum. Documentation providing justification for this privilege are filed.*

For further information, refer to *Section 3.1.4, Special Privilege Access*, *Section 3.3.1, Standard Global Options (SETROPTS)*, and to the section for the appropriate tape management system.

3.3.4.4 Other Sensitive Privileges

RACF controls a number of other privileges in the TSOAUTH general resource class. Do not grant the Device Mount privilege to on-line TSO users. It may be granted to STC userids that execute TSO in batch on an as-needed basis.

TSOAUTH privileges such as OPER and ACCT, as well as access to the TSO/E CONSOLE facility, will be strictly controlled.

- *(ZTSO0030: CAT II) The IAO will strictly control and limit access to TSOAUTH privileges. Authorization is restricted to authorized personnel and justification for access is documented.*

RACF provides the ability to limit the commands that a TSO user can issue through userid keywords. If command limiting is implemented, the ability to bypass the command limiting will be strictly controlled, and will only be granted to selected users.

The ability to execute privileged programs will be strictly controlled, and will be permitted to the minimum number of users. The IAO will maintain the documentation justifying the requirement to execute these programs.

Limit the number of userids granted OPERATIONS and GROUP-OPERATIONS privileges to the minimum number necessary. Delegation of GROUP-OPERATIONS processing to other personnel by site-defined Group Administrators is forbidden.

- *(RACF0720: CAT II) The IAO will ensure that the number of users granted OPERATIONS privileges are limited and documentation for justification for users outside of operators and system programmers are maintained.*

Refer to *Section 3.1.4, Special Privilege Access,* for further information.

3.3.5 Resource Controls

3.3.5.1 Data Set Controls

Data set controls are provided via the DATASET resource. All data sets are to be fully protected. Protection by default is to be globally enabled (i.e., PROTECTALL). The universal access parameter (UACC) is to be defined as NONE for all data set profiles, since undefined users have access to resources permitted through the use of the UACC. For any data sets requiring global access, use the PERMIT ID(*) command structure. Only permit data set access to users requiring access via data set profiles.

Restrict the use of global data set access to the minimum number of libraries. Certain data sets, such as general purpose load libraries, may use global permissions, but these should be restricted to the appropriate level of access (e.g., FETCH).

Data sets that are in the Linklist should be restricted only to systems, security, and audit personnel since MVS grants FETCH-level access implicitly by the program's presence in the Linklist. However, allowing FETCH-level access to all users does not create an exposure. Libraries that are APF authorized, but are not in the Linklist, are protected so that only required users have access to these programs. Please note that CA-EXAMINE users (systems, security, and auditors) require *read*-level access to these libraries based on the way that CA-EXAMINE opens the data sets.

Great care and consideration needs to go into defining the access given to the various data sets. As an example, by giving *read* access to the CA-1 parameter/control file to all users, unauthorized personnel may be able to determine how CA-1 security is set up, thus jeopardizing the entire tape library. The Security staff, along with the Systems staff, are to work together to define the access needed and restrict the level of access appropriately.

3.3.5.2 Volume Controls

Volume controls are provided via the DASDVOL resource class. Permit access to volumes for which volume-level protection will be provided (rather than data set-level protection) only to users requiring access.

When protecting volumes via the DASDVOL class, use the following controls:

(1) Prevent access to volumes by default. Create a generic DASDVOL profile of "*" with UACC(NONE).

(2) Individual users will be discretely granted the required accesses to specific volumes. The access granted will be the minimum required by users to perform their respective assigned duties.

- *(RACF0760: CAT II) The IAO will ensure that users granted DASDVOL resource class are granted at the userid level only. The DASDVOL profile is limited and requires UACC(NONE). Documentation providing justification for this resource is filed.*

3.3.5.3 Sensitive Utility Controls

Access to sensitive utilities will be strictly controlled. Utility program controls are provided via the PROGRAM resource class. Control access to the data sets in which the utilities reside through the use of data set access permission.

Control maintenance utilities as previously described. (Refer to *Section 3.1.2.5, Special Storage Management Users*.) The ability to execute privileged programs will be strictly controlled, and will be permitted to the minimum number of users.

Audit access to protected programs considered sensitive in nature. These programs will include, at a minimum, those specified in *Section 3.1.5.3, Sensitive Utility Controls.*

The libraries in which sensitive programs and utilities can reside either are part of the system Linklist (they are publicly available), or they reside in libraries that are not in the Linklist (they are considered private libraries for the purpose of program protection). The methods used to protect these programs vary based on whether they are public or private.

(1) Use RACF program controls to control programs and utilities that reside in public (Linklist) libraries:

   (a) Define a profile for each program name and alias in the PROFILE general resource class. Each profile will identify the library that contains the program, and the volume on which the library resides. The profile will also include the AUDIT option to ensure auditing of the use of the program.

(b) Limit user access to each library that contains sensitive programs and utilities to *read* access in the associated DATASET class profile. This protects the programs from being copied, renamed, and then executed.

(2) Programs and utilities that reside in private (non-Linklist) libraries can only be controlled as execute-controlled libraries. This is done by using DATASET class profiles, if RACF, Version 1, Release 8.1, and DFP, Version 3, Release 1.0 or later, are installed. For these libraries, grant only *execute* access to users. *Read* access will not be granted.

Do not confuse RACF program controls with the RACF Program Access to Data Sets (PADS) feature. PADS is a data set control feature, not a program control feature.

- *(RACF0770: CAT II) The IAO will ensure that Access to sensitive utilities are limited and logged. A letter justifying any additional access is filed.*

3.3.5.4 Dynamic List Controls

Dynamic list controls are provided via resources in the FACILITY resource class. This class should already be active and use generic masking, but the sample commands shown below include the relevant SETROPTS commands[14] for the sake of completeness. When protecting the facilities for dynamic lists via the FACILITY class, use the following controls:

(1) Prevent access to these resources by default, and log all access. Create generic and specific profiles as follows:

```
SETROPTS GENERIC(FACILITY)
RDEFINE FACILITY CSVAPF.** AUDIT(ALL) UACC(NONE)
RDEFINE FACILITY CSVAPF.MVS.SETPROG.FORMAT.DYNAMIC
        AUDIT(ALL) UACC(NONE)
RDEFINE FACILITY CSVAPF.MVS.SETPROG.FORMAT.STATIC
        AUDIT(ALL) UACC(NONE)
RDEFINE FACILITY CSVDYNEX.** AUDIT(ALL) UACC(NONE)
RDEFINE FACILITY CSVDYNL.** AUDIT(ALL) UACC(NONE)
RDEFINE FACILITY CSVDYNL.UPDATE.LNKLST
        AUDIT(ALL) UACC(NONE)
SETROPTS CLASSACT(FACILITY)

SETROPTS RACLIST(FACILITY) REFRESH
```

- *(RACF0244: CAT II) The IAO will ensure that FACILITY resource class is active.*

[14] The SETROPTS REFRESH is only shown once; it must be repeated as necessary.

(2) The required access to specific resources is to be discretely granted to specific systems users. Restrict this access to the absolutely minimum number of personnel, and log all access. Sample commands are as follows:

```
RDEFINE FACILITY CSVAPF.SYS1.NEWLIB AUDIT(ALL) UACC(NONE)

PERMIT CSVAPF.SYS1.NEWLIB CLASS(FACILITY) ID(sysprog)
ACCESS(READ)
```

3.3.5.5 MCS Console Controls

MCS console controls are provided via resources in the CONSOLE, OPERCMDS, and TSOAUTH resource classes. These classes should already be active, and OPERCMDS should already use generic masking, but the sample commands shown below include the relevant SETROPTS commands[15] for the sake of completeness. When protecting the facilities for MCS consoles via these classes, use the following controls:

(1) Prevent access to these resources by default, and log all access. Create generic and specific profiles as follows:

```
RDEFINE CONSOLE * AUDIT(ALL) UACC(NONE)
RDEFINE CONSOLE consname AUDIT(ALL) UACC(NONE)
PERMIT consname CLASS(CONSOLE) ID(opergrp) ACCESS(READ)
SETROPTS CLASSACT(CONSOLE)
SETROPTS RACLIST(CONSOLE) REFRESH
SETROPTS GENERIC(OPERCMDS)
RDEFINE OPERCMDS MVS.** AUDIT(ALL) UACC(NONE)
SETROPTS CLASSACT(OPERCMDS)
SETROPTS RACLIST(OPERCMDS) REFRESH
RDEFINE TSOAUTH CONSOLE AUDIT(ALL) UACC(NONE)
SETROPTS CLASSACT(TSOAUTH)
SETROPTS RACLIST(TSOAUTH) REFRESH
```

- *(RACF0248: CAT II) The IAO will ensure that CONSOLE resource class is active.*

(2) The user profile for each real MCS console is to be granted *read* access to the corresponding console resource:

```
PERMIT consname CLASS(CONSOLE) ID(consname) ACCESS(READ)
```

(3) The group and user profiles for operators and systems programmers allowed to use each real MCS console are to be granted *read* access to the corresponding console resource:

```
PERMIT consname CLASS(CONSOLE) ID(opergrp) ACCESS(READ)
```

[15] The SETROPTS REFRESH is only shown once; it must be repeated as necessary.

(4) At the discretion of the IAO, users may be allowed to use the TSO **CONSOLE** command, subject to the restrictions in *Section 3.1.5.5, MCS Console Controls*, *Section 3.1.5.6, OS/390 System Command Controls*, and *Section 3.3.5.6, OS/390 System Command Controls*.

```
ALTUSER userid OPERPARM(AUTH(INFO))
PERMIT MVS.MCSOPER.userid CLASS(OPERCMDS) ID(userid)
    ACCESS(READ)
PERMIT CONSOLE CLASS(TSOAUTH) ID(opergrp) ACCESS(READ)
```

3.3.5.6 OS/390 System Command Controls

OS/390 system command controls are provided via resources in the OPERCMDS resource class. This class should already be active and use generic masking, but the sample commands shown below include the relevant SETROPTS commands[16] for the sake of completeness. When protecting the facilities for OS/390 system commands via the OPERCMDS class, use the following controls:

(1) Prevent access to the OS/390 resources by default, and log all access. Create generic and specific profiles with logging as required using the resources defined in *Table A-29, Controls on OS/390 System Commands*. For example:

```
SETROPTS GENERIC(OPERCMDS)
RDEFINE OPERCMDS MVS.** AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS MVS.ACTIVATE AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS MVS.CANCEL.JOB.** AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS MVS.CONTROL.** AUDIT(ALL(UPDATE))
UACC(NONE)
RDEFINE OPERCMDS MVS.DISPLAY.** UACC(NONE)
RDEFINE OPERCMDS MVS.MONITOR UACC(NONE)
RDEFINE OPERCMDS MVS.STOPMN UACC(NONE)
SETROPTS CLASSACT(OPERCMDS)
SETROPTS RACLIST(OPERCMDS) REFRESH
```

- *(RACF0246: CAT II) The IAO will ensure that OPERCMDS resource class is active.*

(2) Only grant access to OS/390 system commands to the extent documented in the installation SOP. Define additional profiles similarly to those in *Paragraph (1)* above if the existing resource names are too specific or too generic for the controls in the SOP. The RDEFINE statements are to include the AUDIT and UACC values specified in the SOP, or AUDIT(ALL) UACC(NONE) if not specified.

The following is an example of granting a user permission to issue commands against jobs with names beginning ***pfx***, after obtaining permission from the IAO:

[16] The SETROPTS REFRESH is only shown once; it must be repeated as necessary.

```
PERMIT MVS.CANCEL.JOB.pfx* CLASS(OPERCMDS) ID(userid)
    ACCESS(UPDATE)

PERMIT MVS.MODIFY.JOB.pfx* CLASS(OPERCMDS) ID(userid)
    ACCESS(UPDATE)

PERMIT MVS.STOP.JOB.pfx* CLASS(OPERCMDS) ID(userid)
    ACCESS(UPDATE)

SETROPTS RACLIST(OPERCMDS) REFRESH
```

The following is an example of granting group *opergrp* permission to issue ROUTE commands to *sysid* from *consid*, after obtaining permission from the IAO:

```
PERMIT MVS.ROUTE.CMD.sysid CLASS(OPERCMDS) ID(opergrp)
    ACCESS(READ) WHEN(CONSOLE(consid))

SETROPTS RACLIST(OPERCMDS) REFRESH
```

3.4 TOP SECRET

3.4.1 Standard Global Options (Control Options)

The following table depicts the STIG required values for the TSS Control Options records. Default parameter values should be coded for documentation purposes. The options specified are STIG requirements and each site can choose to be more restrictive.

Table A-35. REQUIRED GLOBAL OPTIONS (CONTROL OPTIONS) – TOP SECRET (3.4.1)

REQUIRED GLOBAL OPTIONS (CONTROL OPTIONS) - TOP SECRET		
OPTION	DESCRIPTION	REQUIRED VALUE
ADMINBY	Enables administration information to be recorded for security changes.	ADMINBY
ADSP	Controls global automatic data set protection.	ALL (default) (Pre-always call) YES (MVS, Version 1.x NO (MVS, Version 2.x and above) ***NOTE***: Setting is also dependent on the type(s) of catalogs in use on the system.
AUTH	Controls authorization checking.	OVERRIDE, ALLOVER
AUTOERASE	Controls auto-erase feature necessary to meet NCSC requirements.	Unclassified Systems: Optional Classified Systems: YES CAUTION: Usage will affect performance.
BACKUP	Controls automatic Security File backup.	Site defined ***NOTE***: a time must be specified unless the database is shared and backed up on another system.
BYPASS	Specifies jobs and started tasks that bypass security in an emergency.	As applies to a specific system ***NOTE***: Local changes will be justified in writing with supporting documentation.

REQUIRED GLOBAL OPTIONS (CONTROL OPTIONS) - TOP SECRET		
OPTION	DESCRIPTION	REQUIRED VALUE
CANCEL	Allows TOP SECRET to be canceled via the operating system CANCEL command.	The CANCEL option will not be specified. ***NOTE***: To maintain the integrity of the TOP SECRET environment, the MVS FORCE command will also not be used to terminate the TSS started task.
CPFRCVUND	Identifies whether or not the local node can receive commands transmitted from remote nodes that have not been defined to the CPFNODES list.	NO
DATE	Sets date display format.	MM/DD/YY
DEBUG	Controls debugging feature. Use as directed by CA support.	OFF
DIAGTRAP	Controls diagnostic traps. Use as directed by CA support.	OFF (Ver 5.1 and below) ALL,DEL (Ver 5.2 and above)
DL1B	Controls protection of DBD and PSB for DL/1 batch programs.	NO
DOWN	Controls action taken when TSS address space is inactive.	SB, BW, OW, and either: TW (if users are still defined in SYS1.UADS) - or - TN (if only systems personnel remain defined in SYS1.UADS)
DRC	Modifies or lists particular DRC attributes.	As applies to a specific system
DUFPGM	Identifies programs allowing for extraction or upgrade of INSTDATA.	As applies to a specific system
DUMP	Takes formatted dumps of TSS address space.	As applies to a specific system

REQUIRED GLOBAL OPTIONS (CONTROL OPTIONS) - TOP SECRET		
OPTION	DESCRIPTION	REQUIRED VALUE
EXIT	Installation user exit.	ON ***NOTE***: The Post-Initiation exit point is supplied by SSO Mechanicsburg to support the control of privileged users by NC-PASS. Refer to Section 6.3.3, NC-PASS for TOP SECRET, for further information. ***NOTE***: For non DISA sites this is site defined. ***NOTE***: A review by DISA FSO is required for each exit point activated.
FACILITY	Controls facility processing.	As applies to a specific system. All defined FACILITIES will specify MODE=FAIL.
HPBPW	Days to honor previous batch password.	1-3 days
INACTIVE	Controls users who have been inactive for a specific period.	35 days maximum
INSTDATA	Alters global installation data field.	0
IOTRACE	Controls TSS I/O trace.	OFF
JCT	Identifies JES2 JCT offsets.	As applies to a specific system
JES	Identifies JES2/JES3 subsystems.	NOVERIFY
JOBACID	Controls ACID identification for batch jobs.	Site defined ***NOTE***: To be defined by the IAO.
LOG	Controls incident recording for all facilities.	MSG, SEC9, INIT, SMF
LOGBUF	Allows the maximum number of in-core logging buffers to be used.	32
MODE	Controls processing mode for all facilities.	FAIL

REQUIRED GLOBAL OPTIONS (CONTROL OPTIONS) - TOP SECRET		
OPTION	DESCRIPTION	REQUIRED VALUE
MSG	Alters characteristics of TSS violation messages.	As applies to a specific system ***NOTE***: Local changes will be justified in writing with supporting documentation.
MSUSPEND	Allows Master Security Control ACID (MSCA) to be suspended if password violation occurs.	YES
NEWPW	Selects new password specification rules.	MIN=8, WARN=10, MINDAYS=1, NR=0, ID, TS, SW, RS FA, FN for (Ver 5.3 and above)
NJEUSR	Defines a default ACID for NJE Store-and-Forward nodes. Has no significance on a job's execution node.	NJEUSER(NJESTORE)
NPWRTHRESH	Sets maximum threshold, from 0 to 99, for new passwords to be verified before the complete logon sequence needs restarting.	2
OPTIONS	This parameter replaces optional APARs that have been applied prior to Release 5.1.	4, 33, 34 (Ver 5.2) 4 (Ver 5.3 and above) ***NOTE***: Local changes will be justified in writing with supporting documentation.
PRODUCTS	Specifies special products installed.	TSO/E As applicable to the individual sites ***NOTE***: Local changes will be justified in writing with supporting documentation.
PTHRESH	Specifies password violation threshold.	2
PWEXP	Specifies password expiration interval.	90
PWHIST	Specifies number of previous passwords to be maintained in history file.	10
PWVIEW	Controls display of passwords by administrators.	NO

REQUIRED GLOBAL OPTIONS (CONTROL OPTIONS) - TOP SECRET		
OPTION	DESCRIPTION	REQUIRED VALUE
RECOVER	Controls change recovery. ***NOTE***: Requires the RECFILE DD statement in the TSS STC.	ON
SECTRACE	Controls security diagnostic trace.	OFF ***NOTE***: May be activated on an as-needed basis, only for diagnostic purposes.
SUBACID	Controls on-line job submission.	U, 8
SWAP	Controls TSS address space swapping.	NO
SYSOUT	Spins off TSS activity log; specifies class and destination.	x, LOCAL Class as specified in the Computing Services's *Naming Convention Recommendations*. Class specified is at the sites discretion.
TAPE	Controls tape processing. ***NOTE***: OFF indicates that an External Tape Management System (ETMS) is in use.	OFF
TEMPDS	Controls temporary data set protection.	YES
TIMER	Interval at which data is written from TSS buffers to AUDIT/TRACKING file.	30
VTHRESH	Selects violation threshold and action.	10, NOT, CAN

- *(TSS0249: CAT II) The IAO will ensure ADMINBY control option is set to Adminby to record who when and where information in the ACID security record for administrative changes.*

- *(TSS0250: CAT II) The IAO will ensure ADSP control option is set to (NO) indication that the RACF bit in the DSCB will is not set.*

- *(TSS0260: CAT II) The IAO will ensure AUTH control option is set to (OVERIDE, ALLOVER) TSS merges the user, profile, and all records for its access authorization search.*

- *(TSS0270: CAT II) The IAO will ensure AUTOERASE control option is set to (YES) for Classified systems and is at the sites discretion for Unclassified systems to erase all residual information on DASD.*

- *(TSS0272: CAT II) The IAO will ensure the BACKUP control option specifies a time and BACKUP(OFF) is not specified unless the database is shared and backed up on another system.*

- *(TSS0275: CAT II) The IAO will ensure CANCEL control option is not set to CANCEL. Security administrators do not have the ability to do an O/S CANCEL command to terminate the TSS address space.*

- *(TSS0280: CAT II) The IAO will ensure CPFRCVUND control option is set to (NO). The CPFRCVUND Control Option indicates whether or not the local node can receive commands propagated from nodes which are not defined to the CPFNODES list.*

- *(TSS0310: CAT II) The IAO will ensure the DATE control option is set to MM/DD/YY. The DATE Control Option specifies the format for dates displayed in listing.*

- *(TSS0320: CAT II) The IAO will ensure DEBUG control option is set to (NO). The DEBUG Control Option controls the production of debugging dumps used to determine the cause of abnormal error conditions. Use as directed by CA Support.*

- *(TSS0330: CAT II) The IAO will ensure DIAGTRAP control option is set to (Off) for TSS ver 5.1 and earlier or it is set to (ALL,DEL) for TSS ver 5.2 or greater. DIAGTRAP creates a diagnostic dump. Use as directed by CA Support.*

- *(TSS0350: CAT II) The IAO will ensure DL1B control option is set to (NO). The DL1B Control Option is used to implement PSB and DBD security for IMS batch regions, and to provide access to the TSS application interface program.*

- *(TSS0360: CAT II) The IAO will ensure that DOWN control option is set to(BW,SB,OW) and TW if users are still defined in SYS1.UADS, TN if only systems personnel are defined in SYS1.UADS.*

- *(TSS0380: CAT II) The IAO will ensure that EXIT control option is set to (ON) for DISA sites.*

NOTE: The Post-Initiation exit point is supplied by SSO Mechanicsburg to support the control of privileged users by NC-PASS. Refer to *Section 6.3.3, NC-PASS for TOP SECRET,* for further information. **For non DISA sites this value is site defined.**

- *(TSS0385: CAT I) The IAO will ensure FACILITY control option specify the sub option of MODE=FAIL. The MODE sub option specifies the security mode for the FACILITY.*

- *(TSS0390: CAT II) The IAO will ensure HPBPW control option is set to (3) days maximum. HPBPW Control Option selects the maximum number of days that TSS honors an expired or previous password for batch jobs.*

- *(TSS0400: CAT II) The IAO will ensure INACTIVE control option is set to (35) days maximum. The INACTIVE Control Option selects the number of days before TSS denies an unused ACID access to the system after the ACIDs password has expired.*

- *(TSS0410: CAT II) The IAO will ensure INSTDATA control option is set to (0). The INSTDATA Control Option controls the value of the 4-byte global data installation data area. This value is passed to the security exit developed at a particular site.*

- *(TSS0420: CAT II) The IAO will ensure IOTRACE control option is set to (OFF)). The IOTRACE Control Option controls a diagnostic trace for use by technical support. The trace is produced on the TRACE/LOG data set.*

- *(TSS0430: CAT II) The IAO will ensure JES control option is set to (NOVERIFY)). The JES Control Option indicates whether or not support for the JES Early Verify feature is desired.*

- *(TSS0440: CAT II) The IAO will ensure LOG control option is set to (MSG, SEC9, INIT, SMF). The LOG Control Option identifies the types of events that TSS logs, and specifies whether the events are logged onto the audit tracking file and into the SMF files.*

- *(TSS0450: CAT II) The IAO will ensure LOGBUF control option is set to (32). The LOGBUF Control Option allows the maximum number of in-core logging buffers used by TSS.*

- *(TSS0460: CAT I) The IAO will ensure that MODE control option is set to (FAIL). The MODE Control Option selects the security mode in which TSS operates for all facilities.*

- *(TSS0470: CAT II) The IAO will ensure that MSUSPEND control option is set to (YES). The MSUSPEND Control Option allows the MSCA ACID to be suspended automatically if the password violation threshold is set via the PTHRESH option and that limit is exceeded.*

- *(TSS0480: CAT II The IAO will ensure that NEWPW control option is set to (MIN=8, WARN=10, MINDAYS=1, NR=0, ID, TS, SW, RS) for 5.2 and below (FA, FN) is appended for Ver 5.3 and above. The NEWPW Control Option specifies the rules that TSS l applies when a user selects a new password.*

- *(TSS0490: CAT II) The IAO will ensure that NJEUSER control option is set to (NJESTORE). The NJEUSER Control Option is used to define a default ACID used for NJE store and forward nodes where no other ACID are identified.*

- *(TSS0500: CAT II) The IAO will ensure that NPWRTHRESH control option is set to (2). The NPWRTHRESH Control Option sets the threshold value for the number of attempts allowed for new password verification before complete logon sequence needs restarting.*

- *(TSS0505: CAT II) The IAO will ensure that OPTIONS control option is set in accordance to the STIG, additional OPTIONS entries are justified in writing with supporting documentation.*

- *(TSS0530: CAT II) The IAO will ensure PRODUCTS control option is set to (TSO/E). The site can list any other products at their own discretion. The PRODUCTS Control Option allows the site to list special products that are installed on the system.*

- *(TSS0540: CAT II) The IAO will ensure the PTHRESH control option is set to (2). The PTHRESH Control Option selects a maximum password violation threshold.*

- *(TSS0550: CAT II) The IAO will ensure the PWEXP control option is set to (90). The PWEXP Control Option allows the site to specify a password expiration interval.*

- *(TSS0560: CAT II) The IAO will ensure the PWHIST control option is set to (10). The PWHIST Control Option specifies the number of previous passwords maintained as part of an ACIDs password history file.*

- *(TSS0570: CAT I) The IAO will ensure the PWVIEW control option is set to (NO). The PWVIEW Control Option allows the site to suppress the viewing of a users password.*

- *(TSS0580: CAT II) The IAO will ensure the RECOVER control option is set to (ON). The RECOVER Control Option indicates whether TSS records changes made to the security database onto the recovery file.*

- *(TSS0590: CAT II) The IAO will ensure that SECTRACE control option is set to (OFF). The SECTRACE Control Option activates a diagnostic security trace on the activities of all defined users.*

- *(TSS0600: CAT II) The IAO will ensure that SUBACID control option is set to (U,8). The SUBACID Control Option indicates how TSS derives an ACID for batch jobs that are submitted through an online terminal, from another batch job, or from a started task.*

- *(TSS0610: CAT II) The IAO will ensure that SWAP control option is set to (NO). The SWAP Control Option controls the swapping of the TSS address space by the OS/390 operating system.*

- *(TSS0630: CAT II) The IAO will ensure that TAPE control option is set to (OFF). The TAPE Control Option specifies the type of tape protection in effect at the installation.*

- *(TSS0640: CAT II) The IAO will ensure that TEMPDS control option is set to (YES). The TEMPDS Control Option allows an installation to determine whether or not temporary data sets are protected.*

- *(TSS0650: CAT II) The IAO will ensure that TIMER control option is set to (30). The TIMER Control Option controls the interval at which data is written from TSS buffers to the audit tracking file.*

- *(TSS0730: CAT II) The IAO will ensure the VTHRESH control option is set to (10, NOT, CAN). The VTHRESH Control Option selects an access violation threshold for users, batch jobs and started tasks, and selects the action that TSS takes when the threshold is reached.*

3.4.2 Userid Controls

Every user will be identified to TSS via a TYPE=USER Accessor ID (ACID) record. To TSS, a TYPE=USER ACID definition is used to identify an individual, a started task, or a batch job. Every user type ACID will be fully identified within TSS with the following completed fields:

NAME	User's name

IAOs will ensure that the values for these fields are maintained and current. They will update these fields as needed to reflect such changes as personnel actions, office relations, etc.

The following subsections define the requirements for defining a TOP SECRET user of MVS resources. Additional definitions are required for users who will be accessing UNIX System Services resources. Please refer to *Section 2.5, OS/390 UNIX System Services*, for details.

- *(TSS0740: CAT III) The IAO will ensure that every TYPE=USER ACID is uniquely identified to the system. Within the ACID record, the users NAME field is completed.*

- *(TSS0745 CAT III) The IAO will ensure that Every TYPE=USER ACID is uniquely identified to the system. ACIDS are not shared among multiple users.*

3.4.2.1 Interactive Users

Grant individual users the minimum authorizations necessary to accomplish their assigned functions. Only grant facilities such as the following if needed:

FAC(BATCH)
FAC(TSO)
Other specific on-line systems (ROSCOE, etc.)

Generate shared profile ACIDs (TYPE=PROF) for resources to be shared between multiple interactive users. All departments shall, at a minimum, have one shared profile to define the basic access authorizations for their personnel, and all departmental personnel should be assigned that profile at a minimum. Additional shared profiles should be defined as needed to grant access to other departmental resources (such as departmental shared files and CICS regions) to selected groups of users. As new requirements for resources develop, the resources should be added to existing shared profiles whenever possible.

New shared profiles should be created only when the access requirements do not fit into the existing structure of shared profiles. It is possible that a profile may be associated temporarily with a single ACID until additional users requiring that resource are defined. This is acceptable. However, profiles should not be created for resources that are known to require access by only a single ACID. If a resource is to be granted only to a single user, and never to another user, it should be granted directly to the ACID that requires the access. However, if the resource later must be granted to additional users, then it should be added to an existing or new shared profile as outlined above, and the direct access should be removed from the user ACID.

In certain special and unique instances, an interactive user ACID may require access authorizations or privileges on a temporary basis. In such cases, the authorizations or privileges may be granted directly to the user ACID on a temporary basis. They should be removed from the user ACID when no longer needed.

The following table provides values that will be specified for certain selected fields as user privileges and access are granted:

Table A-36. INTERACTIVE USERS - TOP SECRET (3.4.2.1)

INTERACTIVE USERS - TOP SECRET		
FIELD	DESCRIPTION	REQUIRED VALUE
FAC	Facilities the user is validated to use	BATCH - For batch users TSO - For TSO users NC-PASS - For highly privileged users controlled by NC-PASS. Refer to *Section 6.3.3, NC-PASS for TOP SECRET*, for further information. Other - As necessary
NAME(username)	Specifies the 1- to 32-character name of the user.	Will be completed for all users
PASSWORD	The logon password for the user	Will be completed for all users
INSTDATA	Installation-defined data	Optional
PROF	Profile(s) defining the user's attributes	Will be completed for all users
TSOACCT	Specifies the user's TSO logon account. Used for all billing.	May be required for Fee-for-Service support
TSOLACCT	Specifies the user's default TSO logon account. Used for all billing.	May be required for Fee-for-Service support
TSOAUTH	Used to secure TSO user attributes	Will be completed for all TSO users
TSOLPROC	Specifies the user's default TSO logon procedure	Will be completed for all TSO users
TSOPROC	Specifies the user's TSO logon procedure	Will be completed for all TSO users

NOTE: All highly privileged users controlled by NC-PASS will be granted access to the SECURID ABStract. Refer to *Section 6.3.3*, *NC-PASS for TOP SECRET*, for further information.

- *(TSS0750: CAT II) The IAO will ensure that all interactive ACIDS have the fields specified in the above table completed.*

3.4.2.2 Batch Users

All scheduled production batch user ACIDs will be sourced to the internal reader and defined with a non-expiring password. The password contents will follow the requirements as specified in *Section 3.1.3.1, Password Guidelines*. The BATCH facility will be specified for all routines that will be submitting batch work to the system. Production batch user ACIDs do not require an associated shared profile ACID, and may be directly granted privileges as necessary. However, the guidance regarding the creation of shared profiles vice single-user resource profiles for interactive users (outlined above in *Section 3.4.2.1, Interactive Users*) also applies to batch ACIDs.

Utilize propagation control for system-level address spaces that submit jobs on behalf of users. Typical candidates include batch job schedulers (e.g., CONTROL-M) and multi-user online systems (e.g., CICS, IMS, IDMS, ADABASE/COMPLETE). STC and batch ACIDs with the following attributes will use propagation control:

- The facility of BATCH
- Associated with a Master Facility defined with MULTIUSER and ASUBM

The following command shows the CONTROL-M STC ACID being owned to the PROPCNTL resource class:

```
TSS ADD(deptacid) PROPCNTL(control-m-acid)
```

Refer to *Section 3.1.2.2, Batch Users*, for further information.

- *(TSS0760: CAT II) The IAO will ensure that Batch Job Schedulers (e.g., CONTROL-M) and multi-user online systems (e.g., CICS, IMS, IDMS, ADABASE/COMPLETE) that submit jobs on behalf of users that have the facility of BATCH and are associated with a Master Facility defined with MULTIUSER and ASUBM utilize propagation control.*

- *(TSS0770: CAT II) The IAO will ensure that all batch job ACIDs associated with Batch Job Scheduler have the BATCH facility. Justification is provided if any additional facilities are specified. The ACID associated with the batch job is authorized to the Batch Job Scheduler.*

3.4.2.3 STC Users

All started tasks will be assigned a unique **TYPE=USER** ACID. The Started Task Control (STC) will be granted the minimum authorizations necessary for the STC to function. In addition to the ACID default options, apply the following controls to started tasks:

1. All STC ACIDs will have the STC facility. An STC also may be granted the FAC(BATCH) if it requires the capability to submit batch jobs to the internal reader. It should be noted, however, that this also will allow the STC itself to be executed as a batch job.

2. Each STC ACID will be defined with a password following the requirements as specified in *Section 3.1.3.1, Password Guidelines*. The only exception is that these passwords will be defined as non-expiring. In addition, each STC will have its own unique password. Defining a password for started tasks prevents a user from logging onto a system with the STC ACID.

3. Ensure the OPTIONS control option specifies a value of **4** to disable password checking for STCs. Otherwise operators will be forced to supply a password when STCs are started. Refer to *Section 3.4.1, Standard Global Options (Control Options)* for further details.

4. All STC ACIDs will be sourced to the internal reader. This control will further protect the unauthorized use of STC ACIDs.

   ADD(*stc-acid*) SOURCE(INTRDR)

5. Every STC will be defined to the STC table, associated with a specific procedure, and granted minimum access.

6. All STCs not defined to TSS will fail upon initiation. The following command may be used to associate all undefined STCs with a default action of FAIL:

   TSS ADD(STC) PROCNAME(DEFAULT) ACID(FAIL)

7. Certain started tasks performing critical operating system-related functions may be considered trusted for the purpose of data set and resource access requests. For these STCs, all access requests will be honored. The STIG requirement is to grant the BYPASS privilege or NO***CHK attributes to these STCs.

8. Started task user ACIDs do not require an associated profile ACID, and may be directly granted privileges as necessary.

9. STCs should be granted the NOSUSPEND privilege to exempt an STC's associated ACID from suspension for excessive violations. However, an STC will be canceled for excessive violations.

10. If a valid requirement exists to establish a default STC, the following restrictions also apply:

    a. The IAO will maintain the written request, justification, and authorization.

b. The STC will have no other facilities permitted to it.

c. It will have DSN(*****) ACCESS(NONE).

d. The STC's ACID will be sourced to the internal reader:

   ADD(*stc-acid*) SOURCE(INTRDR)

e. An entry will be made in the STC table identifying the default ACID name as follows:

   TSS ADD(STC) PROCNAME(DEFAULT) ACID(*default name*)

Refer to *Section 3.1.2.3, Started Task Control (STC) Users*, for further information.

- *(TSS0790: CAT II) The IAO will ensure that the default STC ACID is set with a default action of (FAIL).*

- *(TSS0810: CAT II) The IAO will ensure that only trusted STCs are granted the BYPASS privilege.*

- *(TSS0820: CAT II) The IAO will ensure that all STC ACIDS assigned a unique USER ACID, have a corresponding USER ACID defined with the STC FACILITY specified, and have a password generated in accordance with DAA defined requirements, and are sourced to the OS/390 internal reader. All ACIDS with STC FACILITY specified have a corresponding entry defined in the STC RECORD.*

3.4.2.4 Network Users

TSS has the capability to assign a default ACID derived from the name of the physical reader, RJE, or NJE node from which a job is being transmitted. This can be accomplished by using the following control option:

```
FACILITY(BATCH=DEFACID(RDR*TERM))
```

This control option would allow the ACID to be assigned to jobs entering the system from that node without a valid ACID and password. No default ACIDs will be allowed for transmitted jobs. Define all work submitted via physical reader, RJE, or NJE processes with a valid ACID and a password coded in the JCL.

All static batch ACIDs (ACIDs whose passwords never change) originating from a physical reader, RJE, or NJE will be sourced to those readers and/or terminals using the following command:

TSS ADD(*batch-acid*) SOURCE(*device*)

- *(TSS0830: CAT II) The IAO will ensure that all jobs submitted through the RJE process are sourced for submission to restrict the ACID to a specific remote number.*

3.4.2.5 Special Storage Management Users

ACIDs assigned to production storage maintenance tasks, such as DASD management, will be granted the appropriate authorizations necessary to perform their functions. Apply the following controls to storage management ACIDs:

(1) Define all batch ACIDs to the BATCH facility.

(2) Permit access to sensitive programs and utilities using program protection controls, such as the PROGRAM resource class and program pathing.

(3) Permit data set access for backup, recovery, and compaction using the VOLUME resource class. Depending on the storage management software, some data set-level checking may be performed under certain conditions. For such instances, the appropriate data set access authorization should be granted. Refer to the vendor's product documentation for specific requirements.

- *(TSS0840: CAT II) The IAO will ensure that all maintenance ACIDs are controlled through the use of the BATCH facility, program pathing protection, and PRIVPGM resource class.*

3.4.2.6 Emergency Userids

The system emergency administration ACID (an SCA) is to be stored in the safe as the userid capable of performing ACP administration.

The ACID to be used in emergencies for systems programming to resolve problems will be set up with the following attribute:

NAME

This userid is granted access to the TSO and BATCH facilities. All permissions under the TSO AUTH resource class will be permitted. To allow all data to be accessed by this userid, grant the following permission:

TSS PER(*acid*) DSN(*****) ACC(ALL) ACTION(AUDIT)

This will be used in lieu of the NODSNCHK option since auditing of data access can then be performed.

- *(TSS0850: CAT II) The IAO will ensure that all access to emergency ACIDs are limited to resources required to support the specific functions of the owning department and access to these resources are audited.*

Refer to *Section 3.1.2.6, Emergency Userids*, for further information.

3.4.2.7 FTP Userids

The TOP SECRET ACID for an FTP user are to be created with all required fields as discussed in *Section 3.4.2.1, Interactive Users*. For environments running OS/390 UNIX-based FTP software (e.g., IBM Communications Server FTP), the FTP ACID should be configured as an OS/390 UNIX user as discussed in *Section 2.5.2.6.2, Unprivileged Users and Groups*.

If the FTP software requires the OS/390 FTP user to have access to TSO (e.g., KNET), the ACID is to be assigned the appropriate fields for TSO users as discussed in *Section 3.4.2.1, Interactive Users*. The limiting of TSO commands are to be done using the COMMAND keyword to associate the required TSO commands with the user's ACID. The ACID is to be assigned the following additional attribute to limit access to required commands only:

```
COMMAND(TSO,(<command1>,<command2>,...))
```

While the use of FTP ACIDs with non-expiring passwords is discouraged, it is the customer's decision to accept the risks as discussed in *Section 3.1.2.7.1, Risks*. If the customer is willing to officially acknowledge the risks and implement mitigating controls as discussed in *Section 3.1.2.7.2, Mitigating Controls*, the use of FTP ACIDs with non-expiring passwords on TOP SECRET systems will be permitted. The ACID can be assigned the following attributes to exempt the password from expiration and to force logging of all activity:

```
AUDIT
PASSWORD(password,0)
```

3.4.2.8 MCS Console Userids

Define a TOP SECRET Group ACID to provide access to the resources needed by all MCS consoles. If *autolog* is allowed, define a second TOP SECRET Group ACID and permit that group to have *read* access to the CONTROL, DISPLAY, MONITOR, STOPMN, STOPTR, and TRACK commands.

```
TSS CREATE(consnoautolog) TYPE(GROUP)
    NAME('MCS consoles with no autolog')
    DEPT('SYS1')

TSS CREATE(consautolog) TYPE(GROUP)
    NAME('MCS consoles with autolog')
    DEPT('SYS1')
```

Create the TOP SECRET User ACID for an MCS console with only the fields required to allow the console to *autolog*. Do not define TSO or other segments not required for the operation of the console. Do not permit the ACID to access any resources except MVS.MCSOPER.*consolename* in the OPERCMDS class. Use the appropriate TOP SECRET Group ACID defined above as the default group.

```
TSS CREATE(consname) NAME('MCS console name')
    FACILITY(CONSOLE) PASSWORD(password)
    DFLTGRP(congroup) GROUP(congroup)
```

3.4.2.9 OS/390 System Operator Userids

In order to control which OS/390 system operator can issue which commands, each operator is to have a personal user ACID, and the console definition requires operators to log on prior to entering OS/390 commands. If an operator already has a User ACID, that User ACID may be used for the purpose. Otherwise define a new User ACID as specified in *Section 3.4.2.1, Interactive Users*.

Normally several operators at a site have similar duties, responsibilities, and roles, and require the ability to enter the same OS/390 system commands. Where such a group of operators exists, define a TOP SECRET Group ACID and connect the operators to that group, instead of issuing identical permits of individual operators to MCS consoles and OS/390 commands.

3.4.3 Password Controls

3.4.3.1 Password Guidelines

The site is to utilize the capabilities of the TSS control options, and (optionally) the password validation exit entry point (refer to *Section 3.4.3.2, Password Exit Processing*), to enforce the password requirements specified in *Section 3.1.3.1, Password Guidelines.* Several of the password requirements can be enforced through the use of standard TSS mechanisms. The requirements that are not already enforceable by TOP SECRET are as follows:

Table A-37. PASSWORD REQUIREMENTS NOT ENFORCED BY TOP SECRET (3.4.3.1)

PASSWORD REQUIREMENTS NOT ENFORCED BY TOP SECRET
No words found in standard dictionaries will be used.
At least one alphabetic, numeric, and special character will be used.[17]
Each character of the password will be unique.[18]
Passwords will contain no consecutive characters (e.g., 12, AB).
Will not contain the user's name, userid (ACID), or telephone number.[19]

3.4.3.2 Password Exit Processing

As indicated previously (refer to *Section 3.1.3.2, Password Exit Processing*), the site may use the TOP SECRET installation exit to extend the capabilities of TOP SECRET to enforce any or all of the password requirements not already enforced by the ACP. If implemented, use the following exit entry point to implement these controls:

TSSINSTX Entry Point:	PASSWORD

3.4.4 Special Privilege Access

The special privileges discussed in this section are all of an extremely sensitive nature, and are to be rigidly controlled. The number of users granted these privileges are to be kept to an absolute minimum.

[17] TOP SECRET supports this requirement with a limitation. The national character (i.e., @, #, and $) must be specified between the first and last position of the password.

[18] TOP SECRET can prevent passwords from containing repeating characters in succession. Enforcing the DISA requirement that each character of the password be unique requires the use of an exit.

[19] TOP SECRET can prevent a new password that contains a user's ACID (userid), or one whose first four characters are equal to part of the user's name. Restricting the telephone number requires the use of an exit.

3.4.4.1 Access Control Product Modification Privileges

Limit the number of administrative (control) ACIDs to the minimum number necessary. The system MSCA will be a limited-use ACID, which is not available to any individual for day-to-day processing. Limit its use only to performing security administration functions. An SCA will assume the use of, and the responsibility for, the MSCA by changing the MSCA password. The password change command will include a comment indicating the reason. The SCA will remain responsible for the MSCA until the next change/assumption of responsibility.

- *(TSS0870: CAT II) The IAO will ensure that the system MSCA ACID is a limited-use ACID, which is not available to any individual for day-to-day processing it is only used to perform security administration functions.*

- *(TSS0880: CAT II) The IAO will ensure that the system MSCA ACID password changes are documented in the change log and filed with the IAO.*

The IAO defines SCAs and uses them for the day-to-day administrative functions of TSS. Further delegation of responsibilities by the IAO may be accomplished using the other security control authorizations (LSCA, ZCA, VCA, and DCA).

The TSS CONSOLE privilege allows a user to change TSS control options. It will be limited only to authorized security administrators.

- *(TSS0890: CAT II) The IAO will ensure that the TSS Console privilege is limited to only authorized security administrators*

Assign the NOATS parameter to all security administration userids.

- *(TSS0900: CAT II) The IAO will ensure that all security administrators userids have the NOATS parameter specified.*

3.4.4.2 Audit Privileges

The number of administrative (control) ACIDs (SCAs, LSCAs, VCAs, ZCAs, and DCAs) granted audit privileges will be limited to the minimum number necessary and only to authorized users. ACIDs established to perform only audit functions will be restricted to those functions.

- *(TSS0910: CAT II) The IAO will ensure that the number of control ACIDS are limited to as few as possible at a site.*

- *(TSS0920: CAT II) The IAO will ensure that control ACIDS are granted a limited amount of administrative authorities as possible.*

3.4.4.3 Tape Label Bypass Privileges

The tape label bypass privilege is restricted and will only be granted to authorized data center personnel at the user level. Use the following parameter to specify BLP authority:

TSS PERMIT(*user-acid*) VOL(*xxxxxx*) ACCESS(BLP,READ)

- *(TSS0930: CAT II) The IAO will ensure that the TAPE BYPASS LABEL PROCESSING (BLP) privilege is limited to only a few and is documented with the IAO.*

Refer to *Section 3.1.4, Special Privilege Access,* for further information.

3.4.4.4 Other Sensitive Privileges

The TSOAUTH resource class will authorize the resource name of MOUNT. Do not grant the Device Mount privilege to on-line TSO users. It may be granted to STC ACIDs who execute TSO in batch on an as-needed basis.

- *(ZTSO0030: CAT II) The IAO will ensure that the MOUNT resource is assigned only on an as needed basis for userids associated with STCs and LOGONIDS that need to execute TSO in batch.*

Strictly control access to the TSOAUTH resource class CONSOLE resource.

- *(ZTSO0030: CAT II) The IAO will strictly control and limit access to TSOAUTH resources. Authorization is restricted to authorized personnel; and justification for access is documented.*

Limit the assignment of the MISC9 (ALL) or MISC9 (CONSOLE) authorities to IAOs authorized to assign the CONSOLE attribute.

- *(TSS0950: CAT II) The IAO will ensure that only a limited number of ACIDS authorized to assign the CONSOLE attribute are allowed MISC9 (ALL) or MISC9 (CONSOLE) authority.*

The AUDIT attribute should not be turned ON for all users who have the CONSOLE attribute. Only perform auditing activities as necessary when a user is suspected of some wrongdoing.

- *(TSS0960: CAT II) The IAO will ensure that only a limited number of ACIDS authorized to assign the CONSOLE attribute are allowed to specify the AUDIT attribute and is only used when a user is suspected of wrong doing.*

Use the TRACE attribute only for trouble-shooting purposes.

- *(TSS0970: CAT II) The IAO will ensure that the trace attribute is only used for trouble shooting purposes.*

The ability to execute privileged programs will be strictly controlled and only permitted to a minimum number of users.

TSS provides a number of NO*xxx*CHKs (bypass attributes) that permit capabilities by bypassing authorization checking. Use of these NO*xxx*CHKs will be tightly controlled and their access only granted to those required user ACIDs. Avoid NO*xxx*CHKs unless a special requirement necessitates their use. The IAO will document all uses of NO*xxx*CHKs. Documentation will be maintained explaining and justifying any bypass attributes that are granted.

- *(TSS0980: CAT II) The IAO will ensure that the use of NOxxxCHKs is avoided unless a special requirement necessitates their use and the IAO documents all uses of NOxxxCHKs.*

Blanket access to all facilities, FACILITY(ALL), will never be granted.

- *(TSS0990: CAT II) The IAO will ensure that blanket access to all facilities; FACILITY(ALL), is never granted.*

The NOSUSPEND privilege may be granted to STC ACIDs as discussed in *Section 3.4.2.3, Started Task Control (STC) Users*. It will not be granted to any other type of ACID.

The MODE resource is used to specify the operating MODE of a user or profile ACID. The use of this resource will override the MODEs specified in the Global and the FACILITY Control Options and provide the user with unrestricted access. All MODE resources will be owned by the MSCA. Access to the MODE resources will be controlled and access removed in a timely manner when the users access requirement have been resolved. At the IAOs discretion a profile may have access and ACIDs added and removed when the users access requirement have been resolved. This profile can have controls in place that can restrict access to sensitive resources.

TSS PERMIT(*profile*) DSN(SYS1.LINKLIB) ACCESS(FETCH) ACTION(FAIL)

- *(TSS0995: CAT II) The IAO will ensure that the MODE resources is owned by the MSCA. Access is restricted and a letter justifying access is filed.*

Refer to *Section 3.1.4, Special Privilege Access,* for further information.

3.4.5 Resource Controls

TSS provides masking characters that can be used to identify certain values. The MSCA will own the following masking characters:

*	(asterisk or star)
+	(plus sign)
%	(percent sign)

- *(TSS1000: CAT II) The IAO will ensure that the MSCA own all masking characters.*

3.4.5.1 Data Set Controls

Data set controls are provided via the DATASET resource. All data sets will be protected. Protection by default will be enabled. Permit data set access only to authorized users.

The use of global access to data sets should be restricted to the minimum number of libraries. Certain data sets, such as general purpose load libraries, may use global permissions, but these should be restricted to the appropriate level of access (e.g., FETCH).

Data sets that are in the Linklist should be restricted only to systems, security, and audit personnel, since MVS grants FETCH-level access implicitly by the program's presence in the Linklist. However, allowing FETCH-level access to all users does not create an exposure. Libraries that are APF authorized, but are not in the Linklist, should be protected so that only the required users have access to these programs. Please note that CA-EXAMINE users (systems, security, and auditors) requires *read*-level access to these libraries based on the way that CA-EXAMINE opens the data sets.

Great care and consideration needs to go into defining the access given to the various data sets. As an example, by giving *read* access to the CA-1 parameter/control file to all users, unauthorized personnel may be able to determine how CA-1 security is set up, thus jeopardizing the entire tape library. The Security staff, along with the Systems staff, should work together to define the access needed and restrict the level of access appropriately.

The first permission would give the user READ access to all datasets on the system. This permission can be overridden by permissions that identify dataset more explicitly. The second permission would prevent the user access to all SYS1 datasets, this overrides the first permission. The third permission overrides the other permissions for all access to SYS1.LINKLIB and audits these accesses.

```
TSS PERMIT(user) DSN(*) ACCESS(READ)
TSS PERMIT(user) DSN(SYS1.) ACCESS(NONE)
TSS PERMIT(user) DSN(SYS1.LINKLIB) ACCESS(ALL) ACTION(AUDIT)
```

The use of the '*', '**', or '*.' (asterisk or star) mask only for datasets will be restricted to users that have a requirement and justification to access all datasets. These accesses will be logged.

- *(TSS1005: CAT II) The IAO will ensure that all READ and above access to '*', '**', or '*.' (asterisk or star) is restricted to authorized ACIDs, a letter justifying any additional accesses is filed with the IAO, all dataset access is logged.*

3.4.5.2 Volume Controls

Volume-level protection is supplied via the VOLUME resource class. Volume-level protection is only to be used to enhance the protection provided by data set-level authorizations. Both tape and DASD volume protections are available using the VOLUME resource class.

Grant all volume-level access to authorized users through the controls placed on the resident DASD management and tape management software products and utilities.

The granting of blanket access or permissions to volumes (i.e., VOL(*ALL*) ACC(access)) is disallowed. Volumes are to be defined by valid prefixes or discrete volume names for each OS/390 domain, not by the global VOL(*ALL*) entry. This is true for tape volumes (if using the CA-1/TSS interface) as well as for DASD volumes.

This is to ensure that each OS/390 domain is only able to access its own DASD and tapes, regardless of what devices are included in the I/O GEN and *VARYed* on-line.

- *(TSS1030: CAT II) The IAO will ensure that access authorization greater than CREATE is not permitted unless authorized by the IAO.*

3.4.5.3 Sensitive Utility Controls

Access to sensitive utilities will be strictly controlled via PROGRAM protection authorizations. TOP SECRET does not allow masking of program names for program protection control. Control access to the data sets in which the utilities reside through the use of data set access permission at the lowest required access level.

Access to protected programs considered sensitive in nature will be audited. These programs include, at a minimum, those specified in *Section 3.1.5.3, Sensitive Utility Controls.*

- *(TSS1040: CAT II) The IAO will ensure that access to sensitive utilities are strictly controlled via PROGRAM protection authorizations. Access to protected programs considered sensitive in nature are audited.*

3.4.5.4 Dynamic List Controls

Dynamic list controls are provided via resources in the FACILITY resource class. The actual owning ACID specified for *deptacid* are to be named in accordance with installation recommendations. When protecting the facilities for dynamic lists via the FACILITY class, use the following controls:

(1) Prevent access to these resources by default, and log all access. Create generic and specific profiles as follows:

TSS ADDTO(*deptacid*) IBMFAC(CSVAPF.)
TSS ADDTO(*deptacid*) IBMFAC(CSVDYNEX.)
TSS ADDTO(*deptacid*) IBMFAC(CSVDYNL.)

- *(TSS0244: CAT II) The IAO will ensure that CSV resources in the IBMFAC resource class is properly owned.*

(2) The required access to specific resources is to be discretely granted to specific systems users. Restrict this access to the absolutely minimum number of personnel, and log all access. The following is a sample command to grant profile ACID *sysprog* permission to add SYS1.NEWLIB to the APF list:

```
TSS PERMIT(sysprog) IBMFAC(CSVAPF.SYS1.NEWLIB)
    ACCESS(UPDATE) ACTION(AUDIT)
```

3.4.5.5 MCS Console Controls

MCS console controls are provided via resources in the SYSCONS, OPERCMDS, and TSOAUTH resource classes. Name the actual owning ACID specified for *deptacid* in accordance with installation recommendations. When protecting the facilities for MCS consoles via these classes, use the following controls:

(1) Prevent access to these resources by default, and log all access. Create generic and specific profiles for each console *consname* and each authorized group and profile ACID *oprprofileacid* as follows:

```
TSS ADDTO(deptacid) SYSCONS(consname)
TSS PERMIT(oprprofileacid) SYSCONS(consname) ACCESS(READ)
    ACTION(AUDIT)
TSS ADDTO(deptacid) OPERCMDS(MVS.)
TSS ADDTO(deptacid) TSOAUTH(CONSOLE)
```

- *(TSS0246: CAT II) The IAO will ensure that all consoles of the CONSOLE resource class (SYSCONS) are properly owned.*

(2) The user profile for each real MCS console is to be granted *read* access to the corresponding console resource:

```
TSS PERMIT(consname) SYSCONS(consname) ACCESS(READ)
    ACTION(AUDIT)
```

(3) The group and user profiles for operators and systems programmers allowed to use each real MCS console is to be granted *read* access to the corresponding console resource:

```
TSS PERMIT(oprprofileacid)  SYSCONS(consname) ACCESS(READ)
    ACTION(AUDIT)
```

(4) At the discretion of the IAO, users may be allowed to use the TSO **CONSOLE** command, subject to the restrictions in *Section 3.1.5.5, MCS Console Controls, Section 3.1.5.6, OS/390 System Command Controls*, and *Section 3.4.5.6, OS/390 System Command Controls*.

```
TSS ADDTO (userid) MCSAUTH(INFO)
TSS PERMIT(userid) OPERCMDS(MVS.MCSOPER.userid)
    ACCESS(READ) ACTION(AUDIT)
TSS PERMIT(oprprofileacid) TSOAUTH(CONSOLE)
    ACCESS(READ) ACTION(AUDIT)
```

3.4.5.6 OS/390 System Command Controls

OS/390 system command controls are provided via resources in the OPERCMDS resource class. Name the actual owning ACID specified for *deptacid* in accordance with installation recommendations. When protecting the facilities for OS/390 system commands via the OPERCMDS class, use the following controls:

(1) Prevent access to the OS/390 resources by default, and log all access. Create generic and specific permissions with logging as required using the resources defined in *Table A-29, Controls on OS/390 System Commands*. For example:

```
TSS ADDTO(deptacid) OPERCMDS(MVS.)
TSS PERMIT(usracid) OPERCMDS(MVS.ACTIVATE) ACTION(AUDIT)
TSS PERMIT(usracid) OPERCMDS(MVS.CANCEL.JOB.) ACTION(AUDIT)
TSS PERMIT(usracid) OPERCMDS(MVS.CONTROL.) ACCESS(UPDATE)
    ACTION(AUDIT)
TSS PERMIT(usracid) OPERCMDS(MVS.DISPLAY.) ACCESS(READ)
TSS PERMIT(usracid) OPERCMDS(MVS.MONITOR) ACCESS(READ)
TSS PERMIT(usracid) OPERCMDS(MVS.STOPMN) ACCESS(READ)
```

- *(TSS0246: CAT II) The IAO will ensure that MVS. of the OPERCMDS resource class is properly owned.*

(2) Only grant access to OS/390 system commands to the extent documented in the installation SOP. Define additional profiles similarly to those in *Paragraph (1)* above if the existing resource names are too specific or too generic for the controls in the SOP. The TSS PERMIT statements are to include the ACCESS and ACTION values specified in the SOP, or ACTION(AUDIT) if not specified.

The following is an example of granting a profile ACID *usracid* permission to issue commands against jobs with names beginning *pfx*, after obtaining permission from the IAO:

```
TSS PERMIT(usracid) OPERCMDS(MVS.CANCEL.JOB.pfx*)
    ACCESS(UPDATE) ACTION(AUDIT)

TSS PERMIT(usracid) OPERCMDS(MVS.MODIFY.JOB.pfx*)
    ACCESS(UPDATE) ACTION(AUDIT)
```

```
TSS PERMIT(usracid) OPERCMDS(MVS.STOP.JOB.pfx*)
ACCESS(UPDATE) ACTION(AUDIT)
```

The following is an example of granting users with a profile ACID of *oprprofileacid* permission to issue ROUTE commands to *sysid* from *consid*, after obtaining permission from the IAO:

```
TSS PERMIT(oprprofileacid) OPERCMDS(MVS.ROUTE.CMD.sysid)
ACCESS(READ) ACTION(AUDIT) WHEN(CONSOLE(consid))
```

3.4.5.7 TOP SECRET Encryption Key

TOP SECRET uses an encryption key as the means to encrypt and decrypt password information retained in the product's security file. Because this key is used by TOP SECRET to decrypt users' passwords, it is imperative that the key be protected. Also, to change the encryption key, the old encryption key is required as part of the TOP SECRET utility control statements. Use the following recommendations to ensure the proper protection of the security file and encryption key:

(1) The encryption key is to be recorded and locked in an acceptable container (e.g., a safe) in the event that the encryption key is required.

(2) The security control file is to be properly protected. Ensure that only the required personnel have access to the security file and that access to the file is logged.

(3) The encryption key is stored unencrypted in a module that is part of the TOP SECRET Linklist load library. Access levels of *read* or higher to this data set is to be restricted only to the Security and Systems personnel responsible for maintenance of TOP SECRET, and this access is to be logged.

(4) Any backup copies of these libraries with the same levels of control as the original data sets are to be protected. Ensure that full volume backups containing these files are restricted to those specified personnel who require access.

This page is intentionally left blank.

4. NETWORK COMMUNICATION PRODUCTS

4.1 VTAM

Vendor: IBM Corporation

Virtual Telecommunications Access Method (VTAM) is the telecommunications access method used by MVS. It controls communications as follows:

- Between devices on the network controlled by VTAM and host applications
- Between devices on the network and other connected VTAM networks
- Between peer applications using Advanced Program-to-Program Communication (APPC)

Communications between network resources (e.g., terminals or applications) are determined by the rules of the Systems Network Architecture (SNA) protocol. When two network resources satisfy all network security checks, they can be connected to each other. VTAM establishes this connection by creating an SNA session between them. The SNA session lasts as long as both partners want to continue exchanging information.

Once an SNA session is established between a terminal and an application, or between two applications, VTAM exerts no security control over data flow. The content of the data is transparent to VTAM. Both authorized and unauthorized data flows look the same. Security control over the content of the data transmitted is up to the security mechanisms invoked within the individual VTAM applications (e.g., TSO, IMS, CICS, NetView, etc.).

VTAM uses the following facilities to ensure that only authorized devices or applications have access to the network and to VTAM applications within the network:

- VTAM definitions (e.g., LOGAPPL, USSTAB) to limit the connectivity potential of network resources

- A VTAM exit, named the Session Management Exit (SME), to tailor various levels of connectivity within the network or between networks

- VTAM itself does no user-level I&A validation. Because of this, VTAM requires the services of a Session Manager (e.g., CL/GATEWAY, Netmaster, TPX), in addition to the above VTAM-based facilities, to provide a full network security implementation. Refer to *Section 6, Session Managers*, for further information.

4.1.1 VTAM Definitions

The VTAM definitions used to define and control the connectivity of terminal resources are as follows:

LOGAPPL Definitions

These definitions are included in the NCP source or in VTAM major node definitions. They identify an application (e.g., Session Manager) to which a terminal is automatically connected whenever it comes on line, or when it ceases to be connected to any other application. LOGAPPL definitions usually are used for unsecured terminals.

USSTAB Definitions

When using USSTAB definitions, a terminal is connected to VTAM when switched on. When a logon command is received from the terminal, VTAM establishes a session between the terminal and the application (e.g., Session Manager, TSO, or CICS) defined in the USSTAB definition. USSTAB definitions will only be used for secure terminals, and may be used in place of LOGAPPL.

4.1.2 VTAM Session Management Exit

The VTAM Session Management Exit (SME), ISTEXCAA, is an exit program coded by the network systems programmers, and tailored to the needs of the individual site. It is used to control network access based on the characteristics of network resources (terminals and applications), not on user identification. VTAM enters this exit multiple times during each SNA session setup, and the exit controls the acceptance or the rejection of the session.

At each session setup, VTAM enters the exit for the following:

- Initial authorization
- Secondary LU authorization
- Accounting
- Cross-network gateway selection and **ADJSSCP** selection

The SME can perform the following tasks:

(1) Check each logon attempt based on LUname, SSCPname, and the network name of both the secondary and primary LUs. The SME receives information on both LUs, on the CDRM that passed the session request to this VTAM, and on the type of session initiation (SLU-initiated, PLU-initiated, or third-party initiated) being attempted. The SME can accept or reject the session request.

(2) Track the logon and logoff of all sessions passing through VTAM.

(3) Generate SMF records for each logon and logoff to any VTAM application.

The SME consists of the following three components:

- The executable code (ISTEXCAA) and two non-executable modules
- The SME data table (SMETAB)
- The SME message module (SMEMSGS).

The SMETAB is the basis for tailoring the SME. It consists of macros that define the functions to be used in the network, and the sessions to be authorized or rejected. For any given network, it is necessary to define an SMETAB unique to that network.

SME should be used to control the connection of printers, LU 6.2 applications, programmable terminals doing INIT SELF, and terminals from external networks because these devices and applications cannot be controlled via USSTAB or LOGAPPL definitions.

4.1.3 Security Recommendations for VTAM Networks

Use the following recommendations to secure access to VTAM networks:

(1) Use USSTAB or LOGAPPL definitions to control logon from secure terminals. These terminals can log on directly to any VTAM application (e.g., TSO, CICS, etc.) of their choice and bypass Session Manager services. Secure terminals are usually locally attached to the host or connected to the host via a private LAN without access to an external network. Only authorized personnel may enter the area where secure terminals are located.

- *(ZVTM0011: CAT II) The Systems Programmer or IAO will ensure that USSTAB definitions are only used for secure terminals (e.g., terminals that are locally attached to the host or connected to the host via secure leased lines).*

(2) Use LOGAPPL definitions for all unsecured terminals. These terminals will first establish a session with the Session Manager (e.g., CL/GATEWAY, Netmaster) before establishing connectivity with any other VTAM application (e.g., TSO, CICS) in the host. Dial-up terminals or terminals attached to the Internet (e.g., TN3270 terminals, KNET 3270 emulation terminals) are examples of unsecured terminals. The user should be identified before any connection is allowed. Use of any vendor's default installation/configuration option that bypasses Session Manager services is prohibited.

- *(ZVTM0012: CAT II) The Systems Programmer and IAO will ensure that Unsecured terminals controlled by LOGAPPL definitions are forced to go through the Session Manager before establishing connectivity with VTAM applications.*

(3) The Session Manager will perform security verification (such as I&A), and will only show the applications the user is authorized to access. The Session Manager will have an SAF interface with an ACP such as ACF2, RACF, or TOP SECRET. (Refer to *Section 6, Session Managers*, for further information.)

- *(ZVTM0013: CAT II) The Systems Programmer and IAO will ensure that the Session Manager is configured to perform I&A security verification, and only show the applications the user is authorized to access.*

- *(ZVTM0014: CAT II) The Systems Programmer and IAO will ensure that the Session Manager has an active SAF interface with the ACP.*

(4) The Session Manager or VTAM (via USSTAB MSG10) will display a logon banner to the user.

- *(ZVTM0017: CAT II) The Systems Programmer and IAO will ensure that the Session Manager or VTAM displays a logon banner in accordance with DOD requirements.*

(5) Control access to all VTAM system data sets, all VTAM load modules and exit routines, and all VTAM start options and definition statements by the services of an ACP. Restrict this update to allocate access only to the network systems programming staff, and read access to STC. General users are not allowed to access these data sets.

(6) The VTAM SME may be implemented to secure access to network resources because the use of USSTAB and LOGAPPL is not effective or is not possible in some cases.

(7) Code the SMETAB as follows:

(a) Code operand CLSDST=Y in the AUTHTAB macro to allow the Session Manager to initiate logons to VTAM applications selected by terminal users from the selection menu. CLSDST-PASS processing is a way to tell VTAM that the user has been verified by the Session Manager and that application access is authorized.

(b) Code ACCEPT macros for authorized sessions between LUs representing network terminals and the Session Manager.

(c) Code ACCEPT macros for any pairs of LUs (all types of LUs including LU 6.2) authorized to establish sessions with each other. LUs may be in the same network (same net ID) or in interconnected networks (different net ID).

(d) Code REJECT macros for any pairs of LUs (all types of LUs including LU 6.2) that are not authorized to establish sessions with each other. LUs may be in the same network (same net ID) or in interconnected networks (different net ID).

(e) Code the TWOWAY operand in ACCEPT and REJECT macros only if bi-directional session authorization or rejection is necessary.

(f) Use generic entries wherever possible to reduce the number of entries in the SMETAB, and to improve SME performance during session pair authorization.

(g) Code a REPORT macro to generate SMF records to provide an audit trail.

- *(ZVTM0016: CAT II) The Systems Programmer will ensure that if implemented the VTAM SMETAB is defined with all the above features.*

(8) Establish a clear naming standard for network resources to take the best advantage of the services of the SME. Naming recommendations should clearly distinguish between host applications and remote devices, between physically secured and unsecured terminals, and between sensitive and general access applications.

(9) Secure all physical components of the network. Place them in secured locations where they cannot be interfered with, stolen, or damaged. Implement adequate access control to these locations.

(10) Wherever possible, use encryption to protect classified or confidential data (e.g., passwords) transmitted between network end points, and to prevent unauthorized personnel from reading or modifying the data being transferred. Selected encryption implementations should comply with established NSA recommendations.

4.1.4 Special Protection Mechanisms for LU 6.2 (APPC) Applications

Apply the following recommendations when securing LU 6.2 applications:

(1) Use LU 6.2 session-level LU-LU verification to verify the identity of each partner LU during the activation of sessions between LU 6.2 applications. Under this verification mechanism, one LU-LU password is assigned to each LU pair. This unique password is used only as a cryptography key to encrypt/decrypt random data exchanged between the LU partners at session establishment. If all LU-LU verifications are successful, the session can be established between the LU pair.

(2) Utilize LU 6.2 userid verification. This is because user verification, using the Session Manager, is generally not available for LU 6.2 applications. Under this verification mechanism, VTAM allows an LU to send the userid and password in the request to establish a conversation so that the partner LU can verify them.

(3) Use the SME for LU 6.2 session control between two LU 6.2 applications.

4.2 Front End Processors (Hardware and Software)

4.2.1 Overview

Besides VTAM, the Front End Processors (FEPs) are another primary building block of the SNA architecture. The FEPs are either channel attached to OS/390 host mainframes or link attached to each other (connection between local and remote FEPs) in the SNA network. Their software works with VTAM to control and manage the SNA data flow between host VTAM applications and remote network devices. The FEPs are designed to effectively off-load much of the device handling overhead from the host, thereby allowing the host more cycle time for actual applications processing.

4.2.2 FEPs Used in the DOD Networks

The DOD networks use FEPs designed and marketed by four vendors — COMTEN, IBM, CNT, and AMDAHL. The FEPs manufactured by COMTEN are heavily used and constitute by far the majority. Fundamentally, little difference exists among the four types of FEPs. All have similar general characteristics and operate under the same principles and guidelines. The following sections present the general characteristics and discuss security issues relative to all four types of FEPs.

Computer Network Technologies (CNT) CIG i6600 server is designed as an enterprise server capable of supporting up to 8,000 clients. A companion product, the CIG 2500 server running the Brixton PU5 SNA Server software, offers remote FEP functionality. SNA 3270 users connected to the PU5 SNA Server can connect to the 2500 over their local LAN. The 2500, in turn, can connect via the NIPRNet to the PU 2.1 SNA Server in the i6600 providing SNA access to the mainframe host. The 2500 running the Brixton PU5 Server software can also act as a reverse gateway to give 3270 terminal users access to applications running on open systems.

CNT's Universal Web Integrator (UWI) software should provide basic 3270 data-stream to HTML conversion capability, and also should provide a migration path to data integration and data center management functions. This software has been developed on the i6600. This is the foundation for secure web access.

4.2.3 General Characteristics of FEPs

4.2.3.1 Common Hardware Description

The FEP is a computer with the following components:

- A central processing unit
- Addressable memory storage
- Input/output control interfaces
- Channel adapters
- High speed buffer areas for storage of messages coming into the FEP from communication lines and for messages going from the FEP out to the lines

The central processing unit executes the instruction set of the FEP. It operates under the control of a network control program, and passes and receives information to and from any of the components attached to it.

The FEP also has a service subsystem. The service subsystem is a processor with its own memory storage, hard disk, diskette drive to use with removable diskettes, and control panel. The service subsystem operates independently of the main system. In the IBM and the

AMDAHL FEPs, the service subsystem is commonly known as the Maintenance and Operator Subsystem or MOSS.

The service subsystem is used mainly for operation, maintenance, and problem determination of the FEP. The service subsystem supports highly critical functions such as the following:

- FEP initialization
- Initial program loading (IPL) services
- Configuration control
- Recording of FEP component errors
- Notification of error
- Display of machine/component status
- Problem determination
- Transfer of dump to host
- Local/remote console support
- Console password management

The service subsystem continuously monitors the status of the FEP. Errors and abnormal conditions are analyzed and reported via alarms to the FEP consoles and alerts to the NetView console. Network operators can access the service subsystem functions through a local console or a remote console.

4.2.3.2 Vendor Maintenance Support

Obtaining vendor maintenance assistance (e.g., analysis of a hardware failure and identification of a failing component) is different among the FEP systems. In the COMTEN environment, the remote console is used primarily for this purpose. It is attached to the service subsystem over the public telephone network via a modem.

With the IBM FEP, obtaining vendor maintenance assistance is implemented through the Remote Support Facility (RSF).

For the AMDAHL FEP, vendor maintenance assistance is obtained via the AMDAC (Amdahl Diagnostic Assistance Center) console.

4.2.3.3 Software Description

4.2.3.3.1 COMTEN FEP

A family of system control programs called Communications Operating System Version 2 (COS2), and many individual networking and utility software products, run in the COMTEN FEP. The Network Control Program (NCP) is the primary software product that runs under the control of COS2 in the FEP. The NCP performs network control and data routing functions, such as the following, for the devices in the SNA communications network:

- Polling and addressing
- Dialing and answering

- Receiving messages into buffers
- Inserting/deleting transmission control characters as required
- Translating message data from processing code to transmission code and vice versa
- Controlling message traffic on the communication lines
- Communicating with VTAM in the host
- Recording transmission errors and diagnostics
- Performing error recovery and maintenance

Besides the NCP, the use of a set of support software programs (SSPs) is also necessary. Before it can be activated and used for network control with VTAM, a load module of the NCP should first be defined by source definition macros, generated through an assembly/link-editing process, and then loaded into the FEP.

The source definition macros are used to define network configuration resources to the NCP. The macros are also used to define many other NCP characteristics, such as NCP name, load library name, FEP model, etc.

SSPs are developed to run in the host. They consist of utilities that primarily assemble, generate, load, and dump the NCP. SSPs also include the trace analysis program for problem determination.

In the COMTEN environment, SSPs mainly comprise the Enhanced Generation product (EGEN), the COMTEN Language Support System (CLSS), and the Network Support Services (NSS).

EGEN is used for system generation. It accepts macro calls and keywords as input, processes them, and produces a load module for the FEP. CLSS operates in the OS/390 host and provides the means for creating and maintaining source and object code files for COMTEN FEPs. This is accomplished through translation of source code into object code, library management, module linking, and editing capabilities. CLSS is required for EGEN.

NSS is primarily designed to handle host utility functions such as loading and dumping for both local and remote FEPs. NSS also works with COS2 to provide network support. Prior to generation processing, NSS should be installed in the host computer.

4.2.3.3.2 IBM and AMDAHL FEPs

In the IBM and AMDAHL FEPs, the software composition is somewhat different. First, there is no COS2. The NCP assumes the role of the operating system and performs all network control functions as the COMTEN NCP does. IBM SSPs are called the System Support Programs, and Amdahl SSPs are named Host Utilities. As do SSPs in the COMTEN environment, these Host Utilities also primarily assemble, generate, load, and dump the NCP.

4.2.3.4 FEP Operations

4.2.3.4.1 Service Subsystem Operations

Most service subsystem operations are available from the local console, the remote console, the IBM RSF console (IBM only), or the AMDAC console (AMDAHL only). Either console can be used to perform the following tasks:

- Display FEP component status
- Enable or disable channel adapters
- Perform selected operations on the hard disk
- Perform debug and maintenance functions
- Support operational and diagnostic capabilities
- Execute many other service subsystem functions

Access to service subsystem functions via any type of console (local, remote, RSF, or AMDAC) is controlled by passwords defined by the user. It is important to be aware that in the FEP environment, no interface and/or facility exists to invoke the services of an ACP such as ACF2, RACF, or TOP SECRET. All ACPs are designed to only run in the host environment.

4.2.3.4.2 Network Operations

Network operations are controlled through many VTAM commands that affect both VTAM and NCP operations. Access to these commands is possible via any NetView console. NCP Load, NCP Dump, NCP Activate, and NCP Deactivate are the most commonly used commands for the FEPs.

The NCP Load command is used to load the NCP into the FEP in the following ways:

- Load from a Host Data Set Across the Channel

- Load from the FEP Hard Disk

  This process loads the NCP from the FEP hard disk, and is initiated either by a VTAM operator command from the host or from the service subsystem console.

- Load from the FEP Diskette

  This process loads the NCP from diskettes and is initiated by the service subsystem console. In this case, the NCP load module is loaded into the hard disk first, and then uploaded to FEP memory storage. This process is most commonly used to load remote FEPs with small load modules (less than one megabyte).

- Load from a Local FEP (Channel-Attached to the Host) to a Remote FEP Across a SDLC Communication Link Connecting the Two FEPs

This process is commonly used to load big NCP load modules into memory storage of a remote FEP, and is initiated by a VTAM operator command from the host.

The NCP Dump command stores the NCP dump on the FEP hard disk and then transfers it to a host data set. The NCP Activate command makes the NCP ready for the network communication functions. The NCP Deactivate command places the NCP in an inoperable state. All these commands are initiated by a NetView operator from the host.

4.2.4 General Security Recommendations for FEPs

In the SNA network, FEPs are always intermediate nodes. They are located between OS/390 hosts and the remote network devices. Intermediate nodes do not process data. They only receive, buffer, and then pass data through the network. Because of this, FEPs do have different security requirements than other network nodes (e.g., entry nodes or destination nodes).

Use the following recommendations and techniques to secure hardware and software components of the FEPs:

(1) Physical security is the first level of security control for the FEPs. Install all hardware components of the FEPs in secure locations where they cannot be stolen, damaged, or disturbed.

Physical security is critical for the protection of the control panel, the operator console (local and/or remote), and the diskette drive of the service subsystem. Only authorized users are allowed access to and use of those facilities.

(2) Access to service subsystem functions and FEP resources from the control panel and from any console (local or remote) will be rigidly enforced and restricted only to authorized personnel.

- *(ZFEP0011: CAT II) The IAO will ensure that all hardware components of the FEPs are placed in secure locations where they cannot be stolen, damaged, or disturbed.*

(3) Control authorization to use service subsystem console (local or remote) by FEP internal security control through password validation. Restrict access to these passwords to the absolutely minimum number of necessary personnel. Use of vendor default passwords is prohibited. Assign different passwords for the local and remote consoles. Disconnect the local/remote console after three unsuccessful attempts to log on. Passwords used by vendor (COMTEN, IBM, CNT, or AMDAHL) service personnel will be changed after any maintenance is done. All passwords will be changed every 90 days. Restrict permission to change passwords only to authorized personnel.

Refer to *Section 3.1.3.1, Password Guidelines,* for further information on password guidelines.

(4) Use a key-lock switch on the modem supporting the remote console of the service subsystem to prevent unauthorized access. The key-lock switch is only open for scheduled and authorized remote access.

- *(ZFEP0012: CAT II) The IAO will ensure that procedures are in place to restrict access to the functions of the service subsystem from operator consoles (local and/or remote), and to restrict access to the diskette drive of the service subsystem.*

- *(ZFEP0016: CAT II) The IAO will ensure that a password control is in place to restrict access to the service subsystem via the operator consoles (local and/or remote) and is changed every 90 days. A key-lock switch is used to protect the modem supporting the remote console of the service subsystem.*

(5) Control access to NCP system data sets, NCP source definition data sets, NCP load modules, and NCP dump data sets stored in the host by the services of an ACP. Restrict access only to authorized personnel. General users are not allowed to access these data sets.

- *(ZFEP0015: CAT II) The IAO will ensure that data set access authorization restricts UPDATE and/or ALLOCATE access to appropriate personnel and any additional access requires a letter justifying access.*

(6) Control access to host support software programs by the services of an ACP. Restrict access only to authorized personnel. The host support software programs contain utilities that assemble, generate, load, and dump the NCP, and utilities to format and print NCP dumps. General users are not allowed to access these utilities.

(7) Implement, document, and secure all procedures relative to the NCP load and dump processes. Permission to access and use these procedures will be restricted only to authorized personnel.

(8) Any established procedures relative to the NCP load and dump processes will be reviewed to look for potential security exposures, especially when PCs are involved. Document any potential security exposures, and notify IAO and the vendor.

- *(ZFEP0013: CAT II) The IAO will ensure that a documented procedure is available instructing how to load and dump the NCP.*

(9) Only authorized NetView operators can issue Load/Dump/Activate/Deactivate NCP commands. Authorization to issue these commands will be coordinated between the IAO and Operations and Network Support personnel. Also, refer to *Section 12.3, NetView*, for further information about NetView security.

(10) Network operators should use the VTAM display command and the services of the service subsystem to verify daily the current valid version of the NCP load module (generation date and time) and FEP disk contents. The operators are to report any unusual conditions to management immediately.

(11) Implement strict change control/management for any hardware upgrade or software change, FEP memory upgrade, installation of new communication lines, new release of NCP, and new NCPGEN to support additional support of remote devices by the NCP, etc. Explain and fully document any upgrade/change to the current version/configuration.

(12) The current change control/management mechanisms to detect and eliminate potential security exposures will be thoroughly reviewed. Document any potential security exposures, and notify IAO and the vendor.

(13) Maintain a log of all hardware and software upgrades/changes for auditing purposes and problem tracking.

- *(ZFEP0014: CAT II) The IAO will ensure that a log is available to keep track of all hardware upgrades and software changes.*

(14) All network activities should be constantly monitored by NetView operators working in the network control center. NetView operators are instructed not to execute any network change requests that have not been properly scheduled and authorized through the change control/management process. Immediately report any such requests to management.

If implemented properly, all the above security mechanisms acting together should make it difficult, if not impossible, for any illegal intrusion or any unauthorized circumvention of security controls to go undetected and thus inflict damage to systems.

4.3 MQSERIES/WebSphere MQ

The information in this section pertains to Version 5 Release 1,Version 5 Release 2 of MQSeries and Version 5 Release 3 WebSphere MQ, MQSeries was renamed to WebSphere MQ in Version 5 Release 3.

4.3.1 Overview MQSeries/WebSphere MQ

MQSeries / WebSphere MQ, is an IBM software product, that provides applications the ability to communicate with each other across multiple platforms using messages and queues. In order for an application to use MQSeries/WebSphere MQ to transfer data, it must be authorized by the ACP. MQSeries / WebSphere MQ uses the SAF interface of the operating system to request resource access authorization from the ACP, (i.e., ACF2, RACF, and TOP SECRET).

If the ACP authorizes an application to transmit data using MQSeries / WebSphere MQ, the application opens a MQSeries/WebSphere MQ queue and places the data, in message format, into the queue. MQSeries/WebSphere MQ queue managers, which are responsible for controlling the queues and managing the messages on the queues, pass the messages to an internal transmit queue for transfer over the MQSeries/WebSphere MQ channel. Data mover programs, referred to as Message Channel Agents MCAs, perform the message transfer between queue managers. Once the message is transferred to the target queue manager’s queue, MQSeries / WebSphere MQ notifies the receiving application interface that the data is available to the application. The target queue manager reformats the message into the data’s original format and passes the data to the application. The target queue manager may reside in the same OS machine, or on another machine, such as a Unix or Windows NT machine.

If the ACP disallows the request for use of the MQSeries/WebSphere MQ resource, MQSeries/WebSphere MQ prevents access to the object. Since distributed queuing or clients are being used, additional security measures are required. This depends on the platform where the initial request for the use of MQSeries/WebSphere MQ resources is initiated.

MQSeries / WebSphere MQ installation and maintenance is performed using a CMP.

- *(AAMV0010: CAT III) The systems programmer responsible for supporting MQSeries/WebSphere MQ will ensure that MQSeries/WebSphere MQ is installed and maintained using a CMP.*

4.3.1.1 MQSeries / WebSphere MQ Exits

There are six types of WebSphere MQ exits that can be used for customizing channels. They range from security to message reformatting. Earlier releases of WebSphere MQ (i.e., MQSeries) transferred data across channels in clear text.

4.3.1.2 MQSeries Channel Security Exits

In order to secure MQSeries message headers, a GOTS security exit was developed to encrypt the header information and to invoke the ACP on the mainframe. In a distributed queuing environment channel, security exits were implemented to provide authentication between the message channel agents (MCAs) on a channel. The exits authenticated with a unique userid and a password for each channel. The channel security exits work in pairs ensuring that compatible exits are named for both ends of the channel.

The name of the channel security exit is defined in the SCYEXIT parameter of the channel definition. This is the name of the exit called by the channel.

For distributed queues not involving CICS, the channel security exit module resides in the data set specified by the CSQXLIB DD of the channel initiator procedure. For distributed queues involving CICS, the exit resides in a library included in the DFHRPL concatenation.

All Channel Security Exits must be submitted to DISA FSO for Program Integrity Analysis. They must be reviewed and approved by DISA FSO prior to implementation in a production environment.

4.3.1.3 WebSphere MQ Environment

WebSphere MQ uses SSL as the protocol for handling message transmission and interfaces with SAF to request access authorization of the resource. In addition, WebSphere MQ allows for the use of PKI certificates to further enhance security. SSL provides encryption and decryption of the entire message as it is transferred across the channel. As a result, the use of a GOTS security exit between two WebSphere MQ R5.3 sites is not needed. It should be noted that if a WebSphereMQ R5.3 site wants to transfer messages with a MQSeries R5.2 site, the GOTs exit or a compatible equivalent security product must be used.

If a WebSphere MQ exit is developed, it must be submitted to DISA FSO for a Program Integrity Analysis. Once approved by DISA FSO, the exit can be used in the DOD enterprise.

NOTE: Releases 5.2 and 5.1 of MQSeries require the use of a GOTS security exit or CommerceQuest’s Data Integrator/ProtectMQ.

- *(ZWMQ0011: CAT I) The IAO will ensure that the systems programmer responsible for supporting WebSphere MQ has implemented SSL for WebSphere MQ channels.*

- *(ZWMQ0012: CAT II) The IAO will ensure that WebSphere MQ is using DOD approved PKI certificates and that they are stored in a keyring in the ACP database.*

- *(ZWMQ0010: CAT I) The IAO will ensure that the systems programmer responsible for supporting WebSphere MQ has implemented the DISA approved GOTS exit or CommerceQuest’s Data Integrator/ProtectMQ for releases 5.1 and 5.2 of MQSeries or for communications to releases 5.1 and 5.2 of MQSeries.*

4.3.1.4 WebSphere MQ Clustering

A WebSphere MQ cluster is a network of queue managers that are logically grouped to support an application. The queue managers in a cluster normally communicate directly with each other through automatically defined cluster channels. Every queue manager in a cluster is able to make their queues that they host available to every other queue manager in the cluster. When a cluster is created, multiple queue managers are designated to be the full repository queue managers. These queue managers are responsible for storing information about the names of the queue managers and the network connections used by each.

Every queue manager in a cluster has a single transmission queue from which it can transmit messages to any other queue manager in the cluster. Whenever a message is to be sent across a cluster, the message is placed in the cluster transmission queue of the sending queue manager and is transmitted to all other queue managers in the cluster. Any queue manager can send the message to any other queue manager without the need for explicit channel definitions, remote-queue definitions, or transmission queues for each destination.

In order for a queue manager to become a member of a cluster, a cluster sender channel and a cluster receiver channel must be defined. The cluster sender channel points to the full repository queue manager. The cluster receiver channel provides the connection details. The name of the cluster sender channel must match the name of the cluster receiver channel on the full repository queue manager.

Three system queue objects that are used to support clustering are:

1. The queue *SYSTEM.CLUSTER.TRANSMIT.QUEUE* which holds all messages that are ready to be sent to any queue manager in the cluster.

2. The queue *SYSTEM.CLUSTER.COMMAND.QUEUE* which is used to exchange repository information.

3. The queue *SYSTEM.CLUSTER.REPOSITORY.QUEUE* which holds the repository information as a number of persistent messages.

It should be noted that a cluster of VPNs may be used to ensure encryption.

4.3.1.5 Switch Profiles

Switch profiles are special MQSeries/WebSphere MQ profiles that are used to turn on/off security checking for a type of resource. Due to the security exposure this creates, no profiles with the first two qualifiers of *ssid.NO* will be defined to the MQADMIN class, with one exception. Due to the fact that (1) all sensitive MQSeries/WebSphere MQ commands are restricted to queue managers, channel initiators, and designated systems personnel, and (2) no command resource checking is performed on DISPLAY commands, at the discretion of the IAO a *ssid.NO.CMD.RESC.CHECKS* switch profile may be defined to the MQADMIN class.

- *(ZWMQ0051: CAT II) The IAO will ensure that all Switch Profiles do not have the resource ssid.NO defined to the MQADMIN resource class with the exception of ssid.NO.CMD.RESC.CHECKS.*

4.3.1.6 Utilities

Access to the following MQSeries/WebSphere MQ programs will be restricted to the WebSphere MQ administrator and systems programming personnel by defining profiles in the **PROGRAM** class:

CSQUTIL
CSQUCVX
CSQJU003
CSQJU004
CSQ1LOGP

- *(ACF0380: CAT II, ACF0870: CAT II, RACF0770: CAT II, TSS1040: CAT II) The IAO will ensure that the above utilities are restricted to authorized personnel.*

4.3.1.7 Userid Timeouts

Userids signed on to a queue manager will be logged off after 15 minutes of inactivity. This timeout process will be implemented by including the ALTER SECURITY command in the CSQINP1 data set. The format of the command will be specified as follows:

```
ALTER SECURITY INTERVAL(5) TIMEOUT(15)
```

- *(ZWMQ0020: CAT II) The systems programmer responsible for supporting MQSeries/WebSphere MQ will ensure that the timeout is set to 15 and the interval is set to 5.*

4.3.2 General Security Considerations

In a z/OS environment, MQSeries/WebSphere MQ will require its resources and connections to those resources to be restricted. These resources include, but are not limited to MQSeries/WebSphere MQ objects, programs, commands and data sets. It is essential that none of the following are accessed or changed by any unauthorized users or processes:

Connections to MQSeries/WebSphere MQ
MQSeries/WebSphere MQ objects such as queues, processes, and namelists
MQSeries/WebSphere MQ transmission links
MQSeries/WebSphere MQ system control commands
MQSeries/WebSphere MQ messages
Context information associated with MQSeries/WebSphere MQ messages

As a result, MQSeries/WebSphere MQ will depend on multiple layers of security. They are:

- Platform security, (such as ACPs, file permissions and group level controls)
- Communication security, (such as access lists and port control)
- Application security, (such as exits)

Refer to the appropriate STIG for guidance in addressing the above layers of security.

- *(ZWMQ0049: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ resources are active and properly defined.*

- *(ZWMQ0054: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ queues are restricted using queue level security.*

- *(ZWMQ0030: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ started tasks are properly defined.*

- *(ZWMQ0059: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ commands are restricted to authorized personnel.*

- *(ZWMQ0055: CAT II) The IAO will ensure that process security is active, and that all profiles defined to the MQPROC class and that process inquiries are restricted to read access.*

- *(ZWMQ0060: CAT II) The IAO will ensure that a ssid.RESLEVEL profile is only defined for each queue manager.*

- *(ZCIC0020: CAT II) The IAO will ensure that MQSeries/WebSphere MQ-supplied CICS transactions are restricted to CICS regions and the WebSphere MQ administrator.*

- *(ZWMQ0057: CAT II) The IAO will ensure that use of alternate userids is restricted to authorized personnel.*

- *(ZWMQ0058: CAT II) The IAO will ensure that use of context resources are restricted to authorized personnel.*

- *(ZWMQ0052: CAT II) The IAO will ensure that all connections to MQSeries/WebSphere MQ resources are restricted using connection security.*

- *(ZWMQ0056: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ namelist resources are restricted to authorized users.*

- *(ZWMQ0040: CAT II) The IAO will ensure that all update and alter access to MQSeries/WebSphere MQ product and system data sets are restricted to WebSphere MQ administrators, systems programmers, and MQSeries/WebSphere MQ started tasks.*

- *(ZWMQ0062: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ transmission links are restricted according to the Network STIG and Enclave STIG guidelines.*

- *(ZWMQ0053: CAT II) The systems programmer responsible for supporting MQSeries/WebSphere MQ will ensure that the dead-letter queue and its alias are properly defined.*

- *(ZWMQ0061: CAT II) The IAO will ensure that MQSeries/WebSphere MQ configurations from non-approved networks are approved by DISA FSO prior to implementation.*

4.3.2.1 SSL

You can also use the Secure Sockets Layer (SSL) to provide channel security. SSL support is provided with WebSphere MQ, Version 5 Release 3 for z/OS. SSL is an industry-standard protocol that provides a data security layer between application protocols and the communications layer, usually TCP/IP. SSL uses encryption techniques, digital signatures and digital certificates to provide message privacy, message integrity and mutual authentication between clients and servers.

In order to use the Secure Sockets Layer (SSL) for channel security, the following tasks will have to be performed:

- Create a key ring in the ACP to hold all the keys and certificates for the system. The *id* should be the *channel initiator address space*.

- Create a digital certificate for each queue manager. The label of the certificate must be of the form ibmWebSphereMQ*qmgr-name*.

- Connect the certificate and any relevant signer certificates to the key ring in the ACP.

- Point the queue manager to the key repository using the WebSphere MQ ALTER QMGR command.

- Create an AUTHINFO object of AUTHTYPE CRLLDAP, using the WebSphere MQ DEFINE AUTHINFO command to indicate that the Certificate Revocation Lists (CRLs) are stored on a specific LDAP server.

- Set each queue manager to run SSL calls using the WebSphere MQ ALTER QMGR command. There must be at least two of these subtasks.

- Specify the *cipher specification* for the channel on each end of the channel using the WebSphere MQ DEFINE CHANNEL or ALTER CHANNEL command.

- *(ZWMQ0011: CAT I) The systems programmer responsible for supporting WebSphere MQ will ensure that the WebSphere MQ channels use SSL.*

- *(ZWMQ0012: CAT II) The IAO will ensure that DOD approved certificates are stored in the ACP database.*

- *(ZWMQ0015: CAT II) The IAO will ensure that Certificate Revocation Lists (CRLs) are stored on a LDAP server that is restricted to authorized users.*

NOTE: The format of each MQSeries/WebSphere MQ command can be found in the IBM MQSeries/WebSphere MQ Commands manual and the format of each ACP command can be found in the vendor's ACP documentation.

4.3.3 CA-ACF2

The following subsections provide specific guidance and examples as to how MQSeries/WebSphere MQ should be secured using CA-ACF2. All of the following subsections follow the guidelines as specified in *Section 4.3.2, General Security Considerations.*

NOTE: Refer to Section 2.49 of the CA-ACF2 Other Products Guide, MQM (Message Queue Manager) for additional information.

4.3.3.1 ACF2 Security Classes

In order to enable security for MQSeries/WebSphere MQ under ACF2, enter the following commands:

```
SET CONTROL(GSO)
INSERT SAFDEF.MQS ID(MQS) FUNCRET(8) RETCODE(4) MODE(IGNORE)
RACROUTE(REQUEST=EXTRACT,CLASS=MQADMIN) REP
```

In order to ensure that ACF2 checking is performed on all MQSeries/WebSphere MQ resources, the following GSO CLASMAP records should be inserted:

```
MQADMIN
MQCONN
MQCMDS
MQQUEUE
MQPROC
MQNLIST
```

The following is a sample of the commands required to insert the required CLASMAP records:

```
SET CONTROL(GSO)
INSERT CLASMAP.MQADMIN RESOURCE(MQADMIN) RSRCTYPE(MQA)
        ENTITYLN(62)
INSERT CLASMAP.MQQUEUE RESOURCE(MQQUEUE) RSRCTYPE(MQQ)
        ENTITYLN(53)
INSERT CLASMAP.MQNLIST RESOURCE(MQNLIST) RSRCTYPE(MQN)
        ENTITYLN(53)
INSERT CLASMAP.MQCMDS RESOURCE(MQCMDS) RSRCTYPE(MQC)
        ENTITYLN(22)
INSERT CLASMAP.MQCONN RESOURCE(MQCONN) RSRCTYPE(MQK)
        ENTITYLN(10)
INSERT CLASMAP.MQPROC RESOURCE(MQPROC) RSRCTYPE(MQP)
        ENTITYLN(53)
```

The above ACF2 resource type (**RSRCTYPE**) values are the STIG required values.

4.3.3.2 MQSeries/WebSphere MQ Resources

4.3.3.2.1 Started Tasks

Create a started task logonid entry for each queue manager started task procedure *xxxx*MSTR and distributed queuing started task procedure *xxxx*CHIN. Create a corresponding userid for each started task, specifying the following LID parameters:

STC
MUSASS
NOSMC

4.3.3.2.2 Data Sets

Use the following recommendations to ensure that protection of the queue manager and channel initiator data sets is properly in place:

(1) The installation requires that the following data sets be APF authorized. (Refer to *Section 2.1.2.1, Authorized Program Facility [APF])*, for security guidelines and recommendations.)

hlqual.SCSQAUTH
hlqual.SCSQLINK
hlqual.SCSQANLx
hlqual.SCSQSNL
hlqual.SCSQMVR1
hlqual.SCSQMVR2

(2) *Read* access to data sets referenced by the CSQINP1, CSQINP2, and CSQXLIB DDs in the queue manager's procedure will be restricted to the queue manager userid, WebSphere MQ administrator, and systems programming personnel. Log all access to these data sets.

(3) *Write* and *allocate* access to data set profiles protecting all page sets, logs, bootstrap data sets (BSDS), and data sets referenced by the CSQOUTX and CSQSNAP DDs in the queue manager's procedure will be restricted to the queue manager userid, WebSphere MQ administrator, and systems programming personnel. Log all *write* and *allocate* access to these data sets.

(5) *Allocate* access to all archive data sets in the queue manager's procedure will be restricted to the queue manager userid, WebSphere MQ administrator, and systems programming personnel. Log all *allocate* access to these data sets.

- *(ZWMQ0040: CAT II) The IAO will ensure that all update and alter access to MQSeries/WebSphere MQ product and system data sets are restricted to WebSphere MQ administrators, systems programmers, and MQSeries/WebSphere MQ started tasks.*

4.3.3.2.3 Connection Security

Connection security validates userids authorized to connect to queue managers. Connection security will be active and all profiles will be defined to the MQCONN class.

Restrict access to connection security profiles using the following table as a guideline:

Table A-38. CONNECTION SECURITY CONTROLS (4.3.2.2.3)

PROFILE NAME	AUTHORIZED USERS	ACCESS	LOGGING REQUIRED
ssid.BATCH	TSO userids Batch job userids	Log	Y
ssid.CICS	CICS region userids	Log	Y
ssid.IMS	IMS region userids	Log	Y
ssid.CHIN	Channel initiator userids	Log	Y

NOTE: ssid is the name of the queue manager.

The following is a sample of the commands required to allow a batch user (USER1) to connect to a queue manager (QM1):

```
SET RESOURCE(MQK)

COMPILE * STORE

$KEY(QM1) TYPE(MQK)
- UID(USER1) LOG
```

- *(ZWMQ0052: CAT II) The IAO will ensure that all connections to MQSeries/WebSphere MQ resources are restricted using connection security.*

4.3.3.2.4 Queue Security

Queue security validates userids authorized to access message queues. Queue security will be active and all profiles will be defined to the MQQUEUE class.

Message queue access will be restricted to those userids that require the ability to get messages from and put messages to message queues. The profile names for queue security are *ssid.queuename*, where *ssid* is the name of a MQSeries / WebSphere MQ subsystem.

The following is a sample of the commands required to allow a user (USER1) to get messages from or put messages to queues beginning with (PAY.) on subsystem (QM1):

```
SET RESOURCE(MQQ)
```

```
COMPILE * STORE

$KEY(QM1.PAY) TYPE(MQQ)
- UID(USER1) SERVICE(READ,UPDATE)
```

Access authorization to system queues (those queue resources with a first qualifier of *system*) will be restricted to the CSQUTIL utility, MQSeries / WebSphere MQ operations and control panels, channel initiators, MQSeries/WebSphere MQ software monitors, and CICS transactions.

- *(ZWMQ0054: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ queues are restricted using queue level security.*

4.3.3.2.4.1 Dead-Letter Queue Security

Whenever a message cannot be routed by a queue manager, the message is routed to the dead-letter queue assigned to that queue manager. Dead-letter queues are defined when the queue manager is created. The two levels of access for these queues are established using the PUT and GET attributes for the files. Unlike UNIX and NT, authority to *read* and *update* a dead-letter queue on an OS/390 platform is determined by the rules defined for the queue in the ACP. As a result, in order to further restrict messages from being read by unauthorized users, IBM recommends the use of an alias queue. An alias queue is a re-definition of the same queue. Alias queues are used to assign a different name and different attributes to a physical queue. By defining an alias queue for a dead-letter queue with its PUT option set to enabled (i.e., PUT(ENABLED) and its GET option set to disabled (i.e., GET(DISABLED), messages can be further secured. The first level allows applications, as well as some MQSeries/WebSphere MQ objects, to put messages to this queue. The second level restricts the ability to get messages from this queue and protects sensitive data. The ability to get messages from the dead-letter queue will be restricted to message channel agents (MCAs), CKTI (MQSeries/WebSphere MQ-supplied CICS task initiator), channel initiators utility, WebSphere MQ administrators, and any automated application used for dead-letter queue maintenance.

The following scenario describes how a dead-letter queue should be defined securely:

(1) Define the real dead-letter queue with attributes PUT(ENABLED) and GET(ENABLED).

(2) Give ACF2 *write* authority for the dead-letter queue to CKTI, the MQSeries/WebSphere MQ-supplied CICS task initiator, channel initiators, and any automated application used for dead-letter queue maintenance.

(3) Define an alias queue that resolves to the real dead-letter queue, but give the alias queue the attributes PUT(ENABLED) and GET(DISABLED).

(4) To put a message on the dead-letter queue, an application uses the alias queue. The application does the following:

(a) Retrieve the name of the real dead-letter queue. To do this, it opens the queue manager object using MQOPEN and then issues an MQINQ to get the dead-letter queue name.

(b) Build the name of the alias queue by appending the characters ".PUT" to this name, in this case, *ssid*.DEAD.QUEUE.PUT.

(c) Open the alias queue, *ssid*.DEAD.QUEUE.PUT.

(d) Put the message on the real dead-letter queue by issuing an MQPUT against the alias queue.

(5) Give the userid associated with the application ACF2 *write* authority to the alias, but no access to the real dead-letter queue.

NOTE: If an alias queue is not used in place of the dead-letter queue, then the ACF2 rules for the dead-letter queue will be coded to restrict unauthorized users and systems from reading the messages on the file.

- *(ZWMQ0053: CAT II) The systems programmer responsible for supporting MQSeries / WebSphere MQ will ensure that the dead-letter queue and its alias are properly defined.*

- *(ZWMQ0054: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ queues are restricted using queue level security.*

4.3.3.2.5 Process Security

Process security validates userids authorized to issue MQSeries / WebSphere MQ inquiries on process definitions. A process definition object defines an application that is started in response to a trigger event on queue manager. Process security will be active, and all profiles *ssid.processname* will be defined to the **MQPROC** class. Restrict *read* access to those userids requiring access to make process inquiries.

The following is a sample of the commands required to allow a group (GRP1) to inquire on processes beginning with the letter **V** on queue manager (QM1):

```
SET RESOURCE(MQP)

COMPILE * STORE

$KEY(QM1.V*) TYPE(MQP)
- UID(GRP1) LOG
```

- *(ZWMQ0055: CAT II) The IAO will ensure that process security is active, and that all profiles defined to the MQPROC class and that process inquiries are restricted to read access.*

4.3.3.2.6 Namelist Security

A namelist is a MQSeries/WebSphere MQ object that contains a list of queue names. Namelist security validates userids authorized to inquire on namelists. Namelist security will be active, and all profiles *ssid.namelist* will be defined to the MQNLIST. Restrict access to those userids requiring access to make namelist inquiries.

The following is a sample of the commands required to allow a group (GRP1) to inquire on namelist TST1 on queue manager (QM1):

```
SET RESOURCE(MQN)

COMPILE * STORE

$KEY(QM1.TST1) TYPE(MQN)
- UID(GRP1) LOG
```

- *(ZWMQ0056: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ namelist resources are restricted to authorized users.*

4.3.3.2.7 Alternate Userid Security

Alternate userid security allows access to be requested under another userid. Alternate userid security will be active, and all profiles *ssid.ALTERNATE.USER.alternateuserid* will be defined to the MQADMIN class. Restrict access to those userids requiring access to alternate userids.

The following is a sample of the commands required to allow payroll server (PAYSRV1) to specify alternate userids starting with the characters PS on queue manager (QM1):

```
SET RESOURCE(MQA)

COMPILE * STORE

$KEY(QM1.ALTERNATE.USER.PS*) TYPE(MQA)
- UID(PAYSRV1) LOG
```

- *(ZWMQ0057: CAT II) The IAO will ensure that use of alternate userids is restricted to authorized personnel.*

4.3.3.2.8 Context Security

Context security validates whether a userid has authority to pass or set identity and/or origin data for a message. Context security will be active and all profiles *ssid.CONTEXT* will be defined to the MQADMIN class, where *ssid* is the queue manager name.

The following is a sample of the commands required to allow a systems programming group (SYS1) to offload and reload messages for queue manager (QM1):

```
SET RESOURCE(MQA)

COMPILE * STORE

$KEY(QM1.CONTEXT) TYPE(MQA)
- UID(SYS1) LOG
```

- *(ZWMQ0058: CAT II) The IAO will ensure that use of context resources are restricted to authorized personnel.*

4.3.3.2.9 Command Security

Command security validates userids authorized to issue MQSeries/WebSphere MQ commands. Command security will be active, and all profiles will be defined to the MQCMDS class.

Restrict access to command security profiles using the following table:

Table A-39. COMMAND SECURITY CONTROLS (4.3.3.2.9)

COMMAND	PROFILE	ACCESS LEVEL	AUTHORIZED USERS	LOG
ALTER *xxxxx*	ssid.ALTER.xxxxx	LOG	MQ administrator Systems programmers Queue managers	Y
ARCHIVE LOG	*ssid*.ARCHIVE.LOG	LOG	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
CLEAR QLOCAL	*ssid*.CLEAR.QLOCAL	LOG	MQ administrator Systems programmers Queue managers	Y
DEFINE *xxxxx*	ssid.DEFINE.xxxxx	LOG	MQ administrator Systems programmers Queue managers	Y
DELETE *xxxxx*	ssid.DELETE.xxxxx	LOG	MQ administrator Systems programmers Queue managers	Y
DISPLAY *xxxxx*	*ssid*.DISPLAY.*xxxxx*	LOG	Application programmers MQ administrator Systems programmers Queue managers Operators Console automation software	N

COMMAND	PROFILE	ACCESS LEVEL	AUTHORIZED USERS	LOG
PING *xxxxx*	ssid.PING.xxxxx	LOG	Application programmers MQ administrator Systems programmers Queue managers Operators Console automation software	N
RECOVER BSDS	*ssid*.RECOVER.BSDS	LOG	MQ administrator Systems programmers Queue managers	Y
REFRESH *xxxxx*	*ssid*.REFRESH.*xxxxx*	LOG	Security staff MQ administrator Systems programmers Queue managers	Y
RESET *xxxxx*	ssid.RESET.xxxxx	LOG	MQ administrator Systems programmers Queue managers	Y
RESOLVE *xxxxx*	ssid.RESOLVE.xxxxx	LOG	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
RESUME QMGR	*ssid*.RESUME.QMGR	LOG	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
RVERIFY SECURITY	*ssid*.RVERIFY.SECURITY	LOG	Security staff MQ administrator	Y
START *xxxxx*	ssid.START.xxxxx	LOG	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
STOP *xxxxx*	*ssid*.STOP.CHINIT	LOG	MQ administrator Systems programmers Queue managers Operators Console automation software	Y

COMMAND	PROFILE	ACCESS LEVEL	AUTHORIZED USERS	LOG
SUSPEND QMGR	*ssid*.SUSPEND.QMGR	LOG	MQ administrator Systems programmers Queue managers Operators Console automation software	Y

The following is a sample of the commands required to allow a systems programming group (SYS1) to issue the command CLEAR QLOCAL in subsystem QM1:

```
SET RESOURCE(MQC)

COMPILE * STORE

$KEY(QM1.CLEAR.LOCAL) TYPE(MQC)
- UID(SYS1) LOG
```

- *(ZWMQ0059: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ commands are restricted to authorized personnel.*

4.3.3.2.10 RESLEVEL Security

RESLEVEL security profiles control the number of userids checked for API resource security. RESLEVEL security will not be implemented due to the following exposures and limitations:

(1) RESLEVEL is a powerful option that can cause the bypassing of all security checks.

(2) Security audit records are not created when the RESLEVEL profile is utilized.

(3) If the WARNING option is specified on a RESLEVEL profile, no warning messages are produced.

To protect against any profile in the MQADMIN class, such as s*sid.***, resolving to a RESLEVEL profile, a *ssid.RESLEVEL* profile will be defined for each queue manager and no users or groups specified in the access list.

- *(ZWMQ0060: CAT II) The IAO will ensure that a ssid.RESLEVEL profile is only defined for each queue manager.*

4.3.3.2.11 CICS Transaction Security

Access to MQSeries / WebSphere MQ-supplied CICS transactions will be controlled. The following Category 1 transactions will be restricted to CICS regions:

CKAM

CKTI

The following Category 4 transactions will be restricted to systems programming personnel and MQSeries administrators:

CKQC
CKBM
CKRT
CKCN
CKSQ
CKSD
CKRS
CKDP
CKDL

Refer to *Section 8.2, CICS,* for more information on CICS transaction security.

The following is a sample of the commands required to permit a CICS region (CICS1) to execute transaction CKTI:

```
SET RESOURCE(CKC)
COMPILE * STORE
$KEY(CKTI) TYPE(CKC)
- UID(CICS1) ALLOW
```

- *(ZCIC0020: CAT II) The IAO will ensure that MQSeries/WebSphere MQ-supplied CICS transactions are restricted to CICS regions and the WebSphere MQ administrator.*

4.3.4 RACF

The following subsections provide specific guidance and examples as to how MQSeries / WebSphere MQ should be secured using RACF. All of the following subsections follow the guidelines as specified in *Section 4.3.2, General Security Considerations.*

4.3.4.1 RACF Security Classes

In order to ensure that RACF checking is performed on all MQSeries/WebSphere MQ resources, the following RACF security classes should be activated:

MQADMIN	GMQADMIN
MQCONN	
MQCMDS	
MQQUEUE	GMQQUEUE
MQPROC	GMQPROC
MQNLIST	GMQNLIST

4.3.4.2 MQSeries/WebSphere MQ Resources

4.3.4.2.1 Started Tasks

Each queue manager started task procedure *xxxx*MSTR and distributed queuing started task procedure *xxxx*CHIN will have a matching profile defined to the STARTED resource class. Create a corresponding userid for each started task. The STC userids will be defined as PROTECTED userids. Queue manager and channel initiator started tasks will not be defined with the TRUSTED attribute.

4.3.4.2.2 Data Sets

Use the following recommendations to ensure that protection of the queue manager and channel initiator data sets is properly in place:

(1) The installation requires that the following data sets be APF authorized. (Refer to *Section 2.1.2.1, Authorized Program Facility [APF]*, for security guidelines and recommendations.)

hlqual.SCSQAUTH
hlqual.SCSQLINK
hlqual.SCSQANLx
hlqual.SCSQSNL
hlqual.SCSQMVR1
hlqual.SCSQMVR2

(2) *Read* access to data sets referenced by the CSQINP1, CSQINP2, and CSQXLIB DDs in the queue manager’s procedure will be restricted to the queue manager userid, WebSphere MQ administrator, and systems programming personnel. Log all access to these data sets.

(3) *Update* access to data set profiles protecting all page sets, logs, bootstrap data sets (BSDS), and data sets referenced by the CSQOUTX and CSQSNAP DDs in the queue manager’s procedure will be restricted to the queue manager userid, WebSphere MQ administrator, and systems programming personnel. Log all *update* and *alter* access to these data sets.

(4) *Alter* access to all archive data sets in the queue manager’s procedure will be restricted to the queue manager userid, WebSphere MQ administrator, and systems programming personnel. Log all *alter* access to these data sets.

- *(ZWMQ0040: CAT II) The IAO will ensure that all update and alter access to MQSeries/WebSphere MQ product and system data sets are restricted to WebSphere MQ administrators, systems programmers, and MQSeries/WebSphere MQ started tasks.*

4.3.4.2.3 Connection Security

Connection security validates userids authorized to connect to queue managers. Connection security will be active, and all profiles will be defined to the MQCONN class with UACC(NONE) specified.

Restrict access to connection security profiles using the following table as a guideline:

Table A-40. CONNECTION SECURITY CONTROLS (4.3.4.2.3)

PROFILE NAME	AUTHORIZED USERS	ACCESS	LOGGING REQUIRED
ssid.BATCH	TSO userids Batch job userids	Read	Y
ssid.CICS	CICS region userids	Read	Y
ssid.IMS	IMS region userids	Read	Y
ssid.CHIN	Channel initiator userids	Read	Y

NOTE: ssid is the name of the queue manager.

The following is a sample of the commands required to allow a batch user (USER1) to connect to a queue manager (QM1):

```
RDEFINE MQCONN QM1.BATCH UACC(NONE) AUDIT(ALL)
PERMIT QM1.BATCH CLASS(MQCONN) ID(USER1) ACCESS(READ)
```

- *(ZWMQ0052: CAT II) The IAO will ensure that all connections to MQSeries/WebSphere MQ resources are restricted using connection security.*

4.3.4.2.4 Queue Security

Queue security validates userids authorized to access message queues. Message queue access will be restricted to those userids that require the ability to get messages from and put messages to message queues. The profile names for queue security are *ssid.queuename*, where *ssid* is the name of a MQSeries/WebSphere MQ subsystem.

The following is a sample of the commands required to allow a user (USER1) to get messages from or put messages to queues beginning with (PAY.) on subsystem (QM1):

```
RDEFINE MQQUEUE QM1.PAY.** UACC(NONE)
PERMIT QM1.PAY.** CLASS(MQQUEUE) ID(USER1) ACCESS(UPDATE)
```

Access authorization to system queues (those queue resources with a first qualifier of *system*) will be restricted to the CSQUTIL utility, MQSeries / WebSphere MQ operations and control panels, channel initiators, MQSeries / WebSphere MQ software monitors, and CICS transactions.

- *(ZWMQ0054: CAT II) The IAO will ensure that queue security is activated , and all profiles are defined to the MQQUEUE or GMQQUEUE class with UACC(NONE) specified.*

4.3.4.2.4.1 Dead-Letter Queue Security

Whenever a message cannot be routed by a queue manager, the message is routed to the dead-letter queue assigned to that queue manager. Dead-letter queues are defined when the queue manager is created. The two levels of access for these queues are established using the PUT and GET attributes for the files. Unlike UNIX and NT, authority to *read* and *update* a dead-letter queue on an OS/390 platform is determined by the rules defined for the queue in the ACP. As a result, in order to further restrict messages from being read by unauthorized users, IBM recommends the use of an alias queue. An alias queue is a re-definition of the same queue. Alias queues are used to assign a different name and different attributes to a physical queue. By defining an alias queue for a dead-letter queue with its PUT option set to enabled (i.e., PUT(ENABLED) and its GET option set to disabled (i.e., GET(DISABLED), messages can be further secured. The first level allows applications, as well as some MQSeries / WebSphere MQ objects, to put messages to this queue. The second level restricts the ability to get messages from this queue and protects sensitive data. The ability to get messages from the dead-letter queue will be restricted to message channel agents (MCAs), CKTI (MQSeries / WebSphere MQ-supplied CICS task initiator), channel initiators utility, WebSphere MQ administrators, and any automated application used for dead-letter queue maintenance.

The following scenario describes how a dead-letter queue should be defined securely:

(1) Define the real dead-letter queue with attributes PUT(ENABLED) and GET(ENABLED).

(2) Give RACF *update* authority for the dead-letter queue to CKTI, the MQSeries / WebSphere MQ-supplied CICS task initiator, channel initiators, and any automated application used for dead-letter queue maintenance.

(3) Define an alias queue that resolves to the real dead-letter queue, but give the alias queue the attributes PUT(ENABLED) and GET(DISABLED).

(4) To put a message on the dead-letter queue, an application uses the alias queue. The application does the following:

  (a) Retrieve the name of the real dead-letter queue. To do this, it opens the queue manager object using MQOPEN and then issues an MQINQ to get the dead-letter queue name.

  (b) Build the name of the alias queue by appending the characters ".PUT" to this name, in this case, *ssid*.DEAD.QUEUE.PUT.

  (c) Open the alias queue, *ssid*.DEAD.QUEUE.PUT.

  (d) Put the message on the real dead-letter queue by issuing an MQPUT against the alias queue.

(5) Give the userid associated with the application RACF *update* authority to the alias, but no access (authority NONE) to the real dead-letter queue.

NOTE: If an alias queue is not used in place of the dead-letter queue, then the RACF rules for the dead-letter queue will be coded to restrict unauthorized users and systems from reading the messages on the file.

- *(ZWMQ0053: CAT II) The systems programmer responsible for supporting MQSeries. WebSphere MQ will ensure that the dead-letter queue and its alias are properly defined.*

- *(ZWMQ0054: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ queues are restricted using queue level security.*

4.3.4.2.5 Process Security

Process security validates userids authorized to issue MQSeries / WebSphere MQ inquiries on process definitions. A process definition object defines an application that is started in response to a trigger event on a queue manager. Process security will be active, and all profiles *ssid.processname* will be defined to the MQPROC or GMQPROC class with UACC(NONE) specified. Restrict *read* access to those userids requiring access to make process inquiries.

The following is a sample of the commands required to allow a group (GRP1) to inquire on processes beginning with the letter **V** on queue manager (QM1):

```
RDEFINE MQPROC QM1.V* UACC(NONE) AUDIT(ALL)
PERMIT QM1.V* CLASS(MQPROC) ID(GRP1) ACCESS(READ)
```

- *(ZWMQ0055: CAT II) The IAO will ensure that process security is active, and that all profiles defined to the MQPROC class and that process inquiries are restricted to read access.*

4.3.4.2.6 Namelist Security

A namelist is a MQSeries / WebSphere MQ object that contains a list of queue names. Namelist security validates userids authorized to inquire on namelists. Namelist security will be active, and all profiles *ssid.namelist* will be defined to the **MQNLIST** or **GMQNLIST** class with UACC(NONE) specified. Restrict *read* access to those userids requiring access to make namelist inquiries.

The following is a sample of the commands required to allow a group (GRP1) to inquire on namelist TST1 on queue manager (QM1):

```
RDEFINE MQNLIST QM1.TST1.** UACC(NONE) AUDIT(ALL)
PERMIT QM1.TST1.** CLASS(MQNLIST) ID(GRP1) ACCESS(READ)
```

- *(ZWMQ0056: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ namelist resources are restricted to authorized users.*

4.3.4.2.7 Alternate Userid Security

Alternate userid security allows access to be requested under another userid. Alternate userid security will be active, and all profiles *ssid.ALTERNATE.USER.alternateuserid* will be defined to the MQADMIN class with UACC(NONE) specified. Restrict *update* access to those userids requiring access to alternate userids.

The following is a sample of the commands required to allow payroll server (PAYSRV1) to specify alternate userids starting with the characters PS on queue manager (QM1):

```
REDEFINE MQADMIN QM1.ALTERNATE.USER.PS* UACC(NONE)
    AUDIT(ALL)
PERMIT QM1.ALTERNATE.USER.PS* CLASS(MQADMIN) ID(PAYSRV1)
    ACCESS(UPDATE)
```

- *(ZWMQ0057: CAT II) The IAO will ensure that use of alternate userids is restricted to authorized personnel.*

4.3.4.2.8 Context Security

Context security validates whether a userid has authority to pass or set identity and/or origin data for a message. Context security will be active, and all profiles *ssid.CONTEXT* will be defined to the MQADMIN class with UACC(NONE) specified, where *ssid* is the queue manager name. *Read* access is required when the PASS option is specified for an MQOPEN or MQPUT1. *Update* or *control* access is required when the SET or OUTPUT option is specified.

The following is a sample of the commands required to allow a systems programming group (SYS1) to offload and reload messages for queue manager (QM1):

```
RDEFINE MQADMIN QM1.CONTEXT UACC(NONE) AUDIT(ALL)
PERMIT QM1.CONTEXT CLASS(MQADMIN) ID(SYS1) ACCESS(CONTROL)
```

- *(ZWMQ0058: CAT II) The IAO will ensure that use of context resources are restricted to authorized personnel.*

4.3.4.2.9 Command Security

Command security validates userids authorized to issue MQSeries / WebSphere MQ commands. Command security will be active, and all profiles will be defined to the MQCMDS class with UACC(NONE) specified.

Restrict access to command security profiles using the following table as a guideline:

Table A-41. COMMAND SECURITY CONTROLS (4.3.4.2.9)

COMMAND	PROFILE	ACCESS LEVEL	AUTHORIZED USERS	LOG
ALTER *xxxxx*	ssid.ALTER.xxxxx	ALTER	MQ administrator Systems programmers Queue managers	Y
ARCHIVE LOG	*ssid*.ARCHIVE.LOG	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
CLEAR QLOCAL	*ssid*.CLEAR.QLOCAL	ALTER	MQ administrator Systems programmers Queue managers	Y
DEFINE *xxxxx*	ssid.DEFINE.xxxxx	ALTER	MQ administrator Systems programmers Queue managers	Y
DELETE *xxxxx*	ssid.DELETE.xxxxx	ALTER	MQ administrator Systems programmers Queue managers	Y
DISPLAY *xxxxx*	*ssid*.DISPLAY.*xxxxx*	READ	Application programmers MQ administrator Systems programmers Queue manager Operators Console automation software	N
PING *xxxxx*	ssid.PING.xxxxx	CONTROL	Application programmers MQ administrator Systems programmers Queue managers Operators Console automation software	N
RECOVER BSDS	*ssid*.RECOVER.BSDS	CONTROL	MQ administrator Systems programmers Queue managers	Y
REFRESH *xxxxx*	*ssid*.REFRESH.*xxxxx*	ALTER	Security staff MQ administrator Systems programmers Queue managers	Y
RESET *xxxxx*	ssid.RESET.xxxxx	CONTROL	MQ administrator Systems programmers Queue managers	Y

COMMAND	PROFILE	ACCESS LEVEL	AUTHORIZED USERS	LOG
RESOLVE *xxxxx*	ssid.RESOLVE.xxxxx	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
RESUME QMGR	*ssid*.RESUME.QMGR	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
RVERIFY SECURITY	*ssid*.RVERIFY.SECURITY	ALTER	Security staff MQ administrator	Y
START *xxxxx*	ssid.START.xxxxx	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
STOP *xxxxx*	*ssid*.STOP.CHINIT	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
SUSPEND QMGR	*ssid*.SUSPEND.QMGR	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y

The following is a sample of the commands required to allow a systems programming group (SYS1) to issue the command CLEAR QLOCAL in subsystem QM1:

```
RDEFINE MQCMDS QM1.CLEAR.LOCAL UACC(NONE) AUDIT(ALL)
PERMIT QM1.CLEAR.LOCAL CLASS(MQCMDS) ID(SYS1) ACCESS(ALTER)
```

- *(ZWMQ0059: CAT II) The IAO will ensure that all MQSeries / WebSphere MQ commands are restricted to authorized personnel.*

4.3.4.2.10 RESLEVEL Security

RESLEVEL security profiles control the number of userids checked for API-resource security. RESLEVEL security will not be implemented due to the following exposures and limitations:

(1) RESLEVEL is a powerful option that can cause the bypassing of all security checks.

(2) Security audit records are not created when the RESLEVEL profile is utilized.

(3) If the WARNING option is specified on a RESLEVEL profile, no warning messages are produced.

To protect against any profile in the MQADMIN class, such as ssid.**, resolving to a RESLEVEL profile, a *ssid.RESLEVEL* profile will be defined for each queue manager with UACC(NONE) specified and no users or groups specified in the access list.

- *(ZWMQ0060: CAT II) The IAO will ensure that a ssid.RESLEVEL profile is only defined for each queue manager.*

4.3.4.2.11 CICS Transaction Security

Access to MQSeries / WebSphere MQ-supplied CICS transactions will be controlled. The following Category 1 transactions will be restricted to CICS regions:

CKAM
CKTI

The following Category 4 transactions will be restricted to systems programming personnel and MQSeries administrators:

CKQC
CKBM
CKRT
CKCN
CKSQ
CKSD
CKRS
CKDP
CKDL

Refer to *Section 8.2, CICS*, for more information on CICS transaction security.

The following is a sample of the commands required to permit a CICS region (CICS1) to execute transaction CKTI:

```
RDEFINE TCICSTRN CKTI UACC(NONE)
PERMIT CKTI CLASS(TCICSTRN) ID(CICS1) ACCESS(FETCH)
```

- *(ZCIC0020: CAT II) The IAO will ensure that MQSeries/WebSphere MQ-supplied CICS transactions are restricted to CICS regions and the WebSphere MQ administrator.*

4.3.5 CA-TOP SECRET

The following subsections provide specific guidance and examples as to how MQSeries/WebSphere MQ should be secured using CA-TOP SECRET. All of the following subsections follow the guidelines as specified in *Section 4.3.2, General Security Considerations.*

NOTE : TOP SECRET should be installed at the 4.4 9405 or higher level to support MQSeries/WebSphere MQ.

NOTE : Refer to Section 7.13 of the CA-TOP SECRET User Guide, Protecting Message Queue Manager Resources, for additional information.

4.3.5.1 TOP SECRET Security Classes

In order to ensure that TOP SECRET checking is performed on all MQSeries / WebSphere MQ resources, the following RDT entries should exist and be properly owned:

MQADMIN
MQCONN
MQCMDS
MQQUEUE
MQPROC
MQNLIST

Use the following commands to define (establish ownership of) resources for each MQSeries/WebSphere MQ subsystem to TOP SECRET:

TSS ADD(*deptname*) MQADMIN(*da.*)
TSS ADD(*deptname*) MQQUEUE(*da.*)
TSS ADD(*deptname*) MQNLIST(*da.*)
TSS ADD(*deptname*) MQCMDS(*da.*)
TSS ADD(*deptname*) MQPROC(*da.*)
TSS ADD(*deptname*) MQCONN(*da.*)

NOTE: da is the name of the subsystem ID.

4.3.5.2 MQSeries/WebSphere MQ Resources

4.3.5.2.1 Started Tasks

Create a Started Task Table entry for each queue manager started task procedure *xxxx*MSTR and distributed queuing started task procedure *xxxx*CHIN. Create a corresponding userid for each started task. Queue manager and channel initiator started tasks will not be defined with the *BYPASS* attribute.

Define each queue manager *xxxx*MSTR to the TOP SECRET Facility Matrix Table using the following sample commands:

```
FACILITY(USERxx=NAME=xxxxMSTR)
FACILITY(xxxxMSTR=MODE=FAIL,PGM=CSQ,ID=xx)
FACILITY(xxxxMSTR=ACTIVE,SHRPRT,ASUBM,NOABEND)
FACILITY(xxxxMSTR=MULTUSER,XDEF,LUMSG,STMSG,SIGN(S))
FACILITY(xxxxMSTR=INSTDATA,NORNDPW,AUTHINIT)
FACILITY(xxxxMSTR=NOPROMPT,NOAUDIT,RES,WARNPW)
FACILITY(xxxxMSTR=NOTSOC,LCFTRANS,IJU,MSGLC,NOTRACE)
FACILITY(xxxxMSTR=NOEODINIT,NODORMPW,NONPWR)
FACILITY(xxxxMSTR=NOIMSXTND,LOG(INIT,SMF,MSG,SEC9))
FACILITY(xxxxMSTR=DOWN=GLOBAL,LOCKTIME=00,DEFACID=(*NONE*))
```

NOTE: It is not necessary to define each distributed queuing xxxxCHIN as a separate FACility.

4.3.5.2.2 Data Sets

Use the following recommendations to ensure that protection of the queue manager and channel initiator data sets is properly in place:

(1) The installation requires that the following data sets be APF authorized. (Refer to *Section 2.1.2.1, Authorized Program Facility [APF]*, for security guidelines and recommendations.)

hlqual.SCSQAUTH
hlqual.SCSQLINK
hlqual.SCSQANLx
hlqual.SCSQSNL
hlqual.SCSQMVR1
hlqual.SCSQMVR2

(2) *Read* access to data sets referenced by the CSQINP1, CSQINP2, and CSQXLIB DDs in the queue manager's procedure will be restricted to the queue manager userid, WebSphere MQ administrator, and systems programming personnel. Log all access to these data sets.

(3) *Update* access to data set profiles protecting all page sets, logs, bootstrap data sets (BSDS), and data sets referenced by the CSQOUTX and CSQSNAP DDs in the queue manager's procedure will be restricted to the queue manager userid, WebSphere MQ administrator, and systems programming personnel. Log all *update* and *alter* access to these data sets.

(4) *Alter* access to all archive data sets in the queue manager's procedure will be restricted to the queue manager userid, WebSphere MQ administrator, and systems programming personnel. Log all *alter* access to these data sets.

- *(ZWMQ0040: CAT II) The IAO will ensure that all update and alter access to MQSeries/WebSphere MQ product and system data sets are restricted to WebSphere MQ administrators, systems programmers, and MQSeries/WebSphere MQ started tasks.*

4.3.5.2.3 Connection Security

Connection security validates userids authorized to connect to queue managers. Connection security will be active, and all profiles will be defined to the MQCONN class. Restrict access to connection security profiles using the following table:

Table A-42. CONNECTION SECURITY CONTROLS (4.3.5.2.3)

PROFILE NAME	AUTHORIZED USERS	ACCESS	LOGGING REQUIRED
ssid.BATCH	TSO userids Batch job userids	Read	Y
ssid.CICS	CICS region userids	Read	Y
ssid.IMS	IMS region userids	Read	Y
ssid.CHIN	Channel initiator userids	Read	Y

NOTE: ssid is the name of the queue manager.

The following is a sample of the commands required to allow a batch user (USER1) to connect to a queue manager (QM1):

```
TSS ADD(USER1) FAC(QM1MSTR)
TSS PER(USER1) MQCONN(QM1.BATCH) ACC(READ) ACTION(AUDIT)
```

- *(ZWMQ0052: CAT II) The IAO will ensure that all connections to MQSeries/WebSphere MQ resources are restricted using connection security.*

4.3.5.2.4 Queue Security

Queue security validates userids authorized to access message queues. Queue security will be active, and all profiles will be defined to the MQQUEUE class.

Message queue access will be restricted to those userids that require the ability to get messages from and put messages to message queues. The profile names for queue security are *ssid.queuename*, where *ssid* is the name of a MQSeries / WebSphere MQ subsystem.

The following is a sample of the commands required to allow a user (USER1) to get messages from or put messages to queues beginning with (PAY.) on subsystem (QM1):

```
TSS ADD(USER1) FAC(QM1MSTR)
TSS PER(USER1) MQQUEUE(QM1.PAY.) ACC(UPDATE)
```

Access authorization to system queues (those queue resources with a first qualifier of *system*) will be restricted to the CSQUTIL utility, MQSeries / WebSphere MQ operations and control panels, channel initiators, MQSeries / WebSphere MQ software monitors, and CICS transactions.

- *(ZWMQ0054: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ queues are restricted using queue level security.*

4.3.5.2.4.1 Dead-Letter Queue Security

Undeliverable messages can be routed to a dead-letter queue. Two levels of access should be established for these queues. The first level allows applications, as well as some MQSeries / WebSphere MQ objects, to put messages to this queue. The second level restricts the ability to get messages from this queue and protects sensitive data. This will be accomplished by defining an alias queue that resolves to the real dead-letter queue, but defines the alias queue with the attributes PUT(ENABLED) and GET(DISABLED). The ability to get messages from the dead-letter queue will be restricted to message channel agents (MCAs), CKTI (MQSeries/WebSphere MQ-supplied CICS task initiator), channel initiators utility, and any automated application used for dead-letter queue maintenance.

The following scenario describes how to securely define a dead-letter queue:

(1) Define the real dead-letter queue with attributes PUT(ENABLED) and GET(ENABLED).

(2) Give *update* authority for the dead-letter queue to CKTI (the MQSeries/WebSphere MQ-supplied CICS task initiator), channel initiators, and any automated application used for dead-letter queue maintenance.

(3) Define an alias queue that resolves to the real dead-letter queue, but give the alias queue the attributes PUT(ENABLED) and GET(DISABLED).

(4) To put a message on the dead-letter queue, an application uses the alias queue. The application does the following:

(a) Retrieve the name of the real dead-letter queue. To do this, it opens the queue manager object using MQOPEN, and then issues an MQINQ to get the dead-letter queue name.

(b) Build the name of the alias queue by appending the characters "**.**PUT" to this name, in this case, *ssid*.DEAD.QUEUE.PUT.

(c) Open the alias queue, *ssid*.DEAD.QUEUE.PUT.

(d) Put the message on the real dead-letter queue by issuing an MQPUT against the alias queue.

(5) Give the userid associated with the application *update* authority to the alias, but no access to the real dead-letter queue.

NOTE: If an alias queue is not used in place of the dead-letter queue, then the TSS rules for the dead-letter queue will be coded to restrict unauthorized users and systems from reading the messages on the file.

- *(ZWMQ0053: CAT II) The systems programmer responsible for supporting MQSeries/WebSphere MQ will ensure that the dead-letter queue and its alias are properly defined.*

- *(ZWMQ0054: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ queues are restricted using queue level security.*

4.3.5.2.5 Process Security

Process security validates userids authorized to issue MQSeries / WebSphere MQ inquiries on process definitions. A process definition object defines an application that is started in response to a trigger event on a queue manager. Process security will be active, and all profiles *ssid.processname* will be defined to the MQPROC class. Restrict *read* access to those userids requiring access to make process inquiries.

The following is a sample of the commands required to allow a user (USER1) to inquire on processes beginning with the letter **V** on queue manager (QM1):

```
TSS ADD(USER1) FAC(QM1MSTR)
TSS PER(USER1) MQPROC(QM1.V) ACC(READ) ACTION(AUDIT)
```

- *(ZWMQ0055: CAT II) The IAO will ensure that process security is active, and that all profiles defined to the MQPROC class and that process inquiries are restricted to read access.*

4.3.5.2.6 Namelist Security

A namelist is a MQSeries/WebSphere MQ object that contains a list of queue names. Namelist security validates userids authorized to inquire on namelists. Namelist security will be active, and all profiles *ssid.namelist* will be defined to the MQNLIST class with UACC(NONE) specified. Restrict *read* access to those userids requiring access to make namelist inquiries.

The following is a sample of the commands required to allow a user (USER1) to inquire on namelist TST1 on queue manager (QM1):

```
TSS ADD(USER1) FAC(QM1MSTR)
TSS PER(USER1) MQNLIST(QM1.TST1.) ACC(READ) ACTION(AUDIT)
```

- *(ZWMQ0056: CAT II) The IAO will ensure that all MQSeries / WebSphere MQ namelist resources are restricted to authorized users.*

4.3.5.2.7 Alternate Userid Security

Alternate userid security allows access to be requested under another userid. Alternate userid security will be active, and all profiles *ssid.ALTERNATE.USER.alternateuserid* will be defined to the MQADMIN class. Restrict *update* access to those userids requiring access to alternate userids.

The following is a sample of the commands required to allow payroll server (PAYSRV1) to specify alternate userids starting with the characters PS on queue manager (QM1):

```
TSS ADD(USER1) FAC(QM1MSTR)
TSS PER(USER1) MQADMIN(QM1.ALTERNATE.USER.PS) ACC(UPDATE)
    ACTION(AUDIT)
```

- *(ZWMQ0057: CAT II) The IAO will ensure that use of alternate userids is restricted to authorized personnel.*

4.3.5.2.8 Context Security

Context security validates whether a userid has authority to pass or set identity and/or origin data for a message. Context security will be active, and all profiles *ssid.CONTEXT* will be defined to the **MQADMIN** class, where *ssid* is the queue manager name. *Read* access is required when the PASS option is specified for an MQOPEN or MQPUT1. *Update* or *control* access is required when the SET or OUTPUT option is specified.

The following is a sample of the commands required to allow a systems programming group (SYS1) to offload and reload messages for queue manager (QM1):

```
TSS ADD(SYS1) FAC(QM1MSTR)
TSS PER(SYS1) MQADMIN(QM1.CONTEXT) ACC(UPDATE) ACTION(AUDIT)
```

- *(ZWMQ0058: CAT II) The IAO will ensure that use of context resources are restricted to authorized personnel.*

4.3.5.2.9 Command Security

Command security validates userids authorized to issue MQSeries/WebSphere MQ commands. Command security will be active, and all profiles will be defined to the MQCMDS class. Restrict access to command security profiles using the following table:

Table A-43. COMMAND SECURITY CONTROLS (4.3.5.2.9)

COMMAND	PROFILE	ACCESS LEVEL	AUTHORIZED USERS	LOG
ALTER *xxxxx*	ssid.ALTER.xxxxx	ALL	MQ administrator Systems programmers Queue managers	Y
ARCHIVE LOG	*ssid*.ARCHIVE.LOG	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y

COMMAND	PROFILE	ACCESS LEVEL	AUTHORIZED USERS	LOG
CLEAR QLOCAL	*ssid*.CLEAR.QLOCAL	ALL	MQ administrator Systems programmers Queue managers	Y
DEFINE *xxxxx*	ssid.DEFINE.xxxxx	ALL	MQ administrator Systems programmers Queue managers	Y
DELETE *xxxxx*	ssid.DELETE.xxxxx	ALL	MQ administrator Systems programmers Queue managers	Y
DISPLAY *xxxxx*	*ssid*.DISPLAY.*xxxxx*	READ	Application programmers MQ administrator Systems programmers Queue managers Operators Console automation software	N
PING *xxxxx*	ssid.PING.xxxxx	CONTROL	Application programmers MQ administrator Systems programmers Queue managers Operators Console automation software	N
RECOVER BSDS	*ssid*.RECOVER.BSDS	CONTROL	MQ administrator Systems programmers Queue managers	Y
REFRESH *xxxxx*	*ssid*.REFRESH.*xxxxx*	ALL	Security staff MQ administrator Systems programmers Queue managers	Y
RESET *xxxxx*	ssid.RESET.xxxxx	CONTROL	MQ administrator Systems programmers Queue managers	Y
RESOLVE *xxxxx*	ssid.RESOLVE.xxxxx	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y

COMMAND	PROFILE	ACCESS LEVEL	AUTHORIZED USERS	LOG
RESUME QMGR	*ssid*.RESUME.QMGR	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
RVERIFY SECURITY	*ssid*.RVERIFY.SECURITY	ALL	Security staff MQ administrator	Y
START *xxxxx*	ssid.START.xxxxx	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
STOP *xxxxx*	*ssid*.STOP.CHINIT	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y
SUSPEND QMGR	*ssid*.SUSPEND.QMGR	CONTROL	MQ administrator Systems programmers Queue managers Operators Console automation software	Y

The following is a sample of the commands required to allow a systems programming group (SYS1) to issue the command CLEAR QLOCAL in subsystem QM1:

```
TSS ADD(SYS1) FAC(QM1MSTR)
TSS PER(SYS1) MQCMDS(QM1.CLEAR.LOCAL) ACC(ALTER)
    ACTION(AUDIT)
```

- *(ZWMQ0059: CAT II) The IAO will ensure that all MQSeries/WebSphere MQ commands are restricted to authorized personnel.*

4.3.5.2.10 RESLEVEL Security

RESLEVEL security profiles control the number of userids checked for API resource security. RESLEVEL security will not be implemented due to the following exposures and limitations:

(1) RESLEVEL is a powerful option that can cause the bypassing of all security checks.

(2) Security audit records are not created when the RESLEVEL profile is utilized.

(3) If the WARNING option is specified on a RESLEVEL profile, no warning messages are produced.

In order to protect against any profile in the MQADMIN class, such as s*sid*.**, resolving to a RESLEVEL profile, an *ssid*.RESLEVEL permission will be created for each queue manager with an access of *none*.

The following sample command prevents access to *ssid*.RESLEVEL:

```
TSS PER(ALL) MQADMIN(ssid.RESLEVEL) ACCESS(NONE)
```

- *(ZWMQ0060: CAT II) The IAO will ensure that a ssid.RESLEVEL profile is only defined for each queue manager.*

4.3.5.2.11 CICS Transaction Security

Access to MQSeries / WebSphere MQ-supplied CICS transactions will be controlled. The following Category 1 transactions will be restricted to CICS regions:

CKAM
CKTI

The following Category 4 transactions will be restricted to systems programming personnel and MQSeries administrators:

CKQC
CKBM
CKRT
CKCN
CKSQ
CKSD
CKRS
CKDP
CKDL

Refer to *Section 8.2, CICS*, for more information on CICS transaction security.

The following is a sample of the commands required to permit a CICS region (CICS1) to execute transaction CKTI:

```
TSS ADD(deptname) OTRAN(CKTI)
TSS PER(CICS1) OTRAN(CKTI) ACC(EXECUTE,INQUIRE)
```

- *(ZCIC0020: CAT II) The IAO will ensure that MQSeries/WebSphere MQ-supplied CICS transactions are restricted to CICS regions and the WebSphere MQ administrator.*

4.4 IBM Communications Server for OS/390 – TCP/IP

The IBM Communications Server (ICS) for OS/390 is a base element of OS/390. It consists of TCP/IP and SNA functions that support secure networking on an enterprise scale. In releases of OS/390 prior to Release 2.10, this product was called SecureWay Communications Server and eNetwork Communications Server.

This document section addresses a subset of the TCP/IP functions that are packaged in OS/390. These functions are delivered in a group of components that include the Base TCP/IP System as well as several application servers and clients. These components include applications designed to run in the MVS and OS/390 UNIX environments. The components covered here are those most likely to be in use on OS/390 hosts. This includes the following:

The Base TCP/IP System
The TN3270 Telnet Server
The OS/390 UNIX Telnet Server
The FTP Server
The FTP Client
The TFTP Server
The Syslog Daemon

It should be noted that the information in this section has been prepared to address the configuration requirements as of Version 2, Releases 8 and 10 of OS/390. IBM made significant changes to the software between these releases, and parameter differences reflect these changes. Please refer to the *OS/390 IBM Communications Server IP Configuration Guide* document (OS/390 Release 2.10) or the *OS/390 SecureWay Communications Server IP Configuration* document (OS/390 Release 2.8) for more release-specific information.

4.4.1 Base TCP/IP System

The Base TCP/IP System is a term used to describe those parts of the IBM Communications Server for OS/390 that are not application servers or clients. This is primarily the TCP/IP stack itself. The TCP/IP stack runs in an OS/390 address space and provides services that TCP/IP application servers and clients use to communicate with each other and outside networks.

4.4.1.1 General Considerations

The configuration issues addressed relative to the security environment for the Base TCP/IP System include the following:

- Configuration files for the TCP/IP stack
- Configuration files shared by TCP/IP applications
- Configuration statements in the TCPIP.DATA file
- Configuration statements in the PROFILE.TCPIP file
- The AUTOLOG subtask
- SAF Server Access Authorization (SERVAUTH)
- The VMCF and TNF Subsystems
- Sensitive commands

4.4.1.1.1 Configuration Files – TCP/IP Stack

The TCP/IP stack reads two configuration files to determine values for operational parameters. Because system security is impacted by some of the parameter settings, the files themselves should be protected and certain parameter settings should be specified in those files. This section discusses the files and how their names are specified.

During initialization the TCP/IP stack uses fixed search sequences to locate the PROFILE.TCPIP and TCPIP.DATA files. However, uncertainty is reduced and security auditing is enhanced by explicitly specifying the locations of the files. In the TCP/IP started task's JCL, Data Definition (DD) statements can be used to specify the locations of the files. The PROFILE DD statement identifies the PROFILE.TCPIP file and the SYSTCPD DD statement identifies the TCPIP.DATA file.

The location of the TCPIP.DATA file can also be specified by coding the RESOLVER_CONFIG environment variable as a parameter of the ENVAR option in the TCP/IP started task's JCL. In fact, the value of this variable is checked before the SYSTCPD DD statement by some processes. However, not all processes (e.g., TN3270 Telnet Server) will access the variable to get the file location. Therefore specifying the file location explicitly, both on a DD statement and through the RESOLVER_CONFIG environment variable, reduces ambiguity.

- *(ITCP0010: CAT II) The systems programmer responsible for supporting ICS will ensure that the TCP/IP started task's JCL specifies the PROFILE and SYSTCPD DD statements for the PROFILE.TCPIP and TCPIP.DATA configuration files and TCP/IP started task's JCL includes the RESOLVER_CONFIG variable, set to the name of the file specified on the SYSTCPD DD statement.*

Required access controls for the PROFILE.TCPIP and TCPIP.DATA configuration files are described in the ACP-specific subsections that follow.

The PROFILE.TCPIP file can be physically segmented into separate MVS data sets through the use of the INCLUDE statement. The files named on an INCLUDE statement are logically embedded into the PROFILE.TCPIP file. The access controls for any of these data sets have to match those in effect for the PROFILE.TCPIP file.

4.4.1.1.2 Configuration Files – Shared By TCP/IP Applications

Many TCP/IP applications read configuration files to determine values for operational parameters. Parameters related to common functions, such as host name resolution, should usually have the same values for all applications. The most efficient way to ensure this is to share one copy of a file among various applications. Shared files should be properly secured so that they are not accidentally or intentionally altered or deleted. This section discusses common TCP/IP application configuration files and how their names are specified.

Name resolution is a function, called by many TCP/IP applications, that requires configuration information. Name resolution refers to resolving host names into IP addresses. There are several issues that complicate the process of specifying configuration files for name resolution:

OS/390 includes two resolver components, each with its own programmatic interface to name resolution. The first is referred to as native MVS; the second is referred to as OS/390 UNIX.

Each of the two resolvers uses a different search sequence to locate configuration files. The search sequences include checks of environment variables, explicit JCL specifications, and dynamic allocation using default names.

Although a domain name server (specified via IP address) is commonly used, it is possible to use additional files for lookup information instead of, or in addition to, the name server. In the case of the OS/390 UNIX resolver, the search sequence for these files includes HFS files in addition to MVS files.

The following table lists the names of configuration files that are defaults within the search sequences used by IBM TCP/IP applications. Please refer to the *OS/390 IBM Communications Server IP Configuration Guide* document (OS/390 Release 2.10) or the *OS/390 SecureWay Communications Server IP Configuration* document (OS/390 Release 2.8) for documentation of each specific search sequence.

Table A-44. TCP/IP SHARED CONFIGURATION FILES (4.4.1.1.2 a)

TCP/IP SHAREDCONFIGURATION FILES		
FUNCTION	MVS FILE NAME	HFS FILE NAME
Base Resolver configuration	*hlq*.TCPIP.DATA	/etc/resolv.conf
Local hosts tables	*hlq*.HOSTS.SITEINFO	/etc/hosts
	hlq.HOSTS.ADDRINFO	
Protocol information	*hlq*.ETC.PROTO	/etc/protocol
Service information	*hlq*.ETC.SERVICES	/etc/services
Translate table	*hlq*.STANDARD.TCPXLBIN	[n/a]

In this table, the prefix hlq stands for high-level qualifier. During file searches, this high-level qualifier is the value of the DATASETPREFIX parameter specified in the TCPIP.DATA file. If the TCPIP.DATA file itself is not found through an explicit reference or otherwise in the defined search sequence, the data set name TCPIP.TCPIP.DATA is attempted.

Required access controls for the TCP/IP shared configuration files are described in the HFS Object Protection and ACP-specific subsections that follow.

4.4.1.1.3 TCPIP.DATA Configuration Statements

The TCPIP.DATA file acts as the anchor configuration data set for the TCP/IP stack and all TCP/IP servers and clients running in OS/390. During the initialization of TCP/IP servers and clients, the TCPIP.DATA file provides basic information that is essential for proper operation. The TCPIP.DATA file is also used as one of the name resolver configuration data sets. This section describes the configuration parameters that can impact security and the applicable guidelines.

The following table describes several TCPIP.DATA configuration parameters that provide crucial information to TCP/IP applications:

Table A-45. BASE TCP/IP TCPIP.DATA CONFIGURATION STATEMENTS (4.4.1.1.3 a)

BASE TCP/IP TCPIP.DATA CONFIGURATION STATEMENTS	
STATEMENT	FUNCTION
TCPIPJOBNAME	Specifies the job name of the TCP/IP address space. This name is also used as part of the name of some network security resources.
HOSTNAME	Specifies the TCP/IP host portion of the DNS name of the system.
DOMAINORIGIN	Specifies the default domain name used for DNS searches.
DATASETPREFIX	Specifies the high-level qualifier to be used to dynamically allocate other configuration data sets.
NSINTERADDR	Specifies the IP address of a host running a DNS server. Multiple NSINTERADDR statements can be used to specify alternate servers.

- *(ITCP0020: CAT II) The systems programmer responsible for supporting ICS will ensure that the TCPIPJOBNAME, HOSTNAME, DOMAINORIGIN, DATASETPREFIX, and NSINTERADDR statements are coded in the TCPIP.DATA file.*

- *(ITCP0025: CAT II) The IAO will ensure that if any NSINTERADDR statements are coded in the TCPIP.DATA file, they refer to hosts connected directly to networks within the physical premises of the host site and located in areas with physical access limited to authorized personnel.*

Please refer to the *Network Infrastructure Security Technical Implementation Guide*, for guidance on Domain Name Service (DNS) servers.

4.4.1.1.4 PROFILE.TCPIP Configuration Statements

The PROFILE.TCPIP file provides system operation and configuration parameters for the TCP/IP stack. The following groups of configuration parameters are included:

- Operating characteristics
- Physical characteristics
- Port number reservations
- Network routing definitions
- Diagnostic data statements

The following table describes several configuration parameters within these groups that can impact security:

Table A-46. BASE TCP/IP PROFILE.TCPIP CONFIGURATION STATEMENTS (4.4.1.1.4 a)

BASE TCP/IP PROFILE.TCPIP CONFIGURATION STATEMENTS	
STATEMENT	FUNCTION AND SECURITY IMPACT
AUTOLOG	- Specifies the tasks (via JCL PROC names) that should be started when the TCPIP address space starts and, optionally, restarted when a hung condition is detected - Starts system tasks
BEGINVTAM	[See NOTE following the table]
DELETE	- Specifies some previous statements, including PORT and PORTRANGE, that are to be deleted - Alters the configuration specified by previous statements
INCLUDE	- Specifies the name of an MVS data set that contains additional PROFILE.TCPIP statements to be used - Alters the configuration specified by previous statements
IPCONFIG	- Specifies various settings for the IP layer of TCP/IP - Controls routing of data to other networks

BASE TCP/IP PROFILE.TCPIP CONFIGURATION STATEMENTS	
STATEMENT	FUNCTION AND SECURITY IMPACT
NETACCESS (OS/390 Release 2.10)	- Configures network access control using a map to SAF resource names - Controls access to other hosts and networks
PORT (Operands added in OS/390 Release 2.10)	- Specifies an IP port number that is reserved for specific task(s) via job name, restricted to certain IDs via SAF resource name, or restricted from any use - Controls port access
PORTRANGE (Operands added in OS/390 Release 2.10)	- Specifies a range of IP port numbers that are reserved for specific task(s) via job name, restricted to certain IDs via SAF resource name, or restricted from any use - Controls port access
SMFCONFIG	- Specifies SMF logging options for Telnet, FTP, TCP, API, and stack activity - Controls collection of audit data
SMFPARMS	- Specifies SMF logging options for some TCP applications; replaced by SMFCONFIG - Controls collection of audit data
TCPCONFIG	- Specifies various settings for the TCP protocol layer of TCP/IP - Controls port access
TELNETGLOBALS (OS/390 Release 2.10)	[See NOTE following the table]
TELNETPARMS	[See NOTE following the table]
UDPCONFIG	- Specifies various settings for the UDP protocol layer of TCP/IP - Controls port access

NOTE: The TELNETGLOBALS (OS/390 Release 2.10), TELNETPARMS, and BEGINVTAM statement blocks are used to configure the TN3270 Telnet Server and are not described here. Please refer to *Section 4.4.2, TN3270 Telnet Server*, for guidelines on those statements.

The following guidelines are recommended to enhance security:

- The AUTOLOG statement should be used to start the FTP Server and other IBM TCP/IP server applications that can be initiated as started tasks. Using the AUTOLOG facility can enhance reliability, reduce operator intervention, and simplify security issues.

- Where appropriate to the site’s network configuration, the IPCONFIG statement should include the NODATAGRAMFWD operand. This disables IP routing between different network interfaces.

- For systems at OS/390 Release 2.10 and above, the NETACCESS statement should be considered for limiting access to outside networks. Please refer to *Section 4.4.1.1.6, SAF Server Access Authorization (SERVAUTH)*, for additional information.

- A PORT or PORTRANGE statement with the job name operand should be used for ports used by well-known servers such as FTP (20, 21) and Telnet (23) as well as other servers for which ports should be reserved. The job name operand can be, as appropriate, the actual job name, OMVS (the name of the OS/390 UNIX kernel address space), or (for the TN3270 Telnet Server) INTCLIEN.

- For systems at OS/390 Release 2.10 and above, a PORT or PORTRANGE statement with the RESERVED operand should be used to prevent access to specific IP ports that are not in use. This enhances the protection provided by the RESTRICTLOWPORTS operand noted below for the TCPCONFIG and UDPCONFIG statements.

- For systems at OS/390 Release 2.10 and above, a PORT or PORTRANGE statement with the SAF operand should be used to restrict access to specific IP ports used by servers that should have limited access. Please refer to *Section 4.4.1.1.6, SAF Server Access Authorization (SERVAUTH)*, for additional information.

The following guidelines are required to enhance security:

- *(ITCP0030: CAT II) The systems programmer responsible for supporting ICS will ensure that the DELETE statement is not coded in PROFILE.TCPIP files for production systems.*

- *(ITCP0070: CAT II) The IAO will ensure that write and allocate access to the data set(s) specified in the INCLUDE statements are restricted to systems programming personnel and are logged.*

- *(ITCP0030: CAT II) The systems programmer responsible for supporting ICS will ensure that the SMFCONFIG statement is coded with (at least) the FTPCLIENT and TN3270CLIENT operands.*

- *(ITCP0030: CAT II) The systems programmer responsible for supporting ICS will ensure that the SMFPARMS statement is not used.*

- *(ITCP0030: CAT II) The systems programmer responsible for supporting ICS will ensure that the TCPCONFIG and UDPCONFIG statements are coded with (at least) the RESTRICTLOWPORTS operand.*

4.4.1.1.5 AUTOLOG Subtask

The AUTOLOG subtask provides an automatic start and restart facility. It runs in the TCP/IP address space and is configured by coding the appropriate AUTOLOG and PORT or

PORTRANGE statements. Please refer to the previous section for a discussion of these statements.

Using the AUTOLOG facility to start some TCP/IP application servers has operational and security benefits. The operational benefits result from the reduced operator intervention, or from other system management software configuration tasks that would be required for starting or restarting the servers. The security benefits result from the use of established started task security as compared to potential security setups for tasks started via scripts in the OS/390 UNIX environment.

The FTP Server is an ideal candidate for the use of the AUTOLOG facility. Please refer to the chapter titled *File Transfer Protocol (FTP)* in the *IBM Communications Server for OS/390 V2R10 TCP/IP Implementation Guide Volume 2: UNIX Applications* document for examples of the configuration statements required.

Sites should consider using the AUTOLOG facility for starting TCP/IP application servers.

4.4.1.1.6 SERVAUTH

In OS/390 Release 2.10, the TCP/IP stack can invoke additional security checks for TCP/IP users. This capability is referred to as Server Access Authorization (SERVAUTH). Resources in the SERVAUTH SAF class can be used to control access to different kinds of TCP/IP resources:

Stack access refers to the ability for a user to access a specific TCP/IP stack (i.e., TCP/IP address space). This access applies to programs that invoke the socket() call.

Network access refers to the ability for a user to send IP data packets to a specific network or host. Note that this is control of outbound data.

Port access refers to the ability for an application to use a specific IP port.

FTP access refers to the ability to access the HFS for users utilizing FTP. This access applies to systems running with software support related to IBM APAR PQ63326.

The following table lists the SAF resource names and the related configuration statements that are used to implement Server Access Authorization:

Table A-47. TCP/IP SERVAUTH RESOURCE DEFINITIONS (4.4.1.1.6 a)

TCP/IP SERVAUTH RESOURCE DEFINITIONS		
RESOURCE TYPE	SAF RESOURCE NAME	RELATED TCP/IP CONFIGURATION STATEMENTS
Stack	EZB.STACKACCESS.sysname.tcpipname	TCPIPJOBNAME

TCP/IP SERVAUTH RESOURCE DEFINITIONS		
RESOURCE TYPE	SAF RESOURCE NAME	RELATED TCP/IP CONFIGURATION STATEMENTS
Network	EZB.NETACCESS.sysname.tcpipname.n_resname	TCPIPJOBNAME NETACCESS
Port	EZB.PORTACCESS.sysname.tcpipname.p_resname	TCPIPJOBNAME PORT PORTRANGE
FTP	EZB.FTP.sysname.ftpdaemonname.ACCESS.HFS	N/A

In this table the following conventions are used:

- The term *sysname* refers to the value of the MVS SYSNAME system symbol that is defined in SYS1.PARMLIB(IEASYMxx).

- The term *tcpipname* refers to the name of the TCP/IP address space. This matches the name specified in TCPIP.DATA TCPIPJOBNAME statement.

- The term *n_resname* refers to the resource name defined on the PROFILE.TCPIP NETACCESS statement.

- The term *p_resname* refers to the resource name defined on the PROFILE.TCPIP PORT or PORTRANGE statement.

- The term *ftpdaemonname* refers to the JCL procedure name used to start the FTP Server.

The following issues should be considered when using these access controls:

- If the applicable stack access resource is not defined, RACF does not perform access control and all users have access to the stack. ACF2 and TOP SECRET automatically check the stack access resource.

- When network access resources are defined and DNS servers (NSINTERADDR) are used, all users need access to the network on which the DNS servers reside.

- If the DEFAULT operand is not specified on the NETACCESS statement, only the networks explicitly listed are checked for access.

- The ID associated with the FTP Server should have access to the networks defined in NETACCESS if transfers to those networks are to be permitted. In addition, the ID associated with the task performing the transfer should also have access.

- A specific implementation of port access that also uses SERVAUTH resources is available to provide enhanced security for the TN3270 Telnet Server. Please refer to *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, for additional information on configuring access control for TN3270 Telnet ports. Please note that this support is available starting with OS/390 Release 2.8.

At this time, the following guidance applies to the use of the Server Access Authorization:

- Access to the stack, network, port, or FTP resources can be granted to all authenticated users. However, sites should seriously consider defining specific access controls where practical.

- The default access to the EZB-prefixed resources in the SERVAUTH class should be no access. This policy anticipates the new resources that are added in OS releases beyond OS/390 Release 2.10.

Please refer to the ACP-specific subsections that follow for the required access controls.

4.4.1.1.7 VMCF and TNF Subsystems

The Pascal socket TCP/IP interface is used by several IBM applications including the SMTP and LPD servers and the PING and TRACERTE commands. In turn, the Pascal socket API uses the Virtual Machine Communication Facility (VMCF) and Termination Notification Facility (TNF) subsystems for inter-address space communications. This section describes some requirements for enabling VMCF and TNF.

VMCF and TNF can be configured as restart able or non-restart able subsystems. The restart able configuration provides better error detection and operational capabilities. The information in this section assumes that a restart able configuration is being used. The EZAZSSI program is used to start the subsystems in restart able mode.

The tasks required to enable the subsystems are as follows:

- The IEFSSNxx members of SYS1.PARMLIB should be updated to identify both the VMCF and TNF subsystems.

- The EZAZSSI started task should be started during system IPL.

The statements needed to define the EZAZSSI started task to the ACP are described in the ACP-specific subsections that follow.

4.4.1.1.8 Sensitive Commands

The system operator interface to the TCP/IP address space uses the following typical MVS commands—DISPLAY (D), MODIFY (F), and VARY (V). Refer to *Section 3.1.5.6, OS/390 System Command Controls*, for guidance on securing these commands.

4.4.1.2 HFS Object Protection

In order to protect the Base TCP/IP component, special security settings should be applied to selected HFS directories and files.

- *(ITCP0040: CAT II) The systems programmer responsible for supporting ICS will ensure that the permission bits and user audit bits for HFS objects that are part of the Base TCP/IP component are configured according to the settings in the following table:*

Table A-48. BASE TCP/IP HFS OBJECT SECURITY SETTINGS (4.4.1.2 a)

BASE TCP/IP HFS OBJECT SECURITY SETTINGS			
DIRECTORY or FILE	PERMISSION BITS	USER AUDIT BITS	FUNCTION
/etc/hosts	0744	faf	Hosts name database
/etc/protocol	0744	faf	Protocols name database
/etc/resolv.conf	0740	faf	Name resolution file
/etc/services	0740	faf	Network services and aliases
/usr/lpp/tcpip/sbin	0755	faf	Daemon executable library
/usr/lpp/tcpip/bin	0755	faf	Command executable library

Some of the files listed above (e.g., /etc/resolv.conf) are not used in every configuration. While the absence of a file is generally not a security issue, the existence of a file that has not been properly secured can often be an issue. Therefore, all directories and files that do exist will have the specified permission and audit bit settings.

The following commands can be used (from a user account with an effective UID(0)) to update the permission bits and audit bits:

```
-   chmod  0744  /etc/hosts
-   chaudit  w=sf,rx+f  /etc/hosts
-   chmod  0744  /etc/protocol
-   chaudit  w=sf,rx+f  /etc/protocol
-   chmod  0740  /etc/resolv.conf
-   chaudit  w=sf,rx+f  /etc/resolv.conf
-   chmod  0740  /etc/services
-   chaudit  w=sf,rx+f  /etc/services
-   chmod  0755  /usr/lpp/tcpip/bin
-   chaudit  w=sf,rx+f  /usr/lpp/tcpip/bin
-   chmod  0755  /usr/lpp/tcpip/sbin
-   chaudit  w=sf,rx+f  /usr/lpp/tcpip/sbin
```

4.4.1.3 ACF2

This section describes the commands needed to implement the security guidelines for the Base TCP/IP System under the ACF2 ACP. The following task categories are described:

- Resource definitions
- Started task definitions
- Data set protection.

4.4.1.3.1 Resources

As of OS/390 Release 2.10, the Base TCP/IP System component defines resources in the SERVAUTH SAF class to control stack, network, port, and FTP access. These resources are discussed in *Section 4.4.1.1.6, SAF Server Access Authorization (SERVAUTH)*. As noted in that section, access to these resources needs to be controlled.

- *(ITCP0050: CAT III) The IAO will ensure that the SERVAUTH resource class is mapped to the STIG required resource type SER.*

- *(ITCP0050: CAT III) The IAO will ensure that the generic resource EZB. is defined to the SERVAUTH resource class and no access is specified.*

- *(ITCP0050: CAT III) The IAO will ensure that only authenticated users are permitted access to the network (NETACCESS), port (PORTACCESS), stack (STACKACCESS), and FTP resources in the SERVAUTH class.*

If no SERVAUTH class records were previously defined, use the following ACF commands:

```
SET CONTROL(GSO)
CHANGE INFODIR TYPES(R-RSER)
```

The following rules can be used to establish the basic security required for SERVAUTH resources:

```
$KEY(EZB) TYPE(SER)
-  UID(-) PREVENT
FTP.-  UID(-) SERVICE(READ) ALLOW
NETACCESS.-  UID(-) SERVICE(READ) ALLOW
PORTACCESS.-  UID(-) SERVICE(READ) ALLOW
STACKACCESS.-  UID(-) SERVICE(READ) ALLOW
```

The following operator commands are required to complete the update:

- If no SERVAUTH class records were previously defined:

```
F ACF2,REFRESH(INFODIR)
F ACF2,REBUILD(SER)
```

The following notes apply to these controls:

- According to Computer Associates' *OS/390 and z/OS Security Cookbook* for eTrust CA-ACF2, stack (EZB.STACKACCESS) resources are validated automatically. As a result, adequate access definitions for stack resources are critical to proper system availability.

- To be effective in restricting access, the network (EZB.NETACCESS) resource control requires configuration of the NETACCESS statement in the PROFILE.TCPIP file.

- To be effective in restricting access, the port (EZB.PORTACCESS) resource control requires configuration of a PORT or PORTRANGE statement in the PROFILE.TCPIP file.

- An additional control for TN3270 ports exists in the SERVAUTH class. Refer to *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, for a description and the ACF2-specific subsection in *Section 4.4.2, TN3270 Telnet Server*, for implementation details.

4.4.1.3.2 Started Tasks

The Base TCP/IP System component requires the definition of a started task for the TCP/IP address space and for the EZAZSSI started task.

- *(ITCP0060: CAT II) The IAO will ensure that the user account used for the TCP/IP address space is defined with the following characteristics:*

  - *Named TCPIP or, in the case of multiple instances, prefixed with TCPIP*
  - *Privilege to run as a started task*
  - *OS/390 UNIX attributes: UID(0), home directory '/', shell program /bin/sh*
  - *Access to the BPX.DAEMON resource*
  - *Access to resources in the CSFSERV class, if hardware encryption is enabled*

- *(ITCP0060: CAT II) The IAO will ensure that the user account used for the EZAZSSI started task is defined with the following characteristics:*

  - *Named EZAZSSI*
  - *Privilege to run as a started task*

The following commands can be used to create the user accounts that are required for the TCP/IP address space and the EZAZSSI started task:

```
SET LID
INSERT TCPIP NAME(TCPIP) GROUP(STCTCPX) STC MUSASS
INSERT EZAZSSI NAME(EZAZSSI) GROUP(STCTCPX) STC
SET PROFILE(USER) DIVISION(OMVS)
INSERT TCPIP UID(0) HOME(/) PROGRAM(/bin/sh)
```

NOTE: At eTrust CA-ACF2 6.4 and above, the PROGRAM field in the user profile record has been renamed to OMVSPGM.

The following additions (in bold) to the indicated rule sets can be used to assign the privileges that are required for the TCP/IP address space:

```
$KEY(BPX) TYPE(FAC)
…
DAEMON UID(TCPIP-uid) SERVICE(READ) ALLOW
…
```

If the OS/390 host machine has hardware encryption installed and enabled, resources owned by the Integrated Cryptographic Service Facility (ICSF) component have been defined. The following rule set additions are required to allow the TN3270 Telnet Server process to access the ICSF resources. Please refer to *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, for information on how these resources are used.

- $KEY(CSFCKI) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFCKM) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFDEC) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFENC) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFOWH) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFRNG) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFPKB) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFPKX) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFPKE) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFPKD) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFPKI) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFDSG) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW
- $KEY(CSFDSV) TYPE(CSF)
- UID(*TCPIP-uid*) SERVICE(READ) ALLOW

The following operator commands are required to complete the updates:

```
F ACF2,REBUILD(FAC)
F ACF2,REBUILD(CSF)
```

These commands and definitions assume that the default type code for CSFSERV resources is CSF.

4.4.1.3.3 Data Sets

There are three groups of data sets associated with the Base TCP/IP System that require protection:

- Product distribution and target data sets that are supplied by the vendor
- Local configuration file data sets for the TCP/IP stack
- Local configuration file data sets that are shared by TCP/IP applications

The product data sets for the Base TCP/IP System are packaged as follows:

- Distribution data sets hold the master copy of the product elements. There is no typical need for general users to access these data sets. The standard naming convention for these data sets is to use the prefix SYS1.TCPIP.AEZA.

- Target data sets hold the execution copy of the product elements. General users are likely to need *read* access to some of these data sets. The standard naming convention for these data sets is to use the prefix SYS1.TCPIP.SEZA.

- *(ITCP0070: CAT II) The IAO will ensure that write and allocate access to product data sets are restricted to systems programming personnel.*

The local configuration data sets for the TCP/IP stack consist of the TCPIP.DATA and PROFILE.TCPIP files. These files contain critical operating parameters, and the ability to change or delete these files will be controlled. Users of TCP/IP applications are likely to need *read* access to these files. The standard convention for these files is to use members in a partitioned data set named SYS1.TCPPARMS.

- *(ITCP0070: CAT II) The IAO will ensure that write and allocate access to the data set(s) containing the TCPIP.DATA and PROFILE.TCPIP configuration files are restricted to systems programming personnel and are logged.*

The local configuration data sets shared by TCP/IP applications may consist of several files. Please refer to *Section 4.4.1.1.2, Configuration Files – Shared by TCP/IP Applications*, for details. As noted there, the names of these data sets are affected by the TCPIP.DATA DATASETPREFIX parameter. Users of TCP/IP applications are likely to need *read* access to these files. The standard naming convention for these data sets is to use the prefix SYS3.TCPIP.

- *(ITCP0070: CAT II) The IAO will ensure that write and allocate access to the data set(s) containing the configuration files shared by TCP/IP applications are restricted to systems programming personnel.*

The following additions (in bold) to the SYS1 and SYS3 rule sets can be used as a base to secure the MVS data sets:

```
$KEY(SYS1)
…
TCPIP.AEZA- UID(sysprog-UID) READ(A) WRITE(A) ALLOC(A) EXEC(A)
TCPIP.SEZA- UID(sysprog-UID) READ(A) WRITE(A) ALLOC(A) EXEC(A)
TCPPARMS UID(sysprog-UID) READ(A) WRITE(L) ALLOC(L) EXEC(A)
TCPPARMS UID(-) READ(A)
…

$KEY(SYS3)
…
TCPIP.ETC- UID(sysprog-UID) READ(A) WRITE(L) ALLOC(L) EXEC(A)
TCPIP.ETC- UID(-) READ(A)
TCPIP.HOSTS- UID(sysprog-UID) READ(A) WRITE(L) ALLOC(L) EXEC(A)
TCPIP.HOSTS- UID(-) READ(A)
TCPIP.STANDARD- UID(sysprog-UID) READ(A) WRITE(L) ALLOC(L) EXEC(A)
TCPIP.STANDARD- UID(-) READ(A)
…
```

4.4.1.4 RACF

This section describes the commands needed to implement the security guidelines for the Base TCP/IP System under the RACF ACP. The following task categories are described:

- Resource definitions
- Started task definitions
- Data set protection.

4.4.1.4.1 Resources

As of OS/390 Release 2.10, the Base TCP/IP System component defines resources in the SERVAUTH SAF class to control stack, network, and port access. These resources are discussed in *Section 4.4.1.1.6, SAF Server Access Authorization (SERVAUTH)*. As noted in that section, access to these resources needs to be controlled.

- *(ITCP0050: CAT III) The IAO will ensure that the default access to EZB-prefixed resources in the SERVAUTH class is no access.*

- *(ITCP0050: CAT III) The IAO will ensure that the generic resource EZB. is defined to the SERVAUTH resource classand no access is granted.*

- *(ITCP0050: CAT III) The IAO will ensure that only authenticated users are permitted access to the network (NETACCESS), port (PORTACCESS), stack (STACKACCESS), and FTP resources in the SERVAUTH class.*

The following commands can be used to establish the basic security required for SERVAUTH resources:

```
If SERVAUTH was not previously defined, SETROPTS CLASSACT(SERVAUTH).
If SERVAUTH was not previously defined, SETROPTS RACLIST(SERVAUTH).
RDEFINE SERVAUTH EZB.** UACC(NONE) OWNER(ADMIN)
RDEFINE SERVAUTH EZB.FTP.** UACC(NONE) OWNER(ADMIN)
PERMIT EZB.FTP.** CLASS(SERVAUTH) ACCESS(READ) ID(*)
RDEFINE SERVAUTH EZB.NETACCESS.** UACC(NONE) OWNER(ADMIN)
PERMIT EZB.NETACCESS.** CLASS(SERVAUTH) ACCESS(READ) ID(*)
RDEFINE SERVAUTH EZB.PORTACCESS.** UACC(NONE) OWNER(ADMIN)
PERMIT EZB.PORTACCESS.** CLASS(SERVAUTH) ACCESS(READ) ID(*)
RDEFINE SERVAUTH EZB.STACKACCESS.** UACC(NONE) OWNER(ADMIN)
PERMIT EZB.STACKACCESS.** CLASS(SERVAUTH) ACCESS(READ) ID(*)
SETROPTS RACLIST(SERVAUTH) REFRESH
```

The following notes apply to these controls:

- To be effective in restricting access, the network (EZB.NETACCESS) resource control requires configuration of the NETACCESS statement in the PROFILE.TCPIP file.

- To be effective in restricting access, the port (EZB.PORTACCESS) resource control requires configuration of a PORT or PORTRANGE statement in the PROFILE.TCPIP file.

- An additional control for TN3270 ports exists in the SERVAUTH class. Refer to *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections,* for a description and the RACF-specific subsection in *Section 4.4.2, TN3270 Telnet Server,* for implementation details.

4.4.1.4.2 Started Tasks

The Base TCP/IP System component requires the definition of a started task for the TCP/IP address space and for the EZAZSSI started task.

- *(ITCP0060: CAT II) The IAO will ensure that the user account used for the TCP/IP address space are defined with the following characteristics:*

  - Named TCPIP or, in the case of multiple instances, prefixed with TCPIP
  - Privilege to run as a started task
  - OS/390 UNIX attributes: UID(0), home directory '/', shell program /bin/sh
  - Access to the BPX.DAEMON resource

- Access to resources in the CSFSERV class, if hardware encryption is enabled.

- *(ITCP0060: CAT II) The IAO will ensure that the user account used for the EZAZSSI started task are defined with the following characteristics:*

  - *Named EZAZSSI*
  - *Privilege to run as a started task*

The following commands can be used to create the user accounts and assign the privileges that are required for the TCP/IP address space and the EZAZSSI started task:

```
ADDUSER TCPIP DFLTGRP(STCTCPX) OWNER(ADMIN) -
    NOPASSWORD NOOIDCARD -
    OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
RDEFINE STARTED TCPIP.* UACC(NONE) OWNER(ADMIN) -
    STDATA(USER(TCPIP) GROUP(STCTCPX) TRUSTED(NO))
PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(TCPIP)
```

If the OS/390 host machine has hardware encryption installed and enabled, resources owned by the ICSF component have been defined. The following PERMIT commands are required to allow the TN3270 Telnet Server process to access the ICSF resources. Please refer to *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, for information on how these resources are used.

```
PERMIT CSFCKI  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFCKM  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFDEC  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFENC  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFOWH  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFRNG  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFPKB  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFPKX  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFPKE  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFPKD  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFPKI  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFDSG  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
PERMIT CSFDSV  CLASS(CSFSERV) ACCESS(READ) ID(TCPIP)
```

```
ADDUSER EZAZSSI DFLTGRP(STCTCPX) OWNER(ADMIN) -
    NOPASSWORD NOOIDCARD
RDEFINE STARTED EZAZSSI.* UACC(NONE) OWNER(ADMIN) -
    STDATA(USER(EZAZSSI) GROUP(STCTCPX) TRUSTED(NO))
```

4.4.1.4.3 Data Sets

There are three groups of data sets associated with the Base TCP/IP System that require protection:

- Product distribution and target data sets that are supplied by the vendor
- Local configuration file data sets for the TCP/IP stack
- Local configuration file data sets that are shared by TCP/IP applications

The product data sets for the Base TCP/IP System are packaged as follows:

Distribution data sets hold the master copy of the product elements. There is no typical need for general users to access these data sets. The standard naming convention for these data sets is to use the prefix SYS1.TCPIP.AEZA.

Target data sets hold the execution copy of the product elements. General users are likely to need *read* access to some of these data sets. The standard naming convention for these data sets is to use the prefix SYS1.TCPIP.SEZA.

- *(ITCP0070: CAT II) The IAO will ensure that update and alter access to product data sets are restricted to systems programming personnel.*

The local configuration data sets for the TCP/IP stack consist of the TCPIP.DATA and PROFILE.TCPIP files. These files contain critical operating parameters and the ability to change or delete these files will be controlled. Users of TCP/IP applications are likely to need *read* access to these files. The standard convention for these files is to use members in a partitioned data set named SYS1.TCPPARMS.

- *(ITCP0070: CAT II) The IAO will ensure that update and alter access to the data set(s) containing the TCPIP.DATA and PROFILE.TCPIP configuration files are restricted to systems programming personnel and are logged.*

The local configuration data sets shared by TCP/IP applications may consist of several files. Please refer to *Section 4.4.1.1.2, Configuration Files – Shared by TCP/IP Applications,* for details. As noted there, the names of these data sets are affected by the TCPIP.DATA DATASETPREFIX parameter. Users of TCP/IP applications are likely to need *read* access to these files. The standard naming convention for these data sets is to use the prefix SYS3.TCPIP.

- *(ITCP0070: CAT II) The IAO will ensure that update and alter access to the data set(s) containing the configuration files shared by TCP/IP applications are restricted to systems programming personnel.*

The following commands can be used to provide the required access control for the MVS data sets:

```
ADDSD ‘SYS1.TCPIP.AEZA*’ OWNER(SYS1) UACC(NONE)
PERMIT ‘SYS1.TCPIP.AEZA*’ ACCESS(ALTER) ID(sysprog-group)
ADDSD ‘SYS1.TCPIP.SEZA*’ OWNER(SYS1) UACC(NONE)
PERMIT ‘SYS1.TCPIP.SEZA*’ ACCESS(ALTER) ID(sysprog-group)

ADDSD ‘SYS1.TCPPARMS’ OWNER(SYS1) UACC(NONE) -
    AUDIT(ALL(UPDATE))
PERMIT ‘SYS1.TCPPARMS’ ACCESS(ALTER) ID(sysprog-group)
PERMIT ‘SYS1.TCPPARMS’ ACCESS(READ) ID(*)

ADDSD ‘SYS3.TCPIP.ETC*’ OWNER(SYS1) UACC(NONE)
PERMIT ‘SYS3.TCPIP.ETC*’ ACCESS(ALTER) ID(sysprog-group)
PERMIT ‘SYS3.TCPIP.ETC*’ ACCESS(READ) ID(*)
ADDSD ‘SYS3.TCPIP.HOSTS*’ OWNER(SYS1) UACC(NONE)
PERMIT ‘SYS3.TCPIP.HOSTS*’ ACCESS(ALTER) ID(sysprog-group)
PERMIT ‘SYS3.TCPIP.HOSTS*’ ACCESS(READ) ID(*)
ADDSD ‘SYS3.TCPIP.STANDARD*’ OWNER(SYS1) UACC(NONE)
PERMIT ‘SYS3.TCPIP.STANDARD*’ ACCESS(ALTER) ID(sysprog-group)
PERMIT ‘SYS3.TCPIP.STANDARD*’ ACCESS(READ) ID(*)
```

4.4.1.5 TOP SECRET

This section describes the commands needed to implement the security guidelines for the Base TCP/IP System under the TOP SECRET ACP. The following task categories are described:

- Resource definitions
- Started task definitions
- Data set protection.

4.4.1.5.1 Resources

As of OS/390 Release 2.10, the Base TCP/IP System component defines resources in the SERVAUTH SAF class to control stack, network, and port access. These resources are discussed in *Section 4.4.1.1.6, SAF Server Access Authorization (SERVAUTH)*. As noted in that section, access to these resources needs to be controlled.

- *(ITCP0050: CAT III) The IAO will ensure that the default access to EZB-prefixed resources in the SERVAUTH class is no access.*

- *(ITCP0050: CAT III) The IAO will ensure that the generic resource EZB. is defined to the SERVAUTH resource classand no access is granted.*

- *(ITCP0050: CAT III) The IAO will ensure that only authenticated users are permitted access to the network (NETACCESS), port (PORTACCESS), stack (STACKACCESS), and FTP resources in the SERVAUTH class.*

The following commands can be used to establish the basic security required for SERVAUTH resources:

```
If SERVAUTH was not previously defined, TSS ADD(ADMIN)  SERVAUTH(EZB).
TSS PERMIT(ALL) SERVAUTH(EZB.FTP.) ACCESS(READ)
TSS PERMIT(ALL) SERVAUTH(EZB.NETACCESS.) ACCESS(READ)
TSS PERMIT(ALL) SERVAUTH(EZB.PORTACCESS.) ACCESS(READ)
TSS PERMIT(ALL) SERVAUTH(EZB.STACKACCESS.) ACCESS(READ)
```

The following notes apply to these controls:

- According to Computer Associates' *Security Cookbook* for eTrust CA-TOP SECRET, access to stack (EZB.STACKACCESS) resources will be given to ACIDs that require it. As a result, adequate access definitions for stack resources are critical to proper system availability.

- To be effective in restricting access, the network (EZB.NETACCESS) resource control requires configuration of the NETACCESS statement in the PROFILE.TCPIP file.

- To be effective in restricting access, the port (EZB.PORTACCESS) resource control requires configuration of a PORT or PORTRANGE statement in the PROFILE.TCPIP file.

- An additional control for TN3270 ports exists in the SERVAUTH class. Refer to *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, for a description and the TOP SECRET-specific subsection in *Section 4.4.2, TN3270 Telnet Server*, for implementation details.

4.4.1.5.2 Started Tasks

The Base TCP/IP System component requires the definition of a started task for the TCP/IP address space and for the EZAZSSI started task.

- *(ITCP0060: CAT II) The IAO will ensure that the user account used for the TCP/IP address space is defined with the following characteristics:*

  - Named TCPIP or, in the case of multiple instances, prefixed with TCPIP
  - Privilege to run as a started task
  - OS/390 UNIX attributes: UID(0), home directory '/', shell program /bin/sh
  - Access to the BPX.DAEMON resource
  - Access to resources in the CSFSERV class, if hardware encryption is enabled.

- *(ITCP0060: CAT II) The IAO will ensure that the user account used for the EZAZSSI started task is defined with the following characteristics:*

  - Named EZAZSSI
  - Privilege to run as a started task

The following commands can be used to create the user accounts and assign the privileges that are required for the TCP/IP address space and the EZAZSSI started task:

```
TSS CREATE(TCPIP) TYPE(USER) NAME(TCPIP)
    DEPT(existing-dept) FACILITY(STC) PASSWORD(password,0)
 TSS ADD(TCPIP) DFLTGRP(STCTCPX) GROUP(STCTCPX)
 TSS ADD(TCPIP) SOURCE(INTRDR)
 TSS ADD(TCPIP) UID(0) HOME(/) OMVSPGM(/bin/sh)
 TSS ADD(TCPIP) MASTFAC(TCP)
 TSS ADD(STC) PROCNAME(TCPIP) ACID(TCPIP)
 TSS PERMIT(TCPIP) IBMFAC(BPX.DAEMON) ACCESS(READ)
```

If the OS/390 host machine has hardware encryption installed and enabled, resources owned by the ICSF component have been defined. The following PERMIT commands are required to allow the TN3270 Telnet Server process to access the ICSF resources. Please refer to *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, for information on how these resources are used.

```
TSS PERMIT(TCPIP)  CSFSERV(CSFCKI)  ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFCKM) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFDEC) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFENC) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFOWH) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFRNG) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFPKB) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFPKX) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFPKE) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFPKD) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFPKI) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFDSG) ACCESS(READ)
TSS PERMIT(TCPIP)  CSFSERV(CSFDSV) ACCESS(READ)
```

```
TSS CREATE(EZAZSSI) TYPE(USER) NAME(EZAZSSI)
    DEPT(existing-dept) FACILITY(STC) PASSWORD(password,0)
TSS ADD(EZAZSSI) DFLTGRP(STCTCPX) GROUP(STCTCPX)
TSS ADD(EZAZSSI) SOURCE(INTRDR)
TSS ADD(EZAZSSI) UID(0) HOME(/) OMVSPGM(/bin/sh)
TSS ADD(EZAZSSI) MASTFAC(TCP)
TSS ADD(STC) PROCNAME(EZAZSSI) ACID(EZAZSSI)
```

4.4.1.5.3 Data Sets

There are three groups of data sets associated with the Base TCP/IP System that require protection:

- Product distribution and target data sets that are supplied by the vendor
- Local configuration file data sets for the TCP/IP stack
- Local configuration file data sets that are shared by TCP/IP applications

The product data sets for the Base TCP/IP System are packaged as follows:

Distribution data sets hold the master copy of the product elements. There is no typical need for general users to access these data sets. The standard naming convention for these data sets is to use the prefix SYS1.TCPIP.AEZA.

Target data sets hold the execution copy of the product elements. General users are likely to need *read* access to some of these data sets. The standard naming convention for these data sets is to use the prefix SYS1.TCPIP.SEZA.

- *(ITCP0070: CAT II) The IAO will ensure that update, create, and scratch access to product data sets are restricted to systems programming personnel.*

The local configuration data sets for the TCP/IP stack consist of the TCPIP.DATA and PROFILE.TCPIP files. These files contain critical operating parameters and the ability to change or delete these files will be controlled. Users of TCP/IP applications are likely to need *read* access to these files. The standard convention for these files is to use members in a partitioned data set named SYS1.TCPPARMS.

- *(ITCP0070: CAT II) The IAO will ensure that update, create, and scratch access to the data set(s) containing the TCPIP.DATA and PROFILE.TCPIP configuration files are restricted to systems programming personnel and are logged.*

The local configuration data sets shared by TCP/IP applications may consist of several files. Please refer to *Section 4.4.1.1.2, Configuration Files – Shared by TCP/IP Applications,* for details. As noted there, the names of these data sets are affected by the TCPIP.DATA DATASETPREFIX parameter. Users of TCP/IP applications are likely to need *read* access to these files. The standard naming convention for these data sets is to use the prefix SYS3.TCPIP.

- *(ITCP0070: CAT II) The IAO will ensure that update, create, and scratch access to the data set(s) containing the configuration files shared by TCP/IP applications are restricted to systems programming personnel.*

The following commands can be used to provide the required access control for the MVS data sets:

```
TSS ADD(SYS1) DSN(SYS1.TCPIP.AEZA-)
TSS PERMIT(sysprog-group) DSN(SYS1.TCPIP.AEZA-) ACCESS(ALL)
TSS ADD(SYS1) DSN(SYS1.TCPIP.SEZA-)
TSS PERMIT(sysprog-group) DSN(SYS1.TCPIP.SEZA-) ACCESS(ALL)

TSS ADD(SYS1) DSN(SYS1.TCPPARMS)
TSS PERMIT(sysprog-group) DSN(SYS1.TCPPARMS) ACCESS(ALL)
    ACTION(AUDIT)
TSS PERMIT(ALL) DSN(SYS1.TCPPARMS) ACCESS(READ)

TSS ADD(SYS1) DSN(SYS3.TCPIP.ETC.-)
TSS PERMIT(sysprog-group) DSN(SYS3.TCPIP.ETC.-) ACCESS(ALL)
TSS PERMIT(ALL) DSN(SYS3.TCPIP.ETC.-) ACCESS(READ)
TSS ADD(SYS1) DSN(SYS3.TCPIP.HOSTS.-)
TSS PERMIT(sysprog-group) DSN(SYS3.TCPIP.HOSTS.-) ACCESS(ALL)
TSS PERMIT(ALL) DSN(SYS3.TCPIP.HOSTS.-) ACCESS(READ)
TSS ADD(SYS1) DSN(SYS3.TCPIP.STANDARD.-)
TSS PERMIT(sysprog-group) DSN(SYS3.TCPIP.STANDARD.-) ACCESS(ALL)
TSS PERMIT(ALL) DSN(SYS3.TCPIP.STANDARD.-) ACCESS(READ)
```

4.4.2 TN3270 Telnet Server

The TN3270 Telnet Server is a closely integrated component of IBM’s Communications Server. It provides the server portion of the client/server application that allows interactive terminal access from clients on TCP/IP networks to applications on an OS/390 host.

The TN3270 Telnet Server runs in the OS/390 TCP/IP address space as part of the Base TCP/IP component. It functions as both a TCP/IP application and a VTAM application. As a TCP/IP application it listens on one or more IP ports for connection requests from clients and sends host data back to the clients. As a VTAM application, it sends data to and receives data from SNA applications. It is the TN3270 Telnet Server that allows IP-based TN3270 clients to access SNA applications using the TN3270, TN3270 Enhanced (TN3270E), or linemode protocols.

As it executes in the TCP/IP address space, the TN3270 Telnet Server shares some security resources with the Base TCP/IP component. These resources include the started task definition and a configuration file. As a result, this section of the document supplements *Section 4.4.1, Base TCP/IP System*, and has to be used together with that section. Those requirements that are specific to the TN3270 Telnet Server are covered in this section.

It should be noted that the information in this section has been prepared to address the configuration requirements as of Version 2, Releases 8 and 10 of OS/390. IBM made significant changes to the software between these releases and parameter differences reflect these changes. Please refer to the *OS/390 IBM Communications Server IP Configuration Guide* document (OS/390 Release 2.10) or the *OS/390 SecureWay Communications Server IP Configuration* document (OS/390 Release 2.8) for documentation of specific syntax.

4.4.2.1 General Considerations

The configuration issues addressed relative to the security environment for the TN3270 Telnet Server include the following:

- Configuration statements in the PROFILE.TCPIP file
- Controlling session setup
- Banner message requirements
- Secure Sockets Layer (SSL) connections
- SMF recording

4.4.2.1.1 PROFILE.TCPIP Configuration Statements

The TN3270 Telnet Server uses configuration statements in the file referred to as the PROFILE.TCPIP file. The PROFILE.TCPIP file is used primarily by the Base TCPIP component. The security requirements for the file itself are addressed in *Section 4.4.1, Base TCP/IP System*.

Because system security is impacted by some of the parameters in the PROFILE.TCPIP file, and the default settings do not provide an adequate level of security, certain parameter settings should be explicitly specified. Those required parameter settings are discussed in this section.

The TN3270 Telnet Server is capable of listening to up to 255 IP ports. The Internet Assigned Numbers Authority (IANA) defines two ports for Telnet in the range of well known ports. Port 23 is assigned to common Telnet connections; port 992 is assigned to Telnet protocol over TLS/SSL. Different ports are used as a simple mechanism to allow connections with different operating characteristics to coexist. Accordingly, the TN3270 Telnet Server configuration statements are organized into statement blocks to allow different ports to be assigned different parameter values.

The configuration statements for the TN3270 Telnet Server are expressed in two statement blocks in OS/390 Release 2.8 and three blocks in Release 2.10:

- TELNETGLOBALS (OS/390 Release 2.10) – Includes parameters that apply to all ports.
- TELNETPARMS – Includes parameters that define the characteristics of one port.
- BEGINVTAM – Includes parameters that define characteristics related to VTAM, such as Logical Unit (LU) names and application relationships.

While there may only be one TELNETGLOBALS block (OS/390 Release 2.10) in the PROFILE.TCPIP file, the number of TELNETPARMS blocks corresponds directly to the number of ports to which the server is listening. BEGINVTAM blocks can correspond to one or more TELNETPARMS blocks, depending on the parameters to be applied. For example, if the site intends for the server to listen to ports 23 and 992, there is one TELNETGLOBALS block (OS/390 Release 2.10), two TELNETPARMS blocks, and one or two BEGINVTAM blocks. The requirements in this section are documented to correspond to this structure.

TELNETGLOBALS Statements

The TELNETGLOBALS block is used in OS/390 Release 2.10 to supply parameters that apply to all TELNETPARMS blocks. The KEYRING statement can be coded once in the TELNETGLOBALS block rather than multiple times in TELNETPARMS blocks. KEYRING specifies the source for digital certificates required for TN3270 SSL processing. Since all ports that are supporting SSL processing should use the same KEYRING file, it reduces ambiguity to have it coded in one place. The TN3270 Telnet Server in OS/390 Release 2.10 allows the use of an MVS data set, an HFS file, or the resident ACP as the source for digital certificates. Use of the ACP is consistent with the security philosophy maintained for OS/390 systems.

- *(ITNT0010: CAT II) The systems programmer responsible for supporting ICS will ensure that for systems at OS/390 Release 2.10 and above, the KEYRING statement, if used, is only coded within the TELNETGLOBALS statement block and specifies the SAF parameter, indicating that the resident ACP manages the digital certificates being used.*

TELNETPARMS Statements

Each TELNETPARMS block specifies parameters for a specific IP port. Several of these parameters have potential impacts to system security and therefore require specific settings.

PORT and SECUREPORT are mutually exclusive within a TELNETPARMS block. A PORT statement defines an IP port for basic sessions. For systems at OS/390 Release 2.8, the SECUREPORT statement defines an IP port for sessions that use the SSL protocol. For systems at OS/390 Release 2.10, the SECUREPORT statement defines an IP port for sessions that may use the SSL protocol, subject to additional configuration options. For additional information on SSL sessions, please refer to *Section 4.4.2.1.4 Secure Sockets Layer (SSL) Connections*.

For systems at OS/390 Release 2.8, the SECUREPORT statement includes the KEYRING operand, and it specifies the MVS data set or HFS file that is the source for digital certificates required for TN3270 SSL processing. Using an MVS data set provides enhanced security compared to an HFS file.

- *(ITNT0010: CAT II) The systems programmer responsible for supporting ICS will ensure that for systems at OS/390 Release 2.8, the KEYRING operand on any SECUREPORT statement, if used, specifies an MVS data set as the source for digital certificates.*

Defining a SECUREPORT can provide the advantages of an additional authentication mechanism and session encryption at the expense of acquiring and maintaining the required digital certificate. The following recommendations apply:

At least one TELNETPARMS block with a PORT statement and one TELNETPARMS block with a SECUREPORT statement should be defined.

Port number 23 should be specified on the PORT statement and port number 992 should be specified on the SECUREPORT statement.

Sites with systems at OS/390 Release 2.10 could define a single port, typically 23, that supports multiple types of Telnet sessions. This allows one port to support basic, SSL sessions, and negotiated SSL sessions. Sites are cautioned to validate that users' TN3270 clients are compatible with the specified kind of configuration. Please refer to *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections,* for additional information about SECUREPORT options.

The TELNETPARMS INACTIVE statement defines the terminal inactivity timeout value. When there has been no client-VTAM activity for the specified number of seconds, the session will be dropped. Note that the value of the INACTIVE parameter can impact the values of the PRTINACTIVE and KEEPINACTIVE (OS/390 Release 2.10) statements. The *STIG requirement* recommends that user sessions be terminated or locked out after 15 minutes of inactivity. Documentation must be maintained with the IAM when this guideline is not followed.

- *(ITNT0010: CAT II) The systems programmer responsible for supporting ICS will ensure that unless documented with the IAM, a TELNETPARMS INACTIVE statement is coded within each TELNETPARMS statement block and specifies a value between 1 and 900. Exceptions are documented with the IAO.*

The TELNETPARMS SMFINIT and SMFTERM statements control SMF recording for the TN3270 Telnet Server. Please refer to *Section 4.4.2.1.5, SMF Recording*, for the required settings.

For OS/390 Release 2.10 systems, the TELNETPARMS TKOSPECLURECON statement can be used to specify that an existing, inactive session with a specific (VTAM) Logical Unit (SPECLU) name can be taken over by a client specifying that name. This capability is intended to allow existing sessions to be recovered when a network connection fails. However, the last screen displayed in the original session is re-sent without any authentication of the client. In some circumstances, this could allow one user to connect to another user's session.

- *(ITNT0010: CAT II) The systems programmer responsible for supporting ICS will ensure that the TELNETPARMS TKOSPECLURECON statement is not coded in any TELNETPARMS statement block.*

The TELNETPARMS CLIENTAUTH, CONNTYPE (OS/390 Release 2.10), ENCRYPTION, and SSLTIMEOUT statements are used to control session processing on SECUREPORT connections. Please refer to *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, for the required settings.

BEGINVTAM Statements

Each BEGINVTAM statement block specifies VTAM session and application parameters for one or more IP ports. Several of these parameters have potential impacts to system security and therefore require specific settings.

The BEGINVTAM ALLOWAPPL and RESTRICTAPPL statements are intended to provide a level of control for access from Telnet clients to VTAM applications. The statements can also include some operating parameters that may be applicable to certain applications. Application access control is required for unsecured terminals and is handled by session manager software. Please refer to *Section 4.1.3, Security Recommendations for VTAM Networks*, and *Section 6, Session Managers*, for recommendations for application access control. The BEGINVTAM ALLOWAPPL statement may be coded to provide operating parameters. It should not be coded with the LU or LUG (OS/390 Release 2.10) operand for the purpose of access control. Since operating parameters can be coded on ALLOWAPPL instead of RESTRICTAPPL, there is no environment where RESTRICTAPPL would be applicable.

- *(ITNT0010: CAT II) The systems programmer responsible for supporting ICS will ensure that the BEGINVTAM RESTRICTAPPL statement is not coded in any BEGINVTAM statement block.*

Some special considerations apply to the following BEGINVTAM statements:

DEFAULTAPPL
HNGROUP
INTERPTCP
IPGROUP
LINEMODEAPPL
LUMAP
PARMSMAP (OS/390 Release 2.10)
PRTMAP
USSTCP

These statements can include operands that specify client source IP addresses or host names.

For IP addresses, consideration should be given to two potential issues—dynamic client IP addresses and address spoofing. The use of Dynamic Host Configuration Protocol (DHCP) or proxy firewalls can result in the same client host using different source IP addresses on successive connections. It can also result in two different client hosts using the same source IP address at different times. IP address spoofing, the unauthorized use of a legitimate IP address, results in the incorrect identification of a client. The impact of these issues is that a client IP address may not be an accurate indicator of the identity of the client.

For host names, consideration should be given to the fact that using host names requires that the TN3270 Telnet Server resolve the name to an IP address. Whether name resolution is done via files or a name server, the data should be kept accurate and its integrity should be assured.

As a result of these considerations, BEGINVTAM statement operands containing IP addresses or host names should be used only when the sources for that information are considered to be trusted.

The BEGINVTAM MSG07 statement provides information to the client when a session attempt fails. When MSG07 is not used, the connection is dropped without providing an error message. Although this parameter is not directly related to security, it can provide helpful diagnostic information in cases where sessions fail due to the values of security-related parameters. In addition, some Telnet client programs may experience auto-reconnect loops when a connection is dropped without an error message. For most OS/390 images, the BEGINVTAM MSG07 statement should be coded.

The BEGINVTAM DEFAULTAPPL, LINEMODEAPPL, LUMAP with DEFAPPL operand (OS/390 Release 2.10), and USSTCP statements can have security implications during the session setup process. Please refer to *Section 4.4.2.1.2, Session Setup Control*, for the required settings.

4.4.2.1.2 Session Setup Control

After a connection from a Telnet client to the TN3270 Telnet Server has been established, the process of session setup with a VTAM application occurs. A number of BEGINVTAM statements will be coded in a specific configuration to ensure that adequate control over access to VTAM applications is maintained.

In *Section 4.1.3, Security Recommendations for VTAM Networks*, connections are described as originating from secure terminals or unsecured terminals. The TN3270 Telnet Server should be configured to address these two types of connections. Terminals should meet two conditions to be considered secure. One condition involves the hardware and configuration. Secure terminals include devices that are directly attached to the host, such as 3270-type terminals coax connected to a 3174 Control Unit. They also include PCs running 3270 terminal emulation clients attached to a private LAN (i.e., a LAN without access to an external network such as the NIPRNet). The other condition involves the location of the terminals. Secure terminals are located in areas with physical access limited to authorized personnel. Examples of terminals that are not secure are those attached via the NIPRNet or via dial-in servers. The intent of this distinction is to allow additional connection options (e.g., bypassing session manager control) to authorized personnel working in controlled access areas. These connection options may be necessary for operational control or for system recovery procedures.

The BEGINVTAM USSTCP statement can be used to specify a customized Unformatted System Services (USS) table for client connections. The USS table can provide a level of access control by restricting the commands that allow connections to VTAM applications. The USS table specified by the USSTCP statement can be the same as the one used by the SNA component of IBM Communications Server.

- *(ITNT0020: CAT II) The systems programmer responsible for supporting ICS will ensure that within each BEGINVTAM statement block, one BEGINVTAM USSTCP statement is coded that specifies only the table name operand. The named table allows access only to* *session manager applications and NC-PASS applications. This USSTCP statement does not specify any type of client identifier, such as host name or IP address, so that the statement applies to all connections not otherwise controlled.*

- *(ITNT0020: CAT II) The systems programmer responsible for supporting ICS will ensure that within each BEGINVTAM statement block, additional BEGINVTAM USSTCP statements that specify a USS table that allows access to other applications are coded only if the statements include a client identifier operand that references only secure terminals.*

The BEGINVTAM DEFAULTAPPL statement can be used to specify the VTAM application to which a client is automatically connected when a session is established using a protocol other than linemode protocol.

- *(ITNT0020: CAT II) The systems programmer responsible for supporting ICS will ensure that any BEGINVTAM DEFAUTLAPPL statement that does not specify a client identifier, or specifies any type of client identifier that would apply to unsecured terminals, specifies a session manager application or an NC-PASS application as the application name.*

The BEGINVTAM LINEMODEAPPL statement can be used to specify the VTAM application to which a client is automatically connected when a session is established using the linemode protocol. Because USSTCP processing does not apply to clients using the linemode protocol, the LINEMODEAPPL statement is used for application access control.

- *(N/A: CAT II) The systems programmer responsible for supporting ICS will ensure that within each BEGINVTAM statement block, one BEGINVTAM LINEMODEAPPL statement is coded that specifies only the application name and, for OS/390 Release 2.10, DEFONLY operands. The named application specifies a session manager application or an NC-PASS application. This LINEMODEAPPL statement does not specify any type of client identifier, such as host name or IP address, so that the statement applies to all linemode connections not otherwise controlled.*

- *(N/A: CAT II) The systems programmer responsible for supporting ICS will ensure that any BEGINVTAM LINEMODEAPPL statement that specifies any type of client identifier that would apply to unsecured terminals specifies a session manager application or an NC-PASS application as the application name.*

For OS/390 Release 2.10 systems, the BEGINVTAM LUMAP statement can specify a default VTAM application using the DEFAPPL operand. This processing is similar to the DEFAULTAPPL and LINEMODEAPPL processing, except that a client identifier should be coded. When a client matches the LUMAP specification, the DEFAPPL specification overrides the DEFAULTAPPL or LINEMODEAPPL specifications.

- *(ITNT0020: CAT II) The systems programmer responsible for supporting ICS will ensure that for systems at OS/390 Release 2.10 and above, any BEGINVTAM LUMAP statement, if used with the DEFAPPL operand and applied to unsecured terminals, specifies only a session manager application or an NC-PASS application.*

4.4.2.1.3 Warning Banner

DOD requires that a logon warning banner be displayed. Within the TN3270 Telnet Server, the banner can be implemented through the USS table that is specified on a BEGINVTAM USSTCP statement. The text associated with message ID 10 (i.e., MSG10) in the USS table is sent to clients that are subject to USSTCP processing.

- *(ITNT0030: CAT II) The systems programmer responsible for supporting ICS will ensure that all USS tables referenced in BEGINVTAM USSTCP statements includes MSG10 text that specifies a warning logon banner.*

4.4.2.1.4 SSL Connections

The TN3270 Telnet Server is capable of using the Secure Sockets Layer (SSL) protocol in sessions with compatible TN3270 clients. The use of SSL can provide server authentication, data integrity, and, optionally, client authentication and data encryption.

For this discussion, references to the SSL protocol should be considered applicable to the Transport Layer Security (TLS) protocol unless otherwise noted. Version 1 of the TLS protocol succeeded, but was not very different from, Version 3 of SSL.

Four general areas of consideration are discussed in this section:

- SSL Connection Options – For OS/390 Release 2.10 systems, a SECUREPORT port can be configured to allow basic, SSL, and negotiated SSL connections over the same port.

- Authentication – Server authentication is assumed in SSL, but client authentication is optional. In addition, various levels of client authentication can be selected.

- Certificate Management – Server certificates, Certificate Authority (CA) certificates, and, optionally, user certificates have to be managed.

- Encryption – Different strengths or no encryption are configuration options.

SSL Connection Options

- Some SSL connection options can have an important impact to security. The first of these applies to systems at OS/390 Release 2.10 and is associated with the use of the TELNETPARMS CONNTYPE and BEGINVTAM PARMSGROUP statements to alter the behavior of SECUREPORT ports. The second option controls a timeout value that applies during SSL connections.

- Systems at OS/390 Release 2.10 can be configured to allow a SECUREPORT port to support different types of connections. The TELNETPARMS CONNTYPE statement has the following options:

- SECURE – SECURE is the default value and matches the behavior of earlier OS/390 releases. It specifies that an SSL or negotiated SSL connection is permitted.

- NEGTSECURE – NEGTSECURE specifies that a negotiated SSL connection is permitted.

- BASIC – BASIC specifies that a basic (i.e., non-SSL) connection is permitted.

- ANY – ANY specifies that a basic, SSL, or negotiated SSL connection is permitted.

- NONE – NONE specifies that no connection is permitted. This is used together with a BEGINVTAM PARMSGROUP statement that overrides this setting with different CONNTYPE values for specifically identified clients.

While sites may select any appropriate CONNTYPE configuration, the following guidelines apply:

- Special care should be taken in choosing the BASIC or ANY options. Using one of these options allows a session to be connected on a SECUREPORT port and to use non-SSL processing. Such a session would not use the authentication or encryption features offered by SSL.

- Sites should maintain at least one port defined with a CONNTYPE of SECURE. This ensures that there is a port on which SSL processing is always performed.

- A TELNETPARMS SSLTIMEOUT statement defines the SSL handshake timeout value. For connections eligible for SSL processing, the TN3270 Telnet server initiates the SSL handshake process and waits for a response from the client. If there is no response within the number of seconds specified by the SSLTIMEOUT statement, the server tries any additional connection types permitted by the TELNETPARMS configuration. If an SSL connection is required and the client does not respond in time, the connection is closed. A large number of connections, waiting for an extended period, could create a denial of service condition.

- Sites should choose an SSLTIMEOUT value that accommodates network performance without allowing individual connections to wait for an extended period. An SSLTIMEOUT value between 5 (the default) and 300 should be used.

Authentication

- Authentication is one of the primary features of SSL processing. The identity of the server and optionally the client is authenticated through the use of digital certificates. The TN3270 Telnet Server supports server only or server and client authentication.

- Server authentication is performed for all SSL connections. It is the process in which the client authenticates the server using the certificate provided by the server during connection processing. The required statements are TELNETPARMS SECUREPORT and, for OS/390 Release 2.10, TELNETGLOBALS KEYRING. These statements define the IP port used for the connection and the location of the digital certificates. Please refer to *Section 4.4.2.1.1, PROFILE.TCPIP Configuration Statements*, for the required settings.

- Client authentication is optional for SSL connections. It is the process in which the server authenticates the client using the certificate provided by the client during connection processing. In addition to the statements necessary for server authentication, a TELNETPARMS CLIENTAUTH statement is required. Options on the CLIENTAUTH statement, along with resources defined in the SAF SERVAUTH class, can be used to configure three levels of client authentication.

- The first level of client authentication is specified by the SSLCERT operand on a TELNETPARMS CLIENTAUTH statement. It provides the lowest level of client authentication. In this level the server validates the certificate sent from the client and checks to verify that the Certificate Authority that signed the client's certificate is considered trusted by the server. Please refer to the following section on certificate management for a discussion on the issue of Certificate Authorities.

- The second level of client authentication is specified by the SAFCERT operand on a TELNETPARMS CLIENTAUTH statement. This level adds an additional check to the first level. In the second level, client certificates should be registered in advance with the ACP. This registration provides a map from a certificate to an ID defined to the ACP. During connection processing, after the level 1 check, a lookup of the client's certificate in the ACP's database is performed. If the ACP does not have an ID associated with the certificate, the connection is not permitted. It should be noted that it is possible to use a facility known as certificate name filtering. Under this facility, individual user certificates are not defined to the ACP. Instead, criteria are defined to the ACP that allow multiple certificates to be mapped to a single ID, based on selected fields from the certificate. Please refer to the following section on certificate management for a discussion on the issue of certificate name filtering.

- The third level of client authentication is specified by a combination of the SAFCERT operand on a TELNETPARMS CLIENTAUTH statement and the definition of resources in the SERVAUTH SAF class. This level adds an additional check to the first two levels. In the third level, the ID that has been assigned by the ACP should have access to the appropriate resource in the SERVAUTH SAF class. This resource represents the specific IP port provided by the specific TCP/IP address space on the system. It has the form EZB.TN3270.*sysname*.*tcpipname*.PORT*nnnnn*, where *sysname* refers to the value of the MVS SYSNAME system symbol that is defined in SYS1.PARMLIB(IEASYMxx), *tcpipname* identifies the TCP/IP address space, and PORT*nnnnn* identifies the port number. During connection processing, after the Level 1 and 2 checks, a check is made to determine that the assigned ID has access to the resource that represents the IP port. If the ACP determines that the access is not allowed, the connection is not permitted.

Although it is expected that client authentication is required at a future date, the following guidelines are currently applicable:

- As resources permit, sites should use client authentication for as many users as possible. Level 1 authentication is less desirable; Level 2 or 3 authentication should be used.

- For users that hold special privileges within the ACP, Level 2 or 3 authentication should be used whenever possible.

Certificate Management

Digital certificates are a primary requirement for SSL processing. In this section the following considerations in managing certificates are discussed:

- Location – There are multiple options for storing certificates that the TN3270 Telnet Server can access.

- Origin – The origin of a certificate, the Certificate Authority, is crucial in determining if the certificate should be trusted.

- Name filtering – Multiple certificates can be mapped to a single ID.

- On OS/390 systems, an MVS data set, an HFS file, or, for OS/390 Release 2.10 and above, the resident ACP can be the storage location for digital certificates used by the TN3270 Telnet Server. When certificates are stored in MVS data sets or HFS files, the GSKKYMAN utility is used to manipulate them. When they are stored by the ACP, commands specific to the ACP are used.

Digital certificates should be stored according to the following guidelines:

- On OS/390 systems at Release 2.10 and above, the ACP should be used as the location for certificates. On older systems, an MVS data set should be used. This guideline is reflected in the requirements specified in *Section 4.4.2.1.1, PROFILE.TCPIP Configuration Statements*.

- Each digital certificate includes Certificate Authority (CA) information as the logical origin of the certificate. The presence of the CA’s information indicates, to some level of trust, that the owner of the certificate is recognized by that CA to be who they claim to be. Each host maintains a list of CAs that are considered trusted. When client authentication is utilized, the CA from the client’s certificate is compared to the host’s list. If there is a match, a major criterion of SSL authentication is satisfied. Therefore, the list of CAs maintained on the host has a crucial impact on authentication decisions.

- Software is available on most host platforms, including OS/390, which allows a host to act as a Certificate Authority. When certificates are created on that host for use on that host, the certificates are considered to be self-signed. Certificates that are self-signed are generally considered to be of limited security value because no independent oversight of user identification is maintained.

- *(ITNT0040: CAT II) The IAO will ensure that for production environments, the list of Certificate Authorities considered trusted by the OS/390 host are limited to those with a trust hierarchy that leads to a DOD PKI Root Certificate Authority.*

- *(ITNT0040: CAT II) The IAO will ensure that for production environments, self-signed certificates are not used.*

Certificate name filtering is a facility that allows multiple certificates to be mapped to a single ACP ID. Rather than matching a certificate stored in the ACP to look up an ID, certificate name filtering uses criteria rules stored in the ACP. A filter rule uses parts of the distinguished name of the certificate owner and/or issuer (CA) to determine an ID to assign to the user. Depending on the filter criteria, a large number of client certificates could map to a single ID.

- *(ITNT0040: CAT II) The IAO will ensure that certificate name filtering is not used unless the filtering rules have been documented to, and approved by, the IAM.*

Encryption

A key benefit from using SSL is the data privacy that is provided by session encryption. During the SSL connection process a mutually acceptable encryption algorithm is selected by the server and client. This algorithm is used to encrypt the data that subsequently flows between the two. However, the level or strength of encryption can vary greatly. In fact, certain configuration options can allow no encryption to be used; others can allow a relatively weak 40-bit algorithm to be used.

A TELNETPARMS ENCRYPTION statement is used to specify the encryption algorithms that the TN3270 Telnet Server can use on the associated port.

- *(ITNT0050: CAT II) The systems programmer responsible for supporting ICS will ensure that a TELNETPARMS ENCRYPTION statement is coded for each statement block that defines a SECUREPORT.*

- *(ITNT0050: CAT II) The systems programmer responsible for supporting ICS will ensure that to prevent the use of null or 40-bit encryption, each TELNETPARMS ENCRYPTION statement does not specify any of the following operands—SSL_NULL_Null, SSL_NULL_MD5, SSL_NULL_SHA, SSL_RC4_MD5_EX, or SSL_RC2_MD5_EX.*

If available and configured on the site's processor, hardware encryption support is used by the TN3270 Telnet Server. The encryption processes are handled via calls to the OS/390 System SSL component that, in turn, performs calls to the ICSF software. In this configuration, resources in the CSFSERV SAF class are used to control access to the ICSF services. Therefore the ID associated with the TN3270 Telnet Server should have appropriate access to the

CSFSERV resources. Please refer to the definition of the TCP/IP started task in the ACP-specific sections within *Section 4.4.1, Base TCP/IP System*, for implementation details.

4.4.2.1.5 SMF Recording

As determined by the TELNETPARMS SMFINIT and SMFTERM statements, the TN3270 Telnet Server can provide audit data in the form of SMF records. SMF record type 118, the TCP/IP Statistics record, can be written with the following subtypes:

- 20 – Session initiation
- 21 – Session termination

SMF data produced by the TN3270 Telnet Server provides information about individual sessions. The data includes the VTAM application, the remote and local IP addresses, and the remote and local IP port numbers. This data may provide valuable information for security audit activities.

- *(ITNT0060: CAT II) The systems programmer responsible for supporting ICS will ensure that the TELNETPARMS SMFINIT and SMFTERM statements are coded with the STD operand within each TELNETPARMS statement block.*

4.4.2.2 HFS Object Protection

There are no HFS objects that require special security procedures for elements of the TN3270 Telnet Server.

4.4.2.3 ACF2

This section describes the commands needed to implement the security guidelines for the TN3270 Telnet Server under the ACF2 ACP. The following task categories are described:

- Resource definitions
- Data set protection

4.4.2.3.1 Resources

The SAF resources associated with the TN3270 Telnet Server are used to enable secure connections between clients and the server. The configurations of secure connections are described in *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*. Although it is not a requirement at this time to configure secure connections, sites should begin to implement these configurations in anticipation of a future requirement.

The following topics are discussed in this section:

- Defining SERVAUTH resource controls for Telnet ports
- Defining digital certificates and key rings for the Telnet server

- Updating certificate resource controls to allow the Telnet server to list user certificates and key rings
- Updating CSFSERV resource controls to allow the Telnet server to access hardware encryption services.

It should be noted that the definitions and controls for digital certificate processing discussed in this section are based on OS/390 Release 2.10 and above. Sites with systems at OS/390 Release 2.8 have to use the GSKKYMAN utility to manipulate digital certificates. The SERVAUTH resource controls for Telnet ports and the CSFSERV resource controls can be configured for OS/390 Release 2.8 and above.

The TN3270 Telnet Server component defines resources in the SERVAUTH SAF class to control access to Telnet ports. The resources are used in configuring the third level of client authentication. Please refer to the discussion on authentication in *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*.

The following addition (in **bold**) to the indicated rule set can be used as an example to assign the privilege that is required for users of a Telnet port configured for the third level of client authentication:

$KEY(EZB) TYPE(SER)
…
TN3270.sysname**.**tcpipname**.PORT00992 UID(**Telnet-port-users-uids**)**
SERVICE(READ) ALLOW
…

The following operator command is required to complete the update:

```
F ACF2,REBUILD(SER)
```

The following considerations apply:

The *sysname* and *tcpipname* references are to the system name and TCP/IP started task name on the subject system. The port number in the rule, 992, could vary according to site network configuration.

If the SERVAUTH resource class has not been defined previously, refer to the ACF2-specific section within *Section 4.4.1, Base TCP/IP System*, for implementation requirements.

For the TN3270 Telnet Server to process SSL connections, the ID associated with the server should have a digital certificate and key ring. In addition, to authenticate client certificates, the certificates of Certificate Authorities should be available. To accomplish this for systems at OS/390 Release 2.10 and above, ACF2 is used as the certificate store.

The following commands can be used to insert the certificate, define a key ring, and connect the server's certificate and those of the Certificate Authorities to the server's key ring:

```
SET PROFILE(USER) DIVISION(CERTDATA)
INSERT TCPIP.CERT01 DSN(certificate-dataset) LABEL(TCPIP-Cert01)
TRUST

SET PROFILE(USER) DIVISION(KEYRING)
INSERT TCPIP.RING01  RINGNAME(TCPIP-Ring01)

CONNECT CERTDATA(TCPIP.CERT01)
KEYRING(TCPIP.RING01)
DEFAULT
CONNECT CERTDATA(CERTDATA-of-DOD-CLASS-3-Root-CA-Certificate)
KEYRING(TCPIP.RING01)
CONNECT CERTDATA(CERTDATA-of-DOD-PKI-Med-Root-CA-Certificate)
KEYRING(TCPIP.RING01)
```

The following operator commands are required to complete the updates:

```
F ACF2,REBUILD(USR),CLASS(P)
F ACF2,OMVS
```

The following considerations apply:

- The commands assume that TCPIP is the ID used for the TN3270 Telnet Server.

- The commands that connect the Certificate Authority certificates assume that these certificates have already been defined to ACF2.

- The values in RINGNAME and CERTDATA operands may include lower case characters.

- For the TN3270 Telnet Server to authenticate clients with digital certificates, the ID associated with the server should have access to read the clients' key rings and certificates. Resources in the FACILITY SAF class control this access.

- The following additions (in bold) to the indicated rule set can be used to assign the privileges that are required for the TCP/IP address space:

  ```
  $KEY(IRR) TYPE(FAC)
  …
  DIGTCERT.LIST UID(TCPIP-uid) SERVICE(UPDATE) ALLOW
  DIGTCERT.LISTRING UID(TCPIP-uid) SERVICE(UPDATE) ALLOW
  …
  ```

The following operator command is required to complete the updates:

```
    F ACF2,REBUILD(FAC)
```

In the discussion on encryption in *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, it is noted that the TN3270 Telnet Server makes use of hardware encryption if it is available and configured on the site's processor. Access to hardware encryption services is controlled via resources in the CSFSERV SAF class.

Because the TN3270 Telnet Server runs in the TCP/IP address space, the ID used for that address space should have appropriate access to the CSFSERV resources. Please refer to the definition of the TCP/IP started task in the ACF2-specific section within *Section 4.4.1, Base TCP/IP System*, for implementation details.

4.4.2.3.2 Data Sets

The vendor elements of the TN3270 Telnet Server are installed in data sets that are functionally owned by the Base TCP/IP System component. The local configuration data set for the TN3270 Telnet Server is shared with the Base TCP/IP System component. Please refer to *Section 4.4.1, Base TCP/IP System*, for the security requirements for those data sets.

For sites with systems at OS/390 Release 2.8 that are configured for SSL connections, there will be a key ring file and a stash file for the TN3270 Telnet Server. As required in *Section 4.4.2.1.1, PROFILE.TCPIP Configuration Statements*, the KEYRING operand on any SECUREPORT statement should specify an MVS data set. This key ring data set is created from the output of the GSKKYMAN utility, along with the companion stash file. Please refer to IBM's *OS/390 SecureWay Communications Server IP Configuration* document for details on creating these files. Because these files are critical in the SSL authentication process, access to them will be strictly controlled. The standard naming convention for these data sets is to use the prefix SYS3.TCPIP.

- *(ITNT0070: CAT II) The IAO will ensure that data sets containing the key ring and stash files for the TN3270 Telnet Server have all access restricted to the ID used for the TCP/IP started task, security personnel, and systems programming personnel.*

- *(ITNT0070: CAT II) The IAO will ensure that all write and allocate access to the data sets containing the key ring and stash files for the TN3270 Telnet Server is logged.*

The following additions (in bold) to the SYS3 rule set can be used as a base to secure the data sets:

```
$KEY(SYS3)
…
TCPIP.sysname.- UID(sysprog-uid) READ(A) WRITE(L) ALLOC(L) EXEC(A)
TCPIP.sysname.- UID(security-uid) READ(A) WRITE(L) ALLOC(L) EXEC(A)
TCPIP.sysname.- UID(TCPIP-uid) READ(A) WRITE(L) ALLOC(L) EXEC(A)
```

…

4.4.2.4 RACF

This section describes the commands needed to implement the security guidelines for the TN3270 Telnet Server under the RACF ACP. The following task categories are described:

- Resource definitions
- Data set protection

4.4.2.4.1 Resources

The SAF resources associated with the TN3270 Telnet Server are used to enable secure connections between clients and the server. The types of secure connections are described in *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*. Although it is not a requirement at this time to configure secure connections, sites should begin to implement these configurations in anticipation of a future requirement.

The following topics are discussed in this section:

- Defining SERVAUTH resource controls for Telnet ports
- Defining digital certificates and key rings for the Telnet server
- Updating certificate resource controls to allow the Telnet server to list user certificates and key rings
- Updating CSFSERV resource controls to allow the Telnet server to access hardware encryption services

It should be noted that the definitions and controls for digital certificate processing discussed in this section are based on OS/390 Release 2.10 and above. Sites with systems at OS/390 Release 2.8 have to use the GSKKYMAN utility to manipulate digital certificates. The SERVAUTH resource controls for Telnet ports and the CSFSERV resource controls can be configured for OS/390 Release 2.8 and above.

The TN3270 Telnet Server component defines resources in the SERVAUTH SAF class to control access to Telnet ports. The resources are used in configuring the third level of client authentication. Please refer to the discussion on authentication in *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*.

The following commands can be used as an example to assign the privilege that is required for users of a Telnet port configured for the third level of client authentication:

- RDEFINE SERVAUTH EZB.TN3270.*sysname.tcpipname*.PORT00992 -
- UACC(NONE) OWNER(ADMIN)
- PERMIT EZB.TN3270.*sysname.tcpipname*.PORT00992 -
- CLASS(SERVAUTH) ACCESS(READ) ID(*Telnet-port-users*)
- SETROPTS RACLIST(SERVAUTH) REFRESH

The following considerations apply:

- The *sysname* and *tcpipname* references are to the system name and TCP/IP started task name on the subject system. The port number in the commands, 992, could vary according to site network configuration.

- If the SERVAUTH resource class has not been defined previously, refer to the RACF-specific section within *Section 4.4.1, Base TCP/IP System*, for implementation requirements.

For the TN3270 Telnet Server to process SSL connections, the ID associated with the server should have a digital certificate and key ring. In addition, to be able to authenticate client certificates, the certificates of Certificate Authorities should be available. To accomplish this for systems at OS/390 Release 2.10 and above, RACF is used as the certificate store.

The following commands can be used to insert the certificate, define a key ring, and connect the server's certificate and those of the Certificate Authorities to the server's key ring:

```
RACDCERT ID(TCPIP) ADD('certificate-dataset')  -
WITHLABEL('TCPIP-Cert01') TRUST -
PASSWORD('pkcs12-cert-pswd') -            /* Used if PKCS #12 format only */
ICSF                                      /* Optional if hardware encryption active */

RACDCERT ID(TCPIP) ADDRING(TCPIP-Ring01)

RACDCERT ID(TCPIP) CONNECT(ID(TCPIP) -
LABEL('TCPIP-Cert01') RING(TCPIP-Ring01) -
DEFAULT)
RACDCERT ID(TCPIP) CONNECT(CERTAUTH  -
LABEL('Label-of-DOD-CLASS-3-Root-CA-Certificate') RING(TCPIP-Ring01))
RACDCERT ID(TCPIP) CONNECT(CERTAUTH  -
LABEL('Label-of-DOD-PKI-Med-Root-CA-Certificate') RING(TCPIP-Ring01))

SETROPTS RACLIST(DIGTCERT) REFRESH
SETROPTS RACLIST(DIGTRING) REFRESH
```

The following considerations apply:

- The commands assume that TCPIP is the ID used for the TN3270 Telnet Server.

- The commands that connect the Certificate Authority certificates assume that these certificates have already been defined to RACF.

- The values in WITHLABEL, ADDRING, LABEL, and RING operands may include lower case characters.

- If the certificate in the data set containing the server's certificate is in PKCS #12 format, the PASSWORD operand is required.

- If hardware encryption is enabled, the ICSF operand may be used.

- For the TN3270 Telnet Server to authenticate clients with digital certificates, the ID associated with the server should have access to read the clients' key rings and certificates. Resources in the FACILITY SAF class control this access.

The following commands can be used to assign the privileges that are required for the TCP/IP address space:

```
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) -
ACCESS(UPDATE) ID(TCPIP)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) -
ACCESS(UPDATE) ID(TCPIP)
SETROPTS RACLIST(FACILITY) REFRESH
```

These commands assume that TCPIP is the ID used for the TN3270 Telnet Server.

In the discussion on encryption in *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, it is noted that the TN3270 Telnet Server makes use of hardware encryption if it is available and configured on the site's processor. Access to hardware encryption services is controlled via resources in the CSFSERV SAF class.

Because the TN3270 Telnet Server runs in the TCP/IP address space, the ID used for that address space should have appropriate access to the CSFSERV resources. Please refer to the definition of the TCP/IP started task in the RACF-specific section within *Section 4.4.1, Base TCP/IP System*, for implementation details.

4.4.2.4.2 Data Sets

The vendor elements of the TN3270 Telnet Server are installed in data sets that are functionally owned by the Base TCP/IP System component. Please refer to *Section 4.4.1, Base TCP/IP System*, for the security requirements for those data sets.

For sites with systems at OS/390 Release 2.8 that are configured for SSL connections, there will be a key ring file and a stash file for the TN3270 Telnet Server. As required in *Section 4.4.2.1.1, PROFILE.TCPIP Configuration Statements*, the KEYRING operand on any SECUREPORT statement will specify an MVS data set. This key ring data set is created from the output of the GSKKYMAN utility, along with the companion stash file. Please refer to IBM's *OS/390 SecureWay Communications Server IP Configuration* document for details on creating these files. Because these files are critical in the SSL authentication process, access to them will be strictly controlled. The standard naming convention for these data sets is to use the prefix SYS3.TCPIP.

- *(ITNT0070: CAT II) The IAO will ensure that data sets containing the key ring and stash files for the TN3270 Telnet Server have all access restricted to the ID used for the TCP/IP started task, security personnel, and systems programming personnel.*

- *(ITNT0070: CAT II) The IAO will ensure that all update and alter access to the data sets containing the key ring and stash files for the TN3270 Telnet Server is logged.*

The following commands can be used to provide the required access control for the MVS data sets:

```
ADDSD ‘SYS3.TCPIP.sysname.**’ OWNER(SYS1) UACC(NONE) -
    AUDIT(ALL(UPDATE))
PERMIT ‘SYS3.TCPIP.sysname.**’ ACCESS(ALTER) ID(sysprog-group)
PERMIT ‘SYS3.TCPIP.sysname.**’ ACCESS(ALTER) ID(security-group)
PERMIT ‘SYS3.TCPIP.sysname.**’ ACCESS(ALTER) ID(TCPIP)
```

4.4.2.5 TOP SECRET

This section describes the commands needed to implement the security guidelines for the TN3270 Telnet Server under the TOP SECRET ACP. The following task categories are described:

- Resource definitions
- Data set protection

4.4.2.5.1 Resources

The SAF resources associated with the TN3270 Telnet Server are used to enable secure connections between clients and the server. The types of secure connections are described in *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*. Although it is not a requirement at this time to configure secure connections, sites should begin to implement these configurations in anticipation of a future requirement.

The following topics are discussed in this section:

- Defining SERVAUTH resource controls for Telnet ports
- Defining digital certificates and key rings for the Telnet server
- Updating certificate resource controls to allow the Telnet server to list user certificates and key rings
- Updating CSFSERV resource controls to allow the Telnet server to access hardware encryption services.

It should be noted that the definitions and controls for digital certificate processing discussed in this section are based on OS/390 Release 2.10 and above. Sites with systems at OS/390 Release 2.8 have to use the GSKKYMAN utility to manipulate digital certificates. The SERVAUTH resource controls for Telnet ports and the CSFSERV resource controls can be configured for OS/390 Release 2.8 and above.

The TN3270 Telnet Server component defines resources in the SERVAUTH SAF class to control access to Telnet ports. The resources are used in configuring the third level of client authentication. Please refer to the discussion on authentication in *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*.

The following command can be used as an example to assign the privilege that is required for users of a Telnet port configured for the third level of client authentication:

```
TSS PERMIT(Telnet-port-users)
SERVAUTH(EZB.TN3270.sysname.tcpipname.PORT00992)ACCESS(READ)
```

The following considerations apply:

- The *sysname* and *tcpipname* references are to the system name and TCP/IP started task name on the subject system. The port number in the command, 992, could vary according to site network configuration.

- If the SERVAUTH resource class has not been defined previously, refer to the TOP SECRET-specific section within *Section 4.4.1, Base TCP/IP System*, for implementation requirements.

- For the TN3270 Telnet Server to process SSL connections, the ID associated with the server should have a digital certificate and key ring. In addition, to be able to authenticate client certificates, the certificates of Certificate Authorities should be available. To accomplish this for systems at OS/390 Release 2.10 and above, TOP SECRET is used as the certificate store.

The following commands can be used to insert the certificate, define a key ring, and connect the server's certificate and those of the Certificate Authorities to the server's key ring:

```
TSS ADD(TCPIP) DIGICERT(CERT01)
DCDSN(certificate-dataset)
LABLCERT('TCPIP-Cert01') TRUST
ICSF                                /* Optional if hardware encryption active */

TSS ADD(TCPIP) KEYRING(RING01)
LABLRING('TCPIP-Ring01')

TSS ADD(TCPIP) KEYRING(RING01)
RINGDATA(TCPIP.CERT01)
DEFAULT
TSS ADD(TCPIP) KEYRING(RING01)
RINGDATA(CERTAUTH.digicert-of-DOD-CLASS-3-Root-CA-Certificate)
TSS ADD(TCPIP) KEYRING(RING01)
RINGDATA(CERTAUTH.digicert-of-DOD-PKI-Med-Root-CA-Certificate)
```

The following considerations apply:

- The commands assume that TCPIP is the ID used for the TN3270 Telnet Server. The commands that connect the Certificate Authority certificates assume that these certificates have already been defined to TOP SECRET.
- The values in LABLCERT, LABLRING and CERTAUTH operands may include lower case characters.

- If hardware encryption is enabled, the ICSF operand may be used.

For the TN3270 Telnet Server to authenticate clients with digital certificates, the ID associated with the server should have access to read the clients' key rings and certificates. Resources in the FACILITY SAF class control this access.

The following commands can be used to assign the privileges that are required for the TCP/IP address space:

```
TSS PERMIT(TCPIP) IBMFAC(IRR.DIGTCERT.LIST)
ACCESS(UPDATE)
TSS PERMIT(TCPIP) IBMFAC(IRR.DIGTCERT.LISTRING)
ACCESS(UPDATE)
```

These commands assume that TCPIP is the ID used for the TN3270 Telnet Server.

In the discussion on encryption in *Section 4.4.2.1.4, Secure Sockets Layer (SSL) Connections*, it is noted that the TN3270 Telnet Server makes use of hardware encryption if it is available and configured on the site's processor. Access to hardware encryption services is controlled via resources in the CSFSERV SAF class.

Because the TN3270 Telnet Server runs in the TCP/IP address space, the ID used for that address space should have appropriate access to the CSFSERV resources. Please refer to the definition of the TCP/IP started task in the TOP SECRET-specific sections within *Section 4.4.1, Base TCP/IP System*, for implementation details.

4.4.2.5.2 Data Sets

The vendor elements of the TN3270 Telnet Server are installed in data sets that are functionally owned by the Base TCP/IP System component. Please refer to *Section 4.4.1, Base TCP/IP System*, for the security requirements for those data sets.

For sites with systems at OS/390 Release 2.8 that are configured for SSL connections, there will be a key ring file and a stash file for the TN3270 Telnet Server. As required in *Section 4.4.2.1.1, PROFILE.TCPIP Configuration Statements*, the KEYRING operand on any SECUREPORT statement will specify an MVS data set. This key ring data set is created from the output of the GSKKYMAN utility, along with the companion stash file. Please refer to IBM's *OS/390 SecureWay Communications Server IP Configuration* document for details on creating these files. Because these files are critical in the SSL authentication process, access to them will be strictly controlled. The standard naming convention for these data sets is to use the prefix SYS3.TCPIP.

- *(ITNT0070: CAT II) The IAO will ensure that data sets containing the key ring and stash files for the TN3270 Telnet Server have all access restricted to the ID used for the TCP/IP started task, security personnel, and systems programming personnel.*

- *(ITNT0070: CAT II) The IAO will ensure that all update and allocate access to the data sets containing the key ring and stash files for the TN3270 Telnet Server is logged.*

The following commands can be used to provide the required access control for the MVS data sets:

```
TSS ADD(SYS3) DSN(SYS3)
TSS PERMIT(sysprog-group) DSN(SYS3.TCPIP.sysname) ACCESS(ALL)
    ACTION(AUDIT)
TSS PERMIT(security-group) DSN(SYS3.TCPIP.sysname) ACCESS(ALL)
    ACTION(AUDIT)
TSS PERMIT(TCPIP) DSN(SYS3.TCPIP.sysname) ACCESS(ALL)
    ACTION(AUDIT)
```

4.4.3 OS/390 UNIX Telnet Server

The OS/390 UNIX Telnet Server, known as otelnetd, provides interactive access to the OS/390 UNIX shell. This access supports client terminal applications that execute in either raw mode (known as character mode) or line mode. This access is necessary for some OS/390 UNIX applications (such as the vi editor) that cannot execute over a connection that uses the TN3270 protocol.

Compared to Telnet servers on other platforms, the OS/390 UNIX Telnet Server has some different characteristics that can impact the system’s security configuration. In the OS/390 UNIX environment, otelnetd is not a constantly running process and one instance does not support multiple users. The following details describe the connection process:

- The inetd daemon listens for connection requests on the assigned TCP port.

- When a connection request is received, the inetd daemon starts a new instance of the otelnetd process. Parameters in the inetd daemon’s configuration file, usually named /etc/inetd.conf, specify the otelnetd command, its parameters, and the user account under which the process starts.

- Each active Telnet session requires an instance of otelnetd executing on the OS/390 system. When the session is terminated, otelnetd exits.

- The otelnetd process begins execution under the user account specified in the inetd configuration file. After the user enters their ID and password, otelnetd switches to the security context of the user's account.

4.4.3.1 General Considerations

The configuration issues addressed relative to the security environment for otelnetd include:

- The startup user account for otelnetd
- Startup parameters specified for the otelnetd command
- The otelnetd login banner

The user account used at the startup of otelnetd is specified in the inetd configuration file. This account is used to perform the identification and authentication of the user requesting the session. Because the account is only used until user authentication is completed, there is no need for a unique account for this function. The OS/390 UNIX kernel account can be used.

- *(IUTN0010: CAT II) The systems programmer responsible for supporting ICS will ensure that the startup user account for otelnetd is the account defined for the OS/390 UNIX kernel.*

The startup parameters used for otelnetd are specified in the inetd configuration file. Some of these parameters have an impact on system security and therefore require specific settings.

- *(IUTN0020: CAT II) The systems programmer responsible for supporting ICS will ensure that the otelnetd startup command includes the options -D login and -c 900, where:*

  ***-D login** indicates that messages should be written to the syslogd facility for login and logout activity*

  ***-c 900** indicates that the Telnet session should be terminated after 15 minutes of inactivity.*

NOTE: The 900 is the maximum value; any value between 1 and 900 is acceptable.

- *(IUTN0020: CAT II) The systems programmer responsible for supporting ICS will ensure that the otelnetd startup command does not include the option -h, where:*

  -h indicates that the logon banner should not be displayed.

NOTE: Other startup options may be used at the site's discretion. An example of a suitable startup command is otelnetd -D login -U -c 300.

DOD requires that a logon warning banner be displayed. Although the OS/390 UNIX Telnet Server does not support the display of a message before the logon prompt, it is possible to display a message immediately after logon.

- *(IUTN0030: CAT II) The systems programmer responsible for supporting ICS will ensure that the /etc/banner file contains the warning logon.*

4.4.3.2 HFS Object Protection

In order to protect the OS/390 UNIX Telnet Server component, special security settings will be applied to selected HFS files.

- *(IUTN0040: CAT II) The systems programmer responsible for supporting ICS will ensure that the permission bits and user audit bits for HFS objects that are part of the OS/390 UNIX Telnet Server component is configured according to the settings in the following table:*

Table A-49. OS/390 UNIX TELNET SERVER HFS OBJECT SECURITY SETTINGS (4.4.3.2 a)

OS/390 UNIX TELNET SERVER HFS OBJECT SECURITY SETTINGS			
DIRECTORY or FILE	PERMISSION BITS	USER AUDIT BITS	FUNCTION
/usr/sbin/otelnetd	1740	fff	Daemon program
/etc/banner	0744	faf	Login banner

NOTE: The /usr/sbin/otelnetd object is a symbolic link to /usr/lpp/tcpip/sbin/otelnetd. The permission and user audit bits on the target of the symbolic link will have the required settings.

The following commands can be used (from a user account with an effective UID(0)) to update the permission bits and audit bits:

```
chmod  1740  /usr/lpp/tcpip/sbin/otelnetd
chaudit  rwx=f  /usr/lpp/tcpip/sbin/otelnetd
chmod  0744  /etc/banner
chaudit  w=sf,rx+f  /etc/banner
```

4.4.3.3 ACF2

There are no ACF2 user accounts or privileges that need to be created or assigned for the OS/390 UNIX Telnet server as long as the OS/390 UNIX kernel account is assigned as the startup account. That account should have already been created with a UID(0) and access to the BPX.DAEMON FACILITY class resource.

4.4.3.4 RACF

There are no RACF user accounts or privileges that need to be created or assigned for the OS/390 UNIX Telnet server as long as the OS/390 UNIX kernel account is assigned as the startup account. That account should have already been created with a UID(0) and access to the BPX.DAEMON FACILITY class resource.

4.4.3.5 TOP SECRET

There are no TOP SECRET user accounts or privileges that need to be created or assigned for the OS/390 UNIX Telnet server as long as the OS/390 UNIX kernel account is assigned as the startup account. That account should have already been created with a UID(0) and access to the BPX.DAEMON FACILITY class resource.

4.4.4 FTP Server

The File Transfer Protocol (FTP) Server component of IBM's Communications Server provides the server portion of the client \ server application for transferring files between hosts. Because the server implements the industry standard File Transfer Protocol on OS/390, standards-compliant clients running on any other host can connect to the server.

It should be noted that the information in this section has been prepared to address the configuration requirements as of Version 2, Releases 8 and 10 of OS/390. IBM fundamentally altered the FTP Server component in OS/390 Version 2, Release 5, to utilize OS/390 UNIX System Services. That architecture, with enhancements, is used in the current releases of Communications Server.

The tasks supported by the FTP Server can be grouped into three categories:

- Standard functions that support transferring data and navigating the server's file systems

- Standard functions beyond the scope of data transfer. These include renaming files, deleting files, and changing account passwords.

- OS/390-specific functions including interfacing with JES2 and DB2.

From a high level point of view, the FTP Server consists of the following elements:

- The FTP daemon program, ftpd, performs initialization and listens for incoming client connections.

- The FTP server program, ftpdns, is executed once for each client connection and processes the client's commands until the connection is terminated.

The startup and execution of the FTP Server on OS/390 differs from its counterparts on other platforms. The differences impact the system's security configuration and therefore justify some brief details:

The FTP daemon is started, processes the configuration files, and verifies that communications with the TCP/IP communications stack is available. In default configurations the daemon's address space has a job name of FTPD.

After initialization is complete, the FTP daemon creates (via the fork service) a new address space that executes the daemon, listening for client connection requests. The original address space terminates. In default configurations the new daemon address space has a job name of FTPD1.

Each time the FTP daemon receives a connection request, it creates (via the fork service) a new address space running the FTP server program. Each active FTP session requires an instance of the FTP server program. In default configurations the server address space initially has a job name of FTPDn, where the **n** is a number 1 through 9.

The FTP server program performs identification and authentication of the client, switches its security context to that of the client, and then processes the commands sent from the client. Following successful client authentication, the job name of the server's address space is changed by default to match the user identification supplied by the client. When the client terminates the connection, the address space running the server program terminates.

The job names assigned to the various address spaces are based on OS/390 UNIX defaults. The use of an eight-character startup JCL procedure can change the default behavior.

4.4.4.1 General Considerations

There are a number of configuration issues addressed relative to the security environment for the FTP Server. The following issues are addressed in the next subsections:

- Startup procedure choices and parameters
- Configuration files
- Configuration statements in the FTP.DATA file
- User exits
- Banner message requirements
- SMF recording
- Interface to JES2
- Interface to DB2

An additional issue involves the use of the Syslog daemon. The FTP daemon writes log messages to the OS/390 system console if the Syslog daemon is not running. If the FTP TRACE option is used and the Syslog daemon is not running, the system console might receive enough messages to significantly degrade system operations. See *Section 4.4.7, Syslog Daemon*, for information on Syslog requirements.

4.4.4.1.1 Startup Procedure Choices and Parameters

As with other OS/390 UNIX-based components, there are multiple ways to start the FTP daemon. It can be started through the AUTOLOG subtask in the TCPIP address space, through an OS/390 started task, or through a command in the /etc/rc file in the OS/390 UNIX environment.

When the FTP daemon is started through the AUTOLOG subtask or through an OS/390 started task, a conventional JCL procedure (PROC) is used. Using a PROC enhances security because it enables the explicit specification of job name, program parameters, and configuration files.

- *(IFTP0010: CAT II) The systems programmer responsible for supporting ICS will ensure that the FTP daemon runs under its own user account. Specifically, it does not share the account defined for the OS/390 UNIX kernel.*

The FTP daemon program can accept parameters in the JCL procedure that is used to start the daemon. The ANONYMOUS and ANONYMOUS= keywords are designed to allow anonymous FTP connections. The INACTIVE keyword is designed to set the timeout value for inactive connections. Controlling these options through the configuration file statements rather than the startup parameters reduces ambiguity.

- *(IFTP0020: CAT II) The systems programmer responsible for supporting ICS will ensure that the startup parameters for the FTP daemon does not include the ANONYMOUS, ANONYMOUS=, or INACTIVE keywords.*

4.4.4.1.2 Configuration Files

The FTP Server components read two configuration files that contain operational parameters. Because system security is impacted by some of the parameter settings, the files themselves should be protected and certain parameter settings should be specified.

The first configuration file is referred to as the **TCPIP.DATA** file. It includes the **DATASETPREFIX** parameter that specifies the high level qualifier that may be used in dynamic allocations of some data sets used by the FTP Server. The **TCPIP.DATA** file is used primarily by the TCP/IP address space. Please refer to *Section 4.4.1, Base TCP/IP System*, for the security requirements for the file and its parameters.

The second configuration file is referred to as the FTP.DATA file. Although the FTP Server components can use default parameter settings if the file is not specified, those default settings do not provide an adequate level of security. Please refer to the next section for a discussion of required parameter settings.

During initialization the FTP daemon searches multiple locations for the TCPIP.DATA and FTP.DATA files according to fixed sequences. However, uncertainty is reduced and security auditing is enhanced by explicitly specifying the locations of the files. In the daemon’s started task JCL, Data Definition (DD) statements can be used to specify the locations of the files. The SYSTCPD DD statement identifies the TCPIP.DATA file and the SYSFTPD DD statement identifies the FTP.DATA file.

- *(IFTP0020: CAT II) The systems programmer responsible for supporting ICS will ensure that the FTP daemon’s started task JCL specifies the SYSTCPD and SYSFTPD DD statements for configuration files.*

Required access controls for the **FTP.DATA** configuration file is described in the ACP-specific subsections that follow.

4.4.4.1.3 FTP.DATA Configuration Statements

The statements in the FTP.DATA configuration file specify the parameters and values that control the operation of the FTP Server components. Several of the parameters should have specific settings to provide a secure configuration. This section details the security-related parameters and how they should be set.

The interpretation of the FTP.DATA configuration file allows for three potential states:

- An option is set when a parameter is not coded.
- An option is set when a parameter is coded without any value assigned to it.
- An option is set when a parameter is coded with a specific value assigned to it.

- *(IFTP0030: CAT II, IFTP0060: CAT II) The systems programmer responsible for supporting ICS will ensure that the FTP Server FTP.DATA configuration statements are coded according to the settings in the following table:*

Table A-50. FTP.DATA CONFIGURATION STATEMENTS (4.4.4.1.3 a)

FTP.DATA CONFIGURATION STATEMENTS	
STATEMENT	NOT CODED, CODED WITHOUT VALUE, OR PARAMETER VALUE
ANONYMOUS	[Not Coded]
BANNER [For OS/390 2.10 and later]	[An HFS file, e.g., /etc/ftp.banner, or an OS/390 data set]
INACTIVE	[A value between 1 and 900]
JESINTERFACELEVEL [For OS/390 2.10 and later]	1 [See Note 1]
SMF	STD
SMFAPPE	[Not Coded]
SMFDEL	[Not Coded]
SMFEXIT	[Not Coded]
SMFJES	[Coded without Value]
SMFLOGN	[Not Coded]
SMFREN	[Not Coded]
SMFRETR	[Not Coded]
SMFSQL	[Coded without Value]
SMFSTOR	[Not Coded]
UMASK	077 [See Note 2]

NOTE: The JESINTERFACELEVEL statement may be coded with a value of **2** if the appropriate SAF resources (in the JESSPOOL and SDSF classes) have been protected. See *Section 4.4.4.1.7, Interface to JE***S**, for information.

NOTE: If the FTP Server requires a UMASK value less restrictive than **077**, requirements should be justified and documented with the IAO.

4.4.4.1.4 User Exits

There are a number of user exit points in the FTP Server component that permit customization of its operating behavior. The following table lists the exits with a brief description of their function:

Table A-51. FTP SERVER USER EXITS (4.4.4.1.4 a)

FTP SERVER USER EXITS	
EXIT NAME	FUNCTION
FTCHKCMD	Control which FTP commands a user is allowed to use
FTCHKIP	Control which client hosts, by IP address, may connect to the FTP server
FTCHKJES	Control which users are allowed to submit batch jobs
FTCHKPWD	Control which users are allowed to log on to the FTP server
FTPOSTPR [For OS/390 2.10 and later]	Perform post processing tasks (such as writing syslog messages) following the completion of some FTP commands
FTPSMFEX	Perform modifications to SMF records

The FTPSMFEX exit is enabled when the SMFEXIT parameter is coded in the FTP.DATA configuration file. The remaining exits are enabled when their load modules are in the FTP daemon’s STEPLIB or in the system link list or link pack area (LPA). The data sets in which the exits reside should be APF-authorized and program controlled.

Implementation of any of the FTP Server user exits is subject to the requirements for exits as specified in *Section 2.1.2, Software Integrity*, and *Section 2.1.2.6, OS/390 and Other Product Exits*. As of the publication of this document, DISA FSO has not approved the use of any FTP Server user exits.

- *(IFTP0040: CAT II) The systems programmer responsible for supporting ICS will ensure that the FTP Server user exits are not implemented.*

4.4.4.1.5 Warning Banner

DOD requires that a logon warning banner be displayed. As of OS/390 Release 2.10, the FTP Server supports the display of a message immediately after a new connection is established. The FTP.DATA BANNER parameter controls this behavior.

- *(IFTP0050: CAT II) The systems programmer responsible for supporting ICS will ensure that for systems at OS/390 Release 2.10 and above, the FTP.DATA BANNER parameter specifies an HFS file or OS/390 data set that contains the warning logon banner.*

4.4.4.1.6 SMF Recording

The FTP Server can provide audit data in the form of SMF records. SMF record type 118, the TCP/IP Statistics record, can be written with the following subtypes:

- 70 – Append
- 71 – Delete and Multiple Delete
- 72 – Invalid Logon Attempt
- 73 – Rename
- 74 – Get (Retrieve) and Multiple Get
- 75 – Put (Store and Store Unique) and Multiple Put

SMF data produced by the FTP Server provides transaction information for both successful and unsuccessful FTP commands. This data may provide valuable information for security audit activities.

- *(IFTP0060: CAT II) The systems programmer responsible for supporting ICS will ensure that the FTP Server is configured to write SMF records for all eligible events.*

4.4.4.1.7 Interface to JES

The FTP Server provides an interface to JES that allows an FTP client to submit, retrieve, and delete jobs as well as listing their status. The interface is enabled for an FTP connection when the FILETYPE JES parameter is specified in the FTP.DATA configuration file or when the client issues a SITE FILETYPE=JES command.

In OS/390 Release 2.8 the FTP client is restricted to listing, retrieving, or deleting only held jobs with names that match their logged in client ID plus one character. There are no security controls that apply to the Release 2.8 environment.

As of OS/390 Release 2.10, the FTP Server's interface to JES2 has been enhanced to permit significantly more client access. By default the behavior of prior releases is maintained. When the JESINTERFACELEVEL parameter in the FTP.DATA configuration file is set to the value **2**, the enhanced support is activated. The additional access is enabled through new commands and is controlled through resources in existing SAF classes.

With JESINTERFACELEVEL 2, access to JES data is controlled at two levels. The first level involves filtering criteria. Users can be given access to SAF resources that allow them to issue commands to alter the filtering criteria. The second level involves the retrieval or deletion commands. Users can be given *read* access to SAF resources to allow them to retrieve JES data or *alter* access to those resources to allow them to retrieve or delete JES data.

The SAF resources that control the filtering commands are subsets of those defined for the System Display and Search Facility (SDSF) product. The resources that control the retrieval and deletion commands are those defined for JES SPOOL data. The following table summarizes the SAF resources, the class they belong to, and the associated FTP client commands:

Table A-52. FTP SERVER JES INTERFACE SAF RESOURCES (4.4.4.1.7 a)

FTP SERVER JES INTERFACE SAF RESOURCES		
SAF RESOURCE	SAF CLASS	CLIENT COMMANDS
nodeid.userid.jobname.jobid.Dsid.dsname	JESSPOOL	DELETE, MDELETE
nodeid.userid.jobname.jobid.Dsid.dsname	JESSPOOL	GET, MGET
ISFCMD.FILTER.PREFIX	SDSF	JESJOBNAME, GET, MGET, DIR, LIST
ISFCMD.FILTER.OWNER	SDSF	JESOWNER, GET, MGET, DIR, LIST
ISFCMD.DSP.INPUT.*jesx* ISFCMD.DSP.ACTIVE.*jesx* ISFCMD.DSP.OUTPUT.*jesx* where *jesx* is the Job Entry Subsystem name (e.g., JES2)	SDSF	JESSTATUS, LIST, NLIST

The JESJOBNAME, JESOWNER, and JESSTATUS commands are actually subcommands of the client SITE command. They set up filtering criteria as follows:

- JESJOBNAME limits data based on the name of the job.
- JESOWNER limits data based on the ID of the owner of the job.
- JESSTATUS limits data based on the JES queue status of the job. This status can be INPUT, ACTIVE, OUTPUT, or ALL.

It is apparent that control of the resources in the JESSPOOL and SDSF SAF classes is critical to maintaining security for the FTP interface to JES. However, because the interface uses resources that are already defined to secure the use of SDSF to access JES data, most sites have already taken the required steps. Information on securing the JESSPOOL class resources is discussed in *Section 5.1.4, Security Controls for JES2 SPOOL Data Sets*.

- *(IFTP0030: CAT II) The systems programmer responsible for supporting ICS will ensure that for systems at OS/390 Release 2.10 and above, the FTP.DATA JESINTERFACELEVEL parameter is set to* ***1*** *unless the resources in the JESSPOOL and SDSF SAF classes have been protected.*

Refer to IBM's *OS/390 IBM Communications Server IP Configuration Guide* document for details on customizing the FTP interface to JES.

4.4.4.1.8 Interface to DB2

The FTP Server provides an interface to IBM's DB2 database management system that allows an FTP client to submit queries (via SQL SELECT) and retrieve the output. The interface is enabled for an FTP connection when the FILETYPE SQL parameter is specified in the FTP.DATA configuration file or when the client issues a SITE FILETYPE=SQL command.

To use this interface, a valid DB2 subsystem name and DB2 plan should be specified in the FTP.DATA configuration file. The FTP client may also set the DB2 subsystem name dynamically using the SITE DB2= command. IBM supplies a sample for defining and permitting access to the DB2 plan in the *OS/390 IBM Communications Server IP Configuration Guide* document (OS/390 Release 2.10) or *OS/390 SecureWay Communications Server IP Configuration* document (OS/390 Release 2.8).

Security for the DB2 interface is controlled via the DB2 access controls. These controls are typically implemented through GRANT operations. The site should ensure that only appropriate DB2 data would be made available when this interface is enabled.

4.4.4.2 HFS Object Protection

In order to protect the FTP Server component, special security settings are applied to selected HFS files.

- *(IFTP0070: CAT II) The systems programmer responsible for supporting ICS will ensure that the permission bits and user audit bits for HFS objects that are part of the FTP Server component are configured according to the settings in the following table:*

Table A-53. FTP SERVER HFS OBJECT SECURITY SETTINGS (4.4.4.2 a)

FTP SERVER HFS OBJECT SECURITY SETTINGS			
DIRECTORY or FILE	PERMISSION BITS	USER AUDIT BITS	FUNCTION
/usr/sbin/ftpd	1740	fff	Daemon program
/usr/sbin/ftpdns	1755	fff	Server program
/etc/ftp.data	0744	faf	Daemon configuration file
/etc/ftp.banner	0744	faf	Connection message

NOTE: The /usr/sbin/ftpd and /usr/sbin/ftpdns objects are symbolic links to /usr/lpp/tcpip/sbin/ftpd and /usr/lpp/tcpip/sbin/ftpdns respectively. The permission and user audit bits on the targets of the symbolic links should have the required settings.

NOTE: The /etc/ftp.data file may not be the configuration file the server uses. It is necessary to check the startup PROC to determine the actual file.

NOTE: The etc/ftp.banner file may not be the banner file the server uses. It is necessary to check the configuration file to determine the actual file.

Some of the files listed above (e.g., /etc/ftp.data) are not used in every configuration. While the absence of a file is generally not a security issue, the existence of a file that has not been properly secured can often be an issue. Therefore, all files that do exist should have the specified permission and audit bit settings.

The following commands can be used (from a user account with an effective UID(0)) to update the permission bits and audit bits:

```
chmod  1740  /usr/lpp/tcpip/sbin/ftpd
chaudit  rwx=f  /usr/lpp/tcpip/sbin/ftpd
chmod  1755  /usr/lpp/tcpip/sbin/ftpdns
chaudit  rwx=f  /usr/lpp/tcpip/sbin/ftpdns
chmod  0744  /etc/ftp.data
chaudit  w=sf,rx+f  /etc/ftp.data
chmod  0744  /etc/ftp.banner
chaudit  w=sf,rx+f  /etc/ftp.banner
```

4.4.4.3 ACF2

This section describes the commands needed to implement the security guidelines for the FTP Server under the ACF2 ACP. The following task categories are described:

- Started task definitions
- Data set protection

4.4.4.3.1 Started Tasks

The FTP Server component requires the definition of one started task for the FTP daemon.

- *(IFTP0010: CAT II) The IAO will ensure that the user account used for the FTP daemon is defined with the following characteristics:*

  - Named FTPD
  - Privilege to run as a started task
  - OS/390 UNIX attributes: UID(0), home directory '/', shell program /bin/sh
  - Access to the BPX.DAEMON resource

The following commands can be used to create the user account that is required for the FTP daemon:

```
SET LID
INSERT FTPD NAME(FTPD) GROUP(STCTCPX) STC
SET PROFILE(USER) DIVISION(OMVS)
INSERT FTPD UID(0) HOME(/) PROGRAM(/bin/sh)
```

NOTE: At eTrust CA-ACF2 6.4 and above, the PROGRAM field in the user profile record has been renamed to OMVSPGM.

The following rule set addition (in bold) can be used to assign the privileges that are required for the FTP daemon:

```
$KEY(BPX) TYPE(FAC)
…
DAEMON UID(FTPD-uid) SERVICE(READ) ALLOW
…
```

The following operator command is required to complete the update:

```
F ACF2,REBUILD(FAC)
```

4.4.4.3.2 Data Sets

The vendor elements of the FTP Server are installed in data sets that are functionally owned by the Base TCP/IP System component. Please refer to *Section 4.4.1, Base TCP/IP System*, for the security requirements for those data sets.

There may be one or more FTP Server data sets that are used to hold site-customized data. [This data could reside in HFS files rather than data sets.] This includes the FTP.DATA configuration file and the file used for the warning banner. Because the FTP Client as well as the FTP Server typically reads the FTP.DATA file, FTP users may require *read* access to it.

- *(IFTP0080: CAT II) The IAO will ensure that if present, the data set containing the FTP.DATA configuration file allows read access to all authenticated users and all other access is restricted to systems programming personnel.*

- *(IFTP0080: CAT II) The IAO will ensure that all write and allocate access to the data set containing the FTP.DATA configuration file is logged.*

- *(IFTP0080: CAT II) The IAO will ensure that if present, the data set containing the FTP banner file allows read access to all authenticated users and all other access is restricted to systems programming personnel.*

The following additions (in bold) to the SYS3 rule set can be used as a base to secure the MVS data sets:

```
$KEY(SYS3)
…
FTP.BANNER UID(sysprog-UID) READ(A) WRITE(A) ALLOC(A) EXEC(A)
FTP.BANNER UID(-) READ(A)
FTP.DATA UID(sysprog-UID) READ(A) WRITE(L) ALLOC(L) EXEC(A)
FTP.DATA UID(-) READ(A)
…
```

4.4.4.4 RACF

This section describes the commands needed to implement the security guidelines for the FTP Server under the RACF ACP. The following task categories are described:

- Started task definitions
- Data set protection

4.4.4.4.1 Started Tasks

The FTP Server component requires the definition of one started task for the FTP daemon.

- *(IFTP0010: CAT II) The IAO will ensure that the user account used for the FTP daemon is defined with the following characteristics:*

  - *Named FTPD*
  - *Privilege to run as a started task*
  - *OS/390 UNIX attributes: UID(0), home directory '/', shell program /bin/sh*
  - *Access to the BPX.DAEMON resource*

The following commands can be used to create the user account and assign the privileges that are required for the FTP daemon:

```
ADDUSER FTPD DFLTGRP(STCTCPX) OWNER(ADMIN) -
    NOPASSWORD NOOIDCARD -
    OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
RDEFINE STARTED FTPD.* UACC(NONE) OWNER(ADMIN) -
    STDATA(USER(FTPD) GROUP(STCTCPX) TRUSTED(NO))
PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(FTPD)
```

4.4.4.4.2 Data Sets

The vendor elements of the FTP Server are installed in data sets that are functionally owned by the Base TCP/IP System component. Please refer to *Section 4.4.1, Base TCP/IP System*, for the security requirements for those data sets.

There may be one or more FTP Server data sets that are used to hold site-customized data. [This data could reside in HFS files rather than data sets.] This includes the FTP.DATA configuration file and the file used for the warning banner. Because the FTP Client as well as the FTP Server typically reads the FTP.DATA file, FTP users may require *read* access to it.

- *(IFTP0080: CAT II) The IAO will ensure that if present, the data set containing the FTP.DATA configuration file allows read access to all authenticated users and all other access is restricted to systems programming personnel.*

- *(IFTP0080: CAT II) The IAO will ensure that all update and alter access to the data set containing the FTP.DATA configuration file is logged.*

- *(IFTP0080: CAT II) The IAO will ensure that if present, the data set containing the FTP banner file allows read access to all authenticated users and all other access is restricted to systems programming personnel.*

The following commands can be used to provide the required access control for the MVS data sets:

```
ADDSD ‘SYS3.FTP.BANNER’ OWNER(SYS1) UACC(NONE)
PERMIT ‘SYS3.FTP.BANNER’ ACCESS(ALTER) ID(sysprog-group)
PERMIT ‘SYS3.FTP.BANNER’ ACCESS(READ) ID(*)
ADDSD ‘SYS3.FTP.DATA’ OWNER(SYS1) UACC(NONE) -
    AUDIT(ALL(UPDATE))
PERMIT ‘SYS3.FTP. DATA’ ACCESS(ALTER) ID(sysprog-group)
PERMIT ‘SYS3.FTP.DATA’ ACCESS(READ) ID(*)
```

4.4.4.5 TOP SECRET

This section describes the commands needed to implement the security guidelines for the FTP Server under the TOP SECRET ACP. The following task categories are described:

- Started task definitions
- Data set protection

4.4.4.5.1 Started Tasks

The FTP Server component requires the definition of one started task for the FTP daemon.

- *(IFTP0010: CAT II) The IAO will ensure that the user account used for the FTP daemon is defined with the following characteristics:*

  - Named FTPD
  - Privilege to run as a started task
  - OS/390 UNIX attributes: UID(0), home directory ‘/’, shell program /bin/sh
  - Access to the BPX.DAEMON resource

The following commands can be used to create the user account and assign the privileges that are required for the FTP daemon:

```
TSS CREATE(FTPD) TYPE(USER) NAME(FTPD)
    DEPT(existing-dept) FACILITY(STC) PASSWORD(password,0)
TSS ADD(FTPD) DFLTGRP(STCTCPX) GROUP(STCTCPX)
TSS ADD(FTPD) SOURCE(INTRDR)
TSS ADD(FTPD) UID(0) HOME(/) OMVSPGM(/bin/sh)
TSS ADD(FTPD) MASTFAC(TCP)
TSS ADD(STC) PROCNAME(FTPD) ACID(FTPD)
TSS PERMIT(FTPD) IBMFAC(BPX.DAEMON) ACCESS(READ)
```

4.4.4.5.2 Data Sets

The vendor elements of the FTP Server are installed in data sets that are functionally owned by the Base TCP/IP System component. Please refer to *Section 4.4.1*, *Base TCP/IP System*, for the security requirements for those data sets.

There may be one or more FTP Server data sets that are used to hold site-customized data. [This data could reside in HFS files rather than data sets.] This includes the FTP.DATA configuration file and the file used for the warning banner. Because the FTP Client as well as the FTP Server typically reads the FTP.DATA file, FTP users may require *read* access to it.

- *(IFTP0080: CAT II) The IAO will ensure that if present, the data set containing the FTP.DATA configuration file allows read access to all authenticated users and all other access is restricted to systems programming personnel.*

- *(IFTP0080: CAT II) The IAO will ensure that all update, create, and scratch access to the data set containing the FTP.DATA configuration file is logged.*

- *(IFTP0080: CAT II) The IAO will ensure that if present, the data set containing the FTP banner file allows read access to all authenticated users and all other access is restricted to systems programming personnel.*

The following commands can be used to provide the required access control for the MVS data sets:

```
TSS ADD(SYS1) DSN(SYS3.FTP.BANNER)
TSS PERMIT(sysprog-group) DSN(SYS3.FTP.BANNER) ACCESS(ALL)
TSS PERMIT(ALL) DSN(SYS3.FTP.BANNER) ACCESS(READ)
TSS ADD(SYS1) DSN(SYS3.FTP.DATA)
TSS PERMIT(sysprog-group) DSN(SYS3.FTP.DATA) ACCESS(ALL)
    ACTION(AUDIT)
TSS PERMIT(ALL) DSN(SYS3.FTP.DATA) ACCESS(READ)
```

4.4.5 FTP Client

The File Transfer Protocol (FTP) Client component of IBM’s Communications Server provides the client portion of the client \ server application for transferring files between hosts. Because the client implements the industry standard File Transfer Protocol on OS/390, it can connect to any other host with a standards-compliant server.

4.4.5.1 General Considerations

The FTP Client can be executed in three environments—batch, TSO, and OS/390 UNIX. A significant security issue arises from the fact that passwords may be embedded in files used in the FTP connection process. This issue is common across these environments, but there are some implementation differences. The next section addresses this issue in these environments.

A consideration that is outside the specific scope of the FTP Client environment is the potential vulnerability in the network. When userids and passwords are sent in the FTP connection process, this information passes in clear text through the network. The OS/390 FTP Client does not provide an alternative to this situation. Encrypted network connections such as those available through a Virtual Private Network (VPN) connection should be used whenever practical. In a future OS release (z/OS Version 1, Release 2), IBM addresses this situation by way of support for SSL-enabled FTP connections.

4.4.5.1.1 Input Commands and NETRC

When the FTP Client connects to a server, the client is required to supply a userid and password for an account on the server. In instances where the connection process is automated, such as a batch job or script, there are two methods for supplying a password: the input command stream and a NETRC file.

Passwords can be included in the stream of FTP commands being sent from the client to the server. Userid and password can be supplied on a single statement or on two consecutive statements. The different environments support this as follows:

In batch jobs the FTP commands are supplied through the INPUT DD statement. The DD statement can refer to a data set or in-stream data (i.e., “DD *”).

In TSO there are two alternatives for supplying FTP commands. As with batch jobs, an INPUT DD statement can refer to a data set. It is also possible, using the REXX facility, to queue the FTP commands to the input stack.

In the OS/390 UNIX environment, there are also multiple alternatives. Standard input to the FTP command can be redirected from a file. It is also possible, using the REXX facility, to queue the FTP commands to the input stack.

A particular concern with the use of in-stream data in batch jobs is related to products such as SDSF, which provide access to JES SPOOL data. If not effectively secured, these products could allow in-stream data to be displayed by users other than the job’s owner.

The following guidelines apply to embedding passwords in FTP command streams:

- *(IFTP0090: CAT II) The IAO will ensure that Userid and password is coded on separate statements to prevent the display of the password in the output file.*

- *(IFTP0100: CAT II) The IAO will ensure that For batch jobs, the INPUT DD statement does not refer to in-stream data (i.e., “DD *”) if that data contains a password.*

- *(IFTP0110: CAT II) The IAO will ensure that any data set or HFS file that contains a password has all access restricted to the owner of the data set or file and, if different, the account under which any associated batch jobs are run.*

A second way to supply passwords for FTP connections is a NETRC file. The file can contain MACHINE, LOGIN, and PASSWORD keywords and associated values that are automatically sent to the server. Multiple sets of keywords and values can be coded on statements in the same file. The different environments support a NETRC file as follows:

In batch jobs the NETRC file is supplied through the NETRC DD statement.

In TSO, a NETRC file can be created as *userid*.NETRC. The data set is automatically used when the FTP command is invoked.

In the OS/390 UNIX environment, a NETRC file can be created as /$HOME/.netrc. The file is automatically used when the ftp command is invoked.

The following guidelines apply to the use of NETRC files:

- *(IFTP0120: CAT II) The IAO will ensure that any NETRC file that contains a password has all access restricted to the owner of the NETRC file and, if different, the account under which any associated batch jobs are run.*

Refer to IBM's *OS/390 IBM Communications Server IP User's Guide* (OS/390 Release 2.10) or *OS/390 SecureWay Communications Server IP User's Guide* (OS/390 Release 2.8) document for details on NETRC statements.

4.4.5.2 HFS Object Protection

The vendor elements of the FTP Client are installed in HFS files that are functionally owned by the Base TCP/IP System component. Please refer to *Section 4.4.1*, *Base TCP/IP System*, for the security requirements for those HFS files.

4.4.5.3 ACF2

There is no need to create special user accounts for the FTP Client. Any user account that has been set up to use OS/390 UNIX (i.e., has a UID value assigned) can access the Client command.

User accounts that are dedicated to FTP use are subject to special restrictions. Please refer to *Section 3.1.2.7*, *FTP Userids*, and *Section 3.2.2.7*, *FTP Userids*, for requirements for those accounts.

The vendor elements of the FTP Client are installed in data sets that are functionally owned by the Base TCP/IP System component. Please refer to *Section 4.4.1*, *Base TCP/IP System*, for the security requirements for those data sets.

4.4.5.4 RACF

There is no need to create special user accounts for the FTP Client. Any user account that has been set up to use OS/390 UNIX (i.e., has a UID value assigned) can access the Client command.

User accounts that are dedicated to FTP use are subject to special restrictions. Please refer to *Section 3.1.2.7, FTP Userids*, and *Section 3.3.2.7, FTP Userids*, for requirements for those accounts.

The vendor elements of the FTP Client are installed in data sets that are functionally owned by the Base TCP/IP System component. Please refer to *Section 4.4.1, Base TCP/IP System*, for the security requirements for those data sets.

4.4.5.5 TOP SECRET

There is no need to create special user accounts for the FTP Client. Any user account that has been set up to use OS/390 UNIX (i.e., has a UID value assigned) can access the Client command.

User accounts that are dedicated to FTP use are subject to special restrictions. Please refer to *Section 3.1.2.7, FTP Userids*, and *Section 3.4.2.7*, *FTP Userids*, for requirements for those accounts.

The vendor elements of the FTP Client are installed in data sets that are functionally owned by the Base TCP/IP System component. Please refer to *Section 4.4.1*, *Base TCP/IP System*, for the security requirements for those data sets.

4.4.6 TFTP Server

The Trivial File Transfer Protocol (TFTP) Server, known as tftpd, supports file transfer according to the industry standard Trivial File Transfer Protocol. Compared to the well-known FTP protocol, TFTP is very limited. It only supports reading or writing a file from or to a server.

4.4.6.1 General Considerations

The TFTP server does not perform any user identification or authentication. Any client that can connect to the IP port (typically 69) on which the server listens can use the server. In addition, unless the Startup command explicitly specifies individual directories, the server can potentially access all HFS files (with the other permission bits on) in all mounted file systems.

The lack of identification and authentication facilities makes the TFTP server unacceptable for systems with security requirements that include discretionary access control and access audit trails.

- *(IFTP0090: CAT II) The IAO will ensure that the TFTP server is not used and the program, TFTPD, is protected as a sensitive utility.*

4.4.6.2 HFS Object Protection

As long as the ACP-specific steps in the following paragraphs are taken, no security changes to the HFS objects of the TFTP server component are required.

4.4.6.3 ACF2

The following commands can be used to protect the TFTP server program:

```
SET CONTROL(GSO)
CHANGE PPGM PGM-MASK(TFTPD  EZATRD) ADD
```

NOTE: TFTPD is an alias for EZATD so entries are required for both names.

4.4.6.4 RACF

The following commands can be used to protect the TFTP server program:

```
RDEFINE PROGRAM TFTPD ADDMEM(‘SYS1.TCPIP.SEZALINK’) -
    UACC(NONE) OWNER(ADMIN)
RDEFINE PROGRAM EZATD ADDMEM(‘SYS1.TCPIP.SEZALINK’) -
    UACC(NONE) OWNER(ADMIN)
```

NOTE: TFTPD is an alias for EZATD so entries are required for both names.

4.4.6.5 TOP SECRET

The following commands can be used to protect the TFTP server program:

```
TSS ADD(ADMIN) PROGRAM(TFTPD,EZATD)
```

NOTE: TFTPD is an alias for EZATD so entries are required for both names.

4.4.7 Syslog Daemon

The Syslog daemon, known as syslogd, is an OS/390 UNIX daemon that provides a central processing point for log messages issued by other OS/390 UNIX processes. It is also possible to receive log messages from other network-connected hosts. Some of the IBM Communications Server components that may send messages to syslog are the FTP, TFTP, OS/390 UNIX Telnet, DNS, and DHCP servers. The messages may be of varying importance levels including general process information, diagnostic information, critical error notification, and audit-class information. Primarily because of the potential to use this information in an audit process, there is a security interest in protecting the syslogd process and its associated data.

The following sections provide guidance for securing the configuration and files that make up the syslogd component. This includes startup timing, address space control, file security, required user accounts, and ACP-defined resources.

4.4.7.1 General Considerations

There are several configuration issues addressed relative to the security environment for syslogd:

- When it is started
- The user account associated with the OS/390 address space
- The name associated with the OS/390 address space
- The security of the output files

It is important that syslogd be started during the initialization phase of the OS/390 system to ensure that significant messages are not lost. As with other OS/390 UNIX daemons, there is more than one way to start syslogd. It can be started as a process in the /etc/rc file or as an OS/390 started task.

- *(ISLG0010: CAT II) The systems programmer responsible for supporting ICS will ensure that Syslogd is started at OS/390 system initialization.*

Security is enhanced when syslogd executes in an address space associated with its own ACP user account. This ensures that the address space has no more, and no less, than the specific privileges required. When started in /etc/rc, the command export _BPX_USERID=SYSLOGD can be used to assign the specific user account.

- *(ISLG0020: CAT II) The systems programmer responsible for supporting ICS will ensure that Syslogd runs under its own user account. Specifically, it does not share the account defined for the OS/390 UNIX kernel.*

The ability to control the syslogd address space is impacted by the name it is assigned. When started in /etc/rc, the command export _BPX_JOBNAME='SYSLOGD' can be used to assign the specific job name.

- *(ISLG0020: CAT II) The systems programmer responsible for supporting ICS will ensure that Syslogd runs with a job/started task name such as SYSLOGD that uniquely identifies it.*

The output of syslogd depends on the options selected in the configuration file. This file is customarily named /etc/syslog.conf, but the name can be overridden in the startup parameters of the daemon. Typically, the daemon writes the messages to one of a series of log files. Different files are used to distinguish the source of the message and its priority level. Although the site is free to design their own log file structure, the following should be considered:

Messages with a priority code of critical (crit) or higher should be written to the OS/390 operator console (via /dev/console) in addition to any permanent file.

The use of the OS/390 operator console, a specific user, or (as of OS/390 Release 2.10) SMF as a destination should be carefully considered in light of the potentially high volume of messages.

The use of another host as a destination should be limited to hosts within the immediate security enclave.

Log files should have security that prevents anyone except the syslogd process and authorized maintenance jobs from writing to or deleting them. A requirement to address this is documented in a following section. Log files should be pre-allocated (i.e., any daemon option to create the files automatically should not be used).

A maintenance process to periodically clear the log files is essential. Logging stops if the target file system becomes full.

Refer to IBM's *OS/390 IBM Communications Server IP Configuration Reference Guide* (OS/390 Release 2.10) or *OS/390 SecureWay Communications Server IP Configuration* (OS/390 Release 2.8) document for details on the syslogd configuration statements.

4.4.7.2 HFS Object Protection

In order to protect the Syslog daemon component, special security settings will be applied to selected HFS files.

- *(ISLG0030: CAT II) The systems programmer responsible for supporting ICS will ensure that the permission bits and user audit bits for HFS objects that are part of the Syslog daemon component are configured according to the settings in the following table:*

Table A-54. SYSLOG DAEMON HFS OBJECT SECURITY SETTINGS (4.4.7.2 a)

SYSLOG DAEMON HFS OBJECT SECURITY SETTINGS			
DIRECTORY or FILE	PERMISSION BITS	USER AUDIT BITS	FUNCTION
/usr/sbin/syslogd	1740	fff	Daemon program
/etc/syslog.conf	0744	faf	Daemon configuration file
[Output log files defined in the configuration file.]	0744	fff	Log files

NOTE: The /usr/sbin/syslogd object is a symbolic link to /usr/lpp/tcpip/sbin/syslogd. The permission and user audit bits on the target of the symbolic link will have the required settings.

NOTE: The /etc/syslog.conf file may not be the configuration file the daemon uses. It is necessary to check the script or JCL used to start the daemon to determine the actual configuration file.

The following commands can be used (from a user account with an effective UID(0)) to update the permission bits and audit bits:

```
chmod  1740  /usr/lpp/tcpip/sbin/syslogd
chaudit  rwx=f  /usr/lpp/tcpip/sbin/syslogd
chmod  0744  /etc/syslog.conf
chaudit  w=sf,rx+f  /etc/syslog.conf
chmod  0744  /log_dir/log_file
chaudit  rwx=f  /log_dir/log_file
```

4.4.7.3 ACF2

The Syslog daemon component requires the definition of a user account for the daemon.

- *(ISLG0020: CAT II) The IAO will ensure that the user account used for the Syslog daemon is defined with the following characteristics:*

  - Named SYSLOGD
  - Privilege to run as a started task
  - OS/390 UNIX attributes: UID(0), home directory '/', shell program /bin/sh
  - Access to the BPX.DAEMON resource.

The following commands can be used to create the user account that is required for the Syslog daemon:

- SET LID
- INSERT SYSLOGD NAME(SYSLOGD) GROUP(STCTCPX) STC
- SET PROFILE(USER) DIVISION(OMVS)
- INSERT SYSLOGD UID(0) HOME(/) PROGRAM(/bin/sh)

NOTE: At eTrust CA-ACF2 6.4 and above, the PROGRAM field in the user profile record has been renamed to OMVSPGM.

The following rule set additions (in bold) can be used to assign the privileges that are required for the Syslog daemon:

```
$KEY(BPX) TYPE(FAC)
…
DAEMON UID(SYSLOGD-uid) SERVICE(READ) ALLOW
…
SMF UID(SYSLOGD-uid) SERVICE(READ) ALLOW
…
```

NOTE: Access to the BPX.SMF resource is only required when the syslogd configuration file specifies $SMF as a destination (available as of OS/390 Release 2.10).

The following operator command is required to complete the update:

```
F ACF2,REBUILD(FAC)
```

4.4.7.4 RACF

The Syslog daemon component requires the definition of a user account for the daemon.

- *(ISLG0020: CAT II) The IAO will ensure that the user account used for the Syslog daemon is defined with the following characteristics:*

  - Named SYSLOGD
  - Privilege to run as a started task
  - OS/390 UNIX attributes: UID(0), home directory '/', shell program /bin/sh
  - Access to the BPX.DAEMON resource

The following commands can be used to create the user account and assign the privileges that are required for the Syslog daemon:

```
ADDUSER SYSLOGD DFLTGRP(STCTCPX) OWNER(ADMIN) -
    NOPASSWORD NOOIDCARD -
    OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
RDEFINE STARTED SYSLOGD.* UACC(NONE) OWNER(ADMIN) -
    STDATA(USER(SYSLOGD) GROUP(STCTCPX) TRUSTED(NO))
PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(SYSLOGD)
PERMIT BPX.SMF CLASS(FACILITY) ACCESS(READ) ID(SYSLOGD)
```

NOTE: Access to the BPX.SMF resource is only required when the syslogd configuration file specifies $SMF as a destination (available as of OS/390 Release 2.10).

4.4.7.5 TOP SECRET

The Syslog daemon component requires the definition of a user account for the daemon.

- *(ISLG0020: CAT II) The IAO will ensure that the user account used for the Syslog daemon is defined with the following characteristics:*

  - Named SYSLOGD
  - Privilege to run as a started task
  - OS/390 UNIX attributes: UID(0), home directory '/', shell program /bin/sh
  - Access to the BPX.DAEMON resource

The following commands can be used to create the user account and assign the privileges that are required for the Syslog daemon:

```
TSS CREATE(SYSLOGD) TYPE(USER) NAME(SYSLOGD)
    DEPT(existing-dept) FACILITY(STC) PASSWORD(password,0)
TSS ADD(SYSLOGD) DFLTGRP(STCTCPX) GROUP(STCTCPX)
TSS ADD(SYSLOGD) SOURCE(INTRDR)
TSS ADD(SYSLOGD) UID(0) HOME(/) OMVSPGM(/bin/sh)
TSS ADD(STC) PROCNAME(SYSLOGD) ACID(SYSLOGD)
TSS PERMIT(SYSLOGD) IBMFAC(BPX.DAEMON) ACCESS(READ)
TSS PERMIT(SYSLOGD) IBMFAC(BPX.SMF) ACCESS(READ)
```

NOTE: Access to the BPX.SMF resource is only required when the syslogd configuration file specifies $SMF as a destination (available as of OS/390 Release 2.10).

This page intentionally left blank.

5. JES2

5.1 General Considerations

OS/390 uses a job entry subsystem (JES) to receive jobs into the operating system, schedule them for processing by MVS, and control their output processing. JES is designed to manage jobs before and after execution, allowing the MVS base control program to manage jobs during execution.

The STIG requirement is to use JES2 as the primary JES. JES2 exercises independent control over its job processing functions. Each JES2 image within the configuration controls its own job input, scheduling, and job output processing functions, such as the following:

- Reads jobs into the system from local and remote sources
- Converts jobs to internal machine-readable form
- Selects jobs for execution
- Processes output from jobs
- Purges jobs from the system
- Manipulates jobs under operator or program control

The JES2 subsystem is subject to various types of potential abuse. For this reason, it is necessary to place restrictions on the facilities that can be used and on the manner of their use. To control access to JES2 facilities and resources, apply the following recommendations when implementing security:

(1) Use the services of an ACP for security control. JES2 internal mechanisms (e.g., initialization statement parameters and installation exits) should not be used for security control.

(2) Collect SMF data for auditing purposes.

(3) JES2 resources should be strictly controlled. Restrict access to those resources necessary for users to accomplish their assigned responsibilities. The resources to be controlled include, but are not limited to, the following:

- JES-owned data sets, including SPOOL, SPOOL off-load, checkpoint, libraries containing executable code, commands, exit routines, cataloged procedures, and initialization parameters

- Input devices including local readers, internal readers, NJE readers, RJE remote workstations, SPOOL off-load receivers, and TSO SUBMIT

- Output devices including local printers, local punching devices, NJE transmissions, RJE remote workstations, and SPOOL off-load devices

- Data residing on the JES2 SPOOL (SYSIN/SYSOUT data sets) including JES News, SYSLOG and JES2 traces

NOTE: It is imperative that the SYSLOG and trace data are secured from unauthorized access. Uncontrolled access could result in a breach in system and data integrity, or a potential security exposure. Access to the SYSLOG and trace data is limited to those personnel authorized by the IAO.

- Commands
- Job submission and naming
- Surrogate user privileges (the ability to submit work on behalf of another)
- Jobs and SYSOUT transmitted to and from other NJE nodes
- Dumps, logs, and traces of JES2 data

(4) If the installation has assigned specific operators responsibility only for a subset of the resources controlled by JES2 (e.g., peripheral equipment), the IAO defines security categories to enable granting those operators access only to the resources they require. This is done by connecting to those groups rather than granting access to each operator individually. Depending on the ACP, this may involve defining additional profiles or may involve naming conventions. Refer to *Section 3.1.5.6, OS/390 System Command Controls*.

(5) Where this section allows the installation to select one of several distinct options, the IAO documents in the installation SOP the choices made and the justification for those choices.

Italicized text in resource names and userids represents variable rather than literal text (e.g., RMT*nnnn* represents the letters RMT followed by a four-digit remote number). *JES2* represents the name of the JES2 subsystem, which is normally JES2.

5.1.1 Userids for Remote Processing

JES2 allows remote workstations to submit jobs, control them, and retrieve their output. JES2 refers to these workstations as RJE workstations.

In addition, JES2 allows jobs and commands to be exchanged between OS/390 systems. JES2 refers to such systems as NJE nodes.

RJE and NJE present the same security issues as any other JES2 I/O device. Entry and output of jobs and commands will be controlled. Apply the following recommendations when implementing security for JES2 RJE and NJE:

(1) Define userid RMT*nnnn* for each RJE workstation, where *nnnn* is the number on the RMT statement or $ADD RMT command. Do not define any profile segments or grant any access rights except as specified in this section. The password controls in *Section 3.1.3, Password Controls*, will apply in full.

- *(ZJES0011: CAT II) The IAO will ensure that RJE workstations have valid logonids with no user privileges and no access to datasets and resources.*

(2) Define userid *nodename* for each NJE node, where *nodename* is the name on the NODE statement or $ADD APPL command. Do not define any profile segments or grant any

access rights except as specified in this section. The password controls in *Section 3.1.3, Password Controls*, will apply in full.

- *(ZJES0012: CAT II) The IAO will ensure that NJE workstations have valid logonids with no user privileges and no access to datasets and resources.*

5.1.2 Security Controls for Input

The JESINPUT class is provided by IBM to control the source of submission for jobs. Optionally, ACF2 and TOP SECRET have their own mechanism to control the source for NJE and RJE submitted jobs. These controls may be used instead of the JESINPUT class. Refer to *Section 3.2.2.4, Network Users*, and *Section 3.4.4.4, Other Sensitive Privileges*, for further information.

This section describes the method to protect the JES2 input resources using the JESINPUT class, as well as the minimal protections to be applied. Additional security may be utilized at the discretion of the IAO.

(1) Define the following with a default access of *none*:

JESINPUT class
INTRDR
nodename
OFF*n*.*
OFF*n*.JR
OFF*n*.SR
R*nnnn*.RD*m*
RDR*nn*
STCINRDR
TSUINRDR

The default access should be *none* except for sources that are permitted to submit jobs for all users. Those sources may be defined as either *none* or *read*.

(2) Grant *read* access to authorized users for each of the following input sources:

INTRDR
nodename
OFF*n*.*
OFF*n*.JR
OFF*n*.SR
R*nnnn*.RD*m*
RDR*nn*
STCINRDR
TSUINRDR

The resource definition will be generic if all of the resources of the same type have identical access controls (e.g., if all off-load receivers are equivalent). The default access will be *none* except for sources that are permitted to submit jobs for all users. Those sources may be defined as either *none* or *read*.

- *(ZJES0021: CAT II) The IAO will ensure that the JESINPUT resource class is defined and required resource(s) is (are) defined to the JESINPUT resource class with no access.*

- *(ZJES0022: CAT II) The IAO will ensure that access authorization for resources defined to the JESINPUT resource class is restricted to the appropriate personnel.*

5.1.3 Security Controls for Output

This section describes the method to protect the JES2 output resources and the minimal protection to be applied. Additional security may be utilized at the discretion of the IAO.

(1) Define the following with a default access of *none*:

WRITER class
*JES2.***

(2) Define resources in the ACP's respective WRITER class for each of the following output destinations:

JES2.LOCAL.*devicename*
JES2.LOCAL.OFF*n*.*
JES2.LOCAL.OFF*n*.JT
JES2.LOCAL.OFF*n*.ST
JES2.LOCAL.PRT*n*
JES2.LOCAL.PUN*n*
JES2.NJE.*nodename*
JES2.RJE.*devicename*

The resource definition will be generic if all of the resources of the same type have identical access controls (e.g., if all off-load transmitters are equivalent). If all users are permitted to route output to a specific destination, the resource controlling it may be defined with a default access of either *none* or *read.* Otherwise it will be defined with a default access of *none*.

(3) If IBM's SDSF product is installed on the system, resources defined to the WRITER resource class control functions related to printers and punches on various SDSF panels. Refer to *Section 14.2.1.7.12, Printers and Punches*, for additional security requirements.

- *(ZJES0031: CAT II) The IAO will ensure that the WRITER resource class is defined and required resource(s) is (are) defined to the WRITER resource class with no access.*

- *(ZJES0032: CAT II) The IAO will ensure that access authorization for resources defined to the WRITER resource class is restricted to the appropriate personnel.*

5.1.4 Security Controls for JES2 SPOOL Data Sets

The following section defines the minimum security controls for the JES2 SPOOL data sets. The SPOOL may have more restrictive security at the direction of the IAO.

(1) Define the following resources in the ACP's respective JESSPOOL class with a default access of *none*:

*localnodeid.***
localnodeid.*JES2*.$TRCLOG.*taskid*.*.JESTRACE
localnodeid.+MASTER+.SYSLOG.*jobid*.*.SYSLOG

(2) Define the following resource in the JESSPOOL class with a default access of *read*:

localnodeid.*jesid*.$JESNEWS.*taskid*.D*newslvl*.JESNEWS

(3) Define a resource in the OPERCMDS class for *JES2*.UPDATE.JESNEWS with a default access of *none*. Permit *control* access for those users responsible for maintaining the JES News data set. All access will be logged.

(4) By default a user has access only to that user's own jobs. However, situations exist where one user legitimately requires access to jobs that run under another user's userid. In particular, if a user routes SYSOUT to an external writer, the external writer should have access to that user's SYSOUT. With the approval of the IAO, the installation may grant a user *read* access to *localnodeid*.*userid*.*jobname*.*jobid*.*dsnumber*.*name* in the JESSPOOL class. All such accesses will be logged.

If frequent situations occur where users working on a common project require selective access to each other's jobs, then the installation may delegate to the individual users the authority to grant access, but only with the approval of the IAO.

If IBM's SDSF product is installed on the system, resources defined to the JESSPOOL resource class control functions related to jobs, output groups, and SYSIN/SYSOUT data sets on various SDSF panels. Refer to *Section 14.2.1.7.24, Jobs, Output Groups, and SYSIN/SYSOUT Data Sets*, for additional security requirements.

(5) If the JES2 trace and SYSLOG data sets are to be transcribed by external writers, grant the userids *read* access to the following:

localnodeid.*JES2*.$TRCLOG.*taskid*.*.JESTRACE
localnodeid.+MASTER+.SYSLOG.*jobid*.*.SYSLOG

This access will be strictly limited to the absolutely minimum number of external writers.

(6) Grant access to the following to systems personnel responsible for diagnosing JES2 problems:

localnodeid.*JES2*.$TRCLOG.*taskid*.*.JESTRACE
localnodeid.+MASTER+.SYSLOG.*jobid*.*.SYSLOG

This access will be strictly limited to the absolutely minimum number of necessary personnel.

(7) Grant access to the following to systems personnel responsible for diagnosing MVS problems:

localnodeid.+MASTER+.SYSLOG.*jobid*.*.SYSLOG

This access will be strictly limited to the absolutely minimum number of necessary personnel.

- *(ZJES0041: CAT II) The IAO will ensure that the JESSPOOL resource class is defined and required resource(s) is (are) defined to the JESSPOOL resource class.*

- *(ZJES0042: CAT II) The IAO will ensure that access authorization for the JESNEWS resource in the OPERCMDS resource class is restricted to the appropriate personnel and , and all access is logged.*

- *(ZJES0044: CAT II) The IAO will ensure that access authorization for resources defined to the JESTRACE and SYSLOG resource in the JESSPOOL resource class is restricted to the appropriate personnel.*

- *(ZJES0046: CAT II) The IAO will ensure that access authorization for resources defined to the SYSIN/SYSOUT spool data set(s) in the JESSPOOL resource class is (are) restricted to the appropriate personnel.*

5.1.5 Security Controls for JES2 Commands

Extended MCS support allows the installation to control the use of JES2 system commands through the ACP. These commands are subject to various types of potential abuse. For this reason, it is necessary to place restrictions on the JES2 system commands that can be entered by particular operators.

Some commands are particularly dangerous and should only be used when less drastic options have been exhausted. Misuse of these commands can create a situation in which the only recovery is an IPL. These commands are referred to as sensitive commands in this section.

To control access to JES2 system commands, apply the following recommendations when implementing security:

(1) Define the *JES2.*** resource in the OPERCMDS class with a default access of *none*.

(2) Define categories of users for the following:

- Network personnel
- Operations personnel (Junior)
- Operations personnel (Senior)
- Systems personnel
- Users without the above responsibilities

Where one of the above includes users with significantly different responsibilities, define as many categories as necessary to give appropriate access to resources at the category level.

(3) Associate each operator with the appropriate security category defined above.

(4) Document in the installation SOP which users have access to which commands, whether that access is logged, and the justification for that access. The documentation of user access should be written in terms of responsibilities and roles rather than individual user names. Where this *STIG* explicitly permits access to particular commands, a reference to this *STIG* is all the justification that is required. Where this *STIG* defers the decision to the IAO, the IAO will consult with Operations and Systems, and the decision should be documented. However, if the IAO, in consultation with Operations and Systems, specifies a more restrictive level of access than specified in this *STIG*, no justification need be given.

As part of this documentation, specify the policies and procedures for the use of sensitive JES2 commands. Restrict access to these commands to the absolutely minimum number of personnel, and log all access. The IAO takes into account the need to issue these commands in emergency situations and for system reconfigurations, shutdowns, and upgrades.

(5) Collect SMF data for specific commands as shown in *Table A-63, Controls on JES2 System Commands*.

(6) If the installation uses JES2 Exit 36 to provide finer granularity than is available with the JES2 resource names in the OPERCMDS class, the IAO will document in the installation SOP the resource class and names used by the exit to control access.

- *(ZJES0056: CAT II) The IAO will ensure that if JES2 Exit 36 is in use the installation SOP documents the resource class and names used by the exit.*

(6) Prepare ACP controls to grant access to commands using *Table A-63, Controls on JES2 System Commands*, as a guideline for each category of users. All required resource logging of JES2 system commands will be performed using the ACP. In general, the commands are controlled by selectively granting access to resources in the OPERCMDS class, with names such as JES2.*command.qualifier*. The permissions granted will be tailored to the users' roles rather than using a wild card for the command object in every permission.

- *(ZJES0051: CAT II) The IAO will ensure that the* JES2.* *resource is defined to the OPERCMDS class with a default of no access and all access is logged.*

(7) If IBM's SDSF product is installed on the system, additional resource access to JES2 system commands is allowed to general users when using SDSF. These resources will be noted in the following table. Refer to *Section 13.2.1.7.8, MVS and JES2 Commands Generated by SDSF*, for additional security requirements.

At the discretion of the IAO, access may be granted to individual users, but the justification will be documented.

- *(ZJES0052: CAT II) The IAO will ensure that access to JES2 system commands listed in the following table are restricted to the appropriate personnel and logged where indicated:*

Table A-55. CONTROLS ON JES2 SYSTEM COMMANDS (5.1.5)

CONTROLS ON JES2 SYSTEM COMMANDS				
COMMAND	RESOURCE(s)	GROUP	AUTH ?	LOG REQ'D
$A A $A J $A S $A T $A JOBQ $A Q $A 'jobname'	*JES2*.MODIFYRELEASE.JOB *JES2*.MODIFYRELEASE.BAT *JES2*.MODIFYRELEASE.STC *JES2*.MODIFYRELEASE.TSU *JES2*.MODIFYRELEASE.JST *JES2*.MODIFYRELEASE.JOB *JES2*.MODIFYRELEASE.JOB	Operations (Jr.) Operations (Sr.)	I Y	Y
$ADD APPL $ADD CONNECT $ADD DESTID $ADD FSS $ADD RMT	*JES2*.ADD.APPL *JES2*.ADD.CONNECT *JES2*.ADD.DESTID *JES2*.ADD.FSS *JES2*.ADD.RMT	Network Operations (Sr.) Systems	S	Y
$ADD PRT $ADD REDIRECT	*JES2*.ADD.PRT *JES2*.ADD.REDIRECT	Operations (Sr.) Systems	I Y	Y
$B device	*JES2*.BACKSP.DEV	Operations (Jr.) Operations (Sr.)	I Y	Y
$C A	*JES2*.CANCEL.AUTOCMD	Operations (Jr.)	I	Y

CONTROLS ON JES2 SYSTEM COMMANDS

COMMAND	RESOURCE(s)	GROUP	AUTH ?	LOG REQ'D
$C J $C Lx.yy $C S $C T $C JOBQ $C 'jobname'	***NOTE:*** The BAT, STC, and TSU resources may be permitted to general users when using SDSF.	Operations (Sr.)	Y	
$C device	*JES2*.CANCEL.DEV ***NOTE*:** The DEV resource may be permitted to general users when using SDSF.	Operations (Jr.)	I	Y
		Operations (Sr.)	Y	
$D A $D ACTRMT $D F $D I $D J $D JOBQ $D 'jobname'	*JES2*.DISPLAY.JOB *JES2*.DISPLAY.ACTRMT *JES2*.DISPLAY.QUE *JES2*.DISPLAY.INITIATOR *JES2*.DISPLAY.BAT *JES2*.DISPLAY.JST *JES2*.DISPLAY.JOB	All	Y	N
$D M	*JES2*.SEND.MESSAGE	Operations	Y	Y
$D N $D PCE $D PRT $D Q $D REBLD $D S $D SPOOL $D T $D TRACE(x) $D U $D init stmt	*JES2*.DISPLAY.JOB *JES2*.DISPLAY.PCE *JES2*.DISPLAY.DEV *JES2*.DISPLAY.JOB *JES2*.DISPLAY.REBLD *JES2*.DISPLAY.STC *JES2*.DISPLAY.SPOOL *JES2*.DISPLAY.TSU *JES2*.DISPLAY.TRACE *JES2*.DISPLAY.DEV *JES2*.DISPLAY.*initstmt*	All	Y	N
$DEL CONNECT	*JES2*.DEL.CONNECT	Network	S	Y
$DEL DESTID	*JES2*.DEL.DESTID	Operations (Sr.)		
$E CKPTLOCK	*JES2*.RESTART.SYS	Systems		
$E J	*JES2*.RESTART.BAT	Operations (Jr.)	N	Y
$E 'jobname'	*JES2*.RESTART.BAT ***NOTE:*** The BAT resource may be permitted to general users when using SDSF.	Operations (Sr.)	I	
$E Lx.yyy	*JES2*.RESTART.DEV	Operations (Jr.)	I	Y

CONTROLS ON JES2 SYSTEM COMMANDS				
COMMAND	RESOURCE(s)	GROUP	AUTH ?	LOG REQ'D
$E LINE $E LOGON	***NOTE:*** The DEV resource may be permitted to general users when using SDSF.	Operations (Sr.)	Y	
$E MEMBER(x)	*JES2*.RESTART.SYS	Operations (Sr.)	S	Y
		Systems	Y	
$E device	*JES2*.RESTART.DEV	Operations (Jr.)	I	Y
$F device	*JES2*.FORWARD.DEV	Operations (Sr.)	Y	
$G A	*JES2*.GMODIFYRELEASE.JOB	Operations (Jr.)	I	Y
$G C	*JES2*.GCANCEL.JOB	Operations (Sr.)	Y	
$G D	*JES2*.GDISPLAY.JOB	Operations	Y	N
$G H	*JES2*.GMODIFYHOLD.JOB	Operations (Jr.)	I	Y
$G R	*JES2*.GROUTE.JOBOUT	Operations (Sr.)	Y	
$H A	*JES2*.MODIFYHOLD.JOB	Operations (Jr.)	I	Y
$H J	*JES2*.MODIFYHOLD.BAT	Operations (Sr.)	Y	
$H JOBQ	*JES2*.MODIFYHOLD.JST			
$H Q	*JES2*.MODIFYHOLD.JOB			
$H S	*JES2*.MODIFYHOLD.STC			
$H T	*JES2*.MODIFYHOLD.TSU			
$H 'jobname'	*JES2*.MODIFYHOLD.JOB NOTE: The BAT, STC, and TSU resources may be permitted to general users when using SDSF.			
$I device	*JES2*.INTERRUPT.DEV	Operations (Jr.)	I	Y
		Operations (Sr.)	Y	
$L J	*JES2*.DISPLAY.BATOUT	All	Y	N
$L JOBQ	*JES2*.DISPLAY.JSTOUT			
$L S	*JES2*.DISPLAY.STCOUT			
$L T	*JES2*.DISPLAY.TSUOUT			
$L 'jobname'	*JES2*.DISPLAY.JOBOUT			
$M	*JES2*.MSEND.CMD	Operations	Y	Y
$N	*JES2*.NSEND.CMD ***NOTE***: The MSEND.CMD resource may be permitted to general users when using SDSF.			
$N device	*JES2*.REPEAT.DEV	Operations (Jr.)	I	Y
		Operations (Sr.)	Y	
$O J	*JES2*.RELEASE.BATOUT	Operations (Jr.)	I	Y

CONTROLS ON JES2 SYSTEM COMMANDS				
COMMAND	RESOURCE(s)	GROUP	AUTH ?	LOG REQ'D
$O JOBQ $O Q $O S $O T $O 'jobname'	STCOUT, and TSUOUT resources may be permitted to general users when using SDSF.	Operations (Sr.)	Y	
$P	*JES2*.STOP.SYS	Operations (Sr.)	Y	Y
		Systems	I	
$P I $P J $P JOBQ $P Lx.yyy $P LINE(x) $P LOGON $P Q $P RMT $P S $P SPOOL $P T $P TRACE(x) $P 'jobname'	*JES2*.STOP.INITIATOR *JES2*.STOP.BAT *JES2*.STOP.JST *JES2*.STOP.DEV *JES2*.STOP.LINE *JES2*.STOP.LOGON *JES2*.STOP.JOBOUT *JES2*.STOP.RMT *JES2*.STOP.STC *JES2*.STOP.SPOOL *JES2*.STOP.TSU *JES2*.STOP.TRACE *JES2*.STOP.JOB	Operations (Jr.)	I	Y
		Operations (Sr.)	Y	
$P device	*JES2*.STOP.DEV	Operations (Jr.)	I	Y
		Operations (Sr.)	Y	
$P JES2	*JES2*.STOP.SYS	Operations (Sr.)	Y	Y
		Systems	I	
$R ALL $R PRT $R PUN $R XEQ	*JES2*.ROUTE.JOBOUT *JES2*.ROUTE.JOBOUT *JES2*.ROUTE.JOBOUT *JES2*.ROUTE.JOBOUT ***NOTE***: The JOBOUT resource may be permitted to general users when using SDSF.	Operations (Jr.)	I	Y
		Operations (Sr.)	Y	
$S	*JES2*.START.SYS	Operations	Y	Y
$S A	*JES2*.START.AUTOCMD	Operations (Jr.)	I	Y

CONTROLS ON JES2 SYSTEM COMMANDS				
COMMAND	RESOURCE(s)	GROUP	AUTH ?	LOG REQ'D
$S I		Operations (Sr.)	Y	
$S Lx.yyy				
$S LINE(x)				
$S LOGON(x)				
$S N				
$S SPOOL				
$S RMT(x)				
$S TRACE(x)				
$S device				
$T A	*JES2*.MODIFY.AUTOCMD	Operations (Jr.)	I	Y
		Operations (Sr.)	Y	
$T ALL	*JES2*.MODIFY.SYS	Operations	Y	Y
$T APPL	*JES2*.MODIFY.APPL	Network	Y	Y
		Operations (Sr.)	I	
		Systems	Y	
$T I	*JES2*.MODIFY.INITIATOR	Operations (Jr.)	I	Y
$T J	*JES2*.MODIFY.BAT	Operations (Sr.)	Y	
$T JOBQ	*JES2*.MODIFY.JST			
$T 'jobname'	*JES2*.MODIFY.JOB			
$T LINE	*JES2*.MODIFY.LINE			
$T LOGON	*JES2*.MODIFY.LOGON			
$T MEMBER	*JES2*.MODIFY.SYS			
$T NUM	*JES2*.MODIFY.NUM			
$T O J	*JES2*.MODIFY.BATOUT			
$T O JOBQ	*JES2*.MODIFY.JSTOUT			
$T O S	*JES2*.MODIFY.STCOUT			
$T O T	*JES2*.MODIFY.TSUOUT			
$T O 'jobname'	*JES2*.MODIFY.JOBOUT			
$T OFFx.yy	*JES2*.MODIFY.OFF			
$T OFFLOAD	*JES2*.MODIFY.OFFLOAD			
$T RMT	*JES2*.MODIFY.RMT			
$T S	*JES2*.MODIFY.STC			
$T T	*JES2*.MODIFY.TSU			
$T device	*JES2*.MODIFY.DEV			
	NOTE: The BATOUT, TSUOUT, and STCOUT resources may be permitted to general users when using SDSF.			
$T NODE	*JES2*.MODIFY.NODE	Network	S	Y
$T SSI	*JES2*.MODIFY.SSI	Operations (Sr.)	S	

CONTROLS ON JES2 SYSTEM COMMANDS				
COMMAND	RESOURCE(s)	GROUP	AUTH ?	LOG REQ’D
$T init stmt	*JES2*.MODIFY.*initstmt*	Systems	S	
$VS	*JES2*.VS	Operations	Y	Y
$Z A $Z I $Z device	*JES2*.HALT.AUTOCMD *JES2*.HALT.INITIATOR *JES2*.HALT.DEV	Operations (Jr.)	I	Y
		Operations (Sr.)	Y	
$Z SPOOL	*JES2*.HALT.SPOOL	Operations (Sr.)	S	Y
		Systems	S	
$Z device	*JES2*.HALT.DEV	Operator	Y	Y

In the authorized and logging columns, **Y=Yes**, **N=No**, **R=Read** (lowest) access only, **I=IAO** discretion, and **S=Sensitive** command[20].

(8) Define resources in the OPERCMDS class for each of the following with a default access of *none*:

JES2.DISPLAY.ACTRMT
JES2.DISPLAY.BAT
JES2.DISPLAY.BATOUT
JES2.DISPLAY.DEV
JES2.DISPLAY.INITIATOR
JES2.DISPLAY.JOB
JES2.DISPLAY.JOBOUT
JES2.DISPLAY.JST
JES2.DISPLAY.JSTOUT
JES2.DISPLAY.PCE
JES2.DISPLAY.QUE
JES2.DISPLAY.REBLD
JES2.DISPLAY.SPOOL
JES2.DISPLAY.STC
JES2.DISPLAY.STCOUT
JES2.DISPLAY.TRACE
JES2.DISPLAY.TSU
JES2.DISPLAY.TSUOUT
JES2.DISPLAY.*initstmt*
JES2.GDISPLAY.JOB

These resources control various $D (display) commands that allow the operator to obtain non-sensitive information. Grant *read* access to these resources to any operator who requires it.

[20] As noted above, access will be limited to the minimum number of senior personnel.

(9) Define resources in the OPERCMDS class for each of the following with a default access of *none*:

JES2.MSEND.MESSAGE
JES2.NSEND.MESSAGE
JES2.SEND.MESSAGE

These resources control various commands that allow the operator to send messages to running jobs, commands to other members of a multi-access spool (MAS) configuration, or commands to remote sites. Grant *read* access to these resources for any operator who is authorized to send such commands and messages.

(10) Define resources in the OPERCMDS class for each of the following with a default access of *none*:

JES2.BACKSP.DEV
JES2.CANCEL.AUTOCMD
JES2.CANCEL.BAT
JES2.CANCEL.DEV
JES2.CANCEL.JOB
JES2.CANCEL.JST
JES2.CANCEL.STC
JES2.CANCEL.TSU
JES2.FORWARD.DEV
JES2.GCANCEL.JOB
JES2.GMODIFYHOLD.JOB
JES2.GROUTE.JOBOUT
JES2.HALT.DEV
JES2.INTERRUPT.DEV
JES2.MODIFY.AUTOCMD
JES2.MODIFY.BAT
JES2.MODIFY.BATOUT
JES2.MODIFY.JOB
JES2.MODIFY.JOBOUT
JES2.MODIFY.JST
JES2.MODIFY.JSTOUT
JES2.MODIFY.STC
JES2.MODIFY.STCOUT
JES2.MODIFY.TSU
JES2.MODIFY.TSUOUT
JES2.MODIFYHOLD.BAT
JES2.MODIFYHOLD.JOB
JES2.MODIFYHOLD.JST
JES2.MODIFYHOLD.STC
JES2.MODIFYHOLD.TSU
JES2.MODIFYRELEASE.BAT

JES2.MODIFYRELEASE.JOB
JES2.MODIFYRELEASE.JST
JES2.MODIFYRELEASE.STC
JES2.MODIFYRELEASE.TSU
JES2.RELEASE.BATOUT
JES2.RELEASE.JOBOUT
JES2.RELEASE.JSTOUT
JES2.RELEASE.STCOUT
JES2.RELEASE.TSUOUT
JES2.REPEAT.DEV
JES2.RESTART.DEV
JES2.ROUTE.JOBOUT
JES2.START.DEV
JES2.STOP.BAT
JES2.STOP.DEV
JES2.STOP.JOB
JES2.STOP.JST
JES2.STOP.STC
JES2.STOP.TSU

Permit *update* access to these resources for those operators responsible for managing the corresponding type of work. However, if an operator is required to change or cancel automatic commands issued by a different operator, grant the operator *control* access to *JES2*.CANCEL.AUTOCMD and *JES2*.MODIFY.AUTOCMD.

(11) Define resources in the OPERCMDS class for each of the following with a default access of *none*:

JES2.HALT.AUTOCMD
JES2.HALT.DEV
JES2.HALT.INITIATOR
JES2.HALT.SPOOL
JES2.MODIFY.INITIATOR
JES2.MODIFY.LINE
JES2.MODIFY.LOGON
JES2.MODIFY.MEMBER
JES2.MODIFY.NODE
JES2.MODIFY.NUM
JES2.MODIFY.OFF
JES2.MODIFY.OFFLOAD
JES2.MODIFY.RMT
JES2.MODIFY.SSI
JES2.MODIFY.SYS
JES2.MODIFY.*initstmt*
JES2.RESTART.BAT
JES2.RESTART.LOGON

JES2.RESTART.MEMBER
JES2.RESTART.SYS
JES2.STOP.INITIATOR
JES2.STOP.LINE
JES2.STOP.LOGON
JES2.STOP.RMT
JES2.STOP.SPOOL
JES2.STOP.SYS
JES2.STOP.TRACE
JES2.START.AUTOCMD
JES2.START.INITIATOR
JES2.START.LINE
JES2.START.LOGON
JES2.START.NET
JES2.START.RMT
JES2.START.SPOOL
JES2.START.SYS
JES2.START.TRACE
JES2.VS

Permit *control* access to these resources for those operators responsible for managing the corresponding type of work. However, those resources marked above in **BOLD** will be restricted to senior personnel. Policies and procedures will be imposed to ensure that they are only used at the direction of network and systems personnel responsible for JES2.

(12) Define resources in the OPERCMDS class for each of the following with a default access of *none*:

JES2.ADD.APPL
JES2.ADD.CONNECT
JES2.ADD.DESTID
JES2.ADD.FSS
JES2.ADD.RMT
JES2.DEL.CONNECT
JES2.DEL.DESTID

The resource definition should be generic if all of the resources of the same type have identical access controls (e.g., all commands issued by systems). The $ADD and $DEL commands are used to reconfigure JES2, and are normally used by or at the direction of systems programmers. Restrict access to these commands to senior personnel. Policies and procedures should be imposed to ensure that they are only used at the direction of network and systems personnel responsible for JES2. Grant authorized operators *control* access to the appropriate resources above.

(13) The resource definitions used when protecting JES2 commands closely relate to the actual commands they protect. When this is combined with the fact that this section only prescribes minimum security levels, and the IAO has the authority to restrict these resources to any additional level, it becomes evident that the total number of possible command resource names is virtually unlimited. Therefore, examples are only given at the minimum required level. By following the examples shown, the resource names for enforcing more stringent security should be evident.

- *(ZJES0054: CAT II) The IAO will ensure that the installation SOP for JES2 system commands reflects users access, logging requirements, justification for access of JES2 system commands and Policies and procedures for use of sensitive JES2 system commands.*

5.1.6 Security Controls for Job Submission, Naming, and Control

Complete the following steps to define and protect the JOB control resources. These represent the minimal security to be installed. More strict controls may be implemented at the discretion of the IAO.

(1) Define the following with a default access of *none*:

JESJOBS class
CANCEL.**
SUBMIT.**

(2) Permit *alter* access to CANCEL.*localnodeid.userid.jobname* for those users allowed to cancel the job, and *read* access to SUBMIT.*localnodeid.jobname.userid* for those users allowed to submit the job. Use generic profiles (wild cards) as much as possible for this purpose. The permissions granted takes into account installation conventions for job names.

5.1.7 Security Controls for Surrogate Users

Apply the following recommendations when implementing security for surrogate users:

(1) Allowing a user to be a surrogate for another user gives the user indirect access to the resources available to the execution user. For this reason, grant surrogate permission to the minimum number of personnel required for running production jobs.

(2) Define a resource *executionuserid*.SUBMIT in the SURROGAT class for each user *executionuserid* on behalf of which a surrogate submits jobs. The default access will be *none*, and logging will be required.

(3) Grant *read* access to *executionuserid*.SUBMIT for each surrogate user.

NOTE: For more in-depth information about surrogate processing, refer to the IBM manual, RACF: Security Implementation for Your Level of RACF.

- *(ZJES0060: CAT II) The IAO will ensure If executionuserid.SUBMIT resources are defined to the SURROGAT resource class all executionuserid.SUBMIT resources defined to the SURROGAT resource class specify a default of no access, all access is logged and access authorization is restricted to the minimum number of personnel required for running production jobs.*

5.1.8 Security Controls for Remote Processing

NJE profiles in the ACP's FACILITY class are used for command and userid authorization from the network. NJE nodes do not sign on as RJE workstations do, but rather perform the FACILITY/USERID verification as each command is issued.

RJE profiles in the FACILITY class are used to force an RJE workstation to log on using a userid (the RJE workstation name) and password using the ACP to perform the validation.

Profiles in the NODES class control how the ACP validates inbound work on an NJE network.

ACP password protection replaces JES2 password protection for remote workstations (specifying RJE passwords in the JES2 startup parameter file). Similarly, ACP command authorization across the network replaces JES2 NJE command authorization. In each case, the workstation/node ID is used as the userid for the purpose of validation. DOD sites should convert to using these ACP protections.

(1) Define the following with a default access of *none*:

FACILITY class
NJE.*
RJE.*

NODES class
node.**

(2) Enable ACP control of NJE nodes and RJE workstations.

(a) For each remote workstation or NJE node, create a userid/user profile.

(b) For each RJE workstation for which the ACP is to check the logon password, create a profile in the ACP’s FACILITY class:

RJE.workstation

where *workstation* is the RJE workstation ID as defined to JES2

NOTE: The mere existence of a profile in the ACP's FACILITY class for a remote workstation forces the workstation password to be checked by the ACP, rather than by JES2. The specification of access rules has no effect.

(c) For each NJE node for which the ACP is to check the command authorization, create a profile in the ACP's FACILITY class as follows:

NJE.nodename

where *nodename* is the NJE nodename as defined to JES2

NOTE: The mere existence of a profile in the ACP's FACILITY class for an NJE node forces the node's command authorization to be checked by the ACP, rather than by JES2. The specification of access rules has no effect.

(d) Because the remote workstation or node ID is also used as a port of entry, it needs to be defined to the ACP's JESINPUT class (if active). If it is not defined and the class is activated, RJE sign-ons or NJE command authorizations will fail because of an incorrect port of entry. Define the workstation and/or node IDs to the ACP's JESINPUT class.

(3) Define profiles in the ACP’s NODES class in accordance with installation policy. A later revision of this document specifies guidelines for those policies when more data is available. A NODES profile name has the following format:

nodeid.keyword.name

where:

nodeid is the name of the node from which inbound work is expected. For jobs, this is the submitting node. For SYSOUT, this is the execution node.

keyword is the type of work to be controlled by the profile.

name is the actual userid, group ID, or security label to be validated. When using NODES profiles to allow the use of these input values, either define these values to the RACF database, or use the ADDMEM operand to translate them into acceptable values for the system. For jobs, the submitter information is substituted. For SYSOUT, the owner information is used.

NOTE: Access lists do not apply to NODES class profiles. The ADDMEM value is used to translate to locally-defined values.

- *(ZJES0014: CAT II) The IAO will ensure that RJE workstations and NJE nodes are defined to the FACILITY resource class.*

5.2 ACF2

The material in this section is based on ACF2, Version 6.1.

5.2.1 Userids for Remote Processing

The following paragraphs identify the minimum security controls for Remote Processing userids. Stricter controls may be specified by the IAO.

(1) Define user profile RMT*nnnn* for each RJE workstation, where *nnnn* is the number on the RMT statement or $ADD RMT command. Do not define any profile segments or grant any access rights. The password controls in *Section 3.1.3, Password Controls,* will apply in full.

```
SET LID
INSERT RMTnnnn NAME(RJE workstation nnnn) GROUP(rmtgrp)
```

(2) Define user profile *nodename* for each NJE node, where *nodename* is the name on the NODE statement or $ADD APPL command. Do not define any profile segments or grant any access rights. The password controls in *Section 3.1.3, Password Controls,* will apply in full.

```
SET LID
INSERT nodename NAME(NJE node nodename) GROUP(rmtgrp)
```

5.2.2 Security Controls for Input

Job and data set input controls are provided via resources in the JESINPUT resource class. Apply the following recommendations when implementing security to JES2 input resources:

(1) Ensure the JESINPUT resource class maps to the STIG required resource type INP as shown in the following example:

```
SET C(GSO)
INSERT CLASMAP.JESINPUT RESOURCE(JESINPUT) RSRCTYPE(INP)
```

(2) Create and compile rule sets with a default access of *none*. The following are sample rules to define resources controlling input to JES2.

```
$KEY(JES2) TYPE(INP)
INTRDR UID(-) PREVENT
nodename UID(-) PREVENT
OFFn.* UID(-) PREVENT
RDRn UID(-) PREVENT
RMTnnnn UID(-) PREVENT
```

```
STCINRDR UID(-) PREVENT
TSUINRDR UID(-) PREVENT
```

The default access of *none* except for sources that are permitted to submit jobs for all users. Those sources may be defined as either *none* or *read*.

(3) Grant *read* access to authorized users for each of the defined input sources.

The following is an example of granting operators with a group of *jesopr* permission to restore jobs into any SPOOL off-load processor after obtaining permission from the IAO:

```
$KEY(JES2) TYPE(INP)
OFF*.JR UID(jesopr) SERVICE(READ) LOG
```

(4) The resource definition should be generic if all of the resources of the same type have identical access controls (e.g., if all off-load receivers are equivalent). The default access is *none* except for sources that are permitted to submit jobs for all users. Those sources may be defined as either *none* or *read*.

5.2.3 Security Controls for Output

Job and data set output controls are provided via resources in the WRITER resource class. Apply the following recommendations when implementing security to JES2 output resources:

(1) Ensure the WRITER resource class maps to the STIG required resource type WTR as shown in the following example:

```
SET C(GSO)
INSERT CLASMAP.WRITER RESOURCE(WRITER) RSRCTYPE(WTR)
```

(2) Define *JES2.*** with a default access of *none*:

```
$KEY(JES2) TYPE(WTR)
UID(-) PREVENT
```

(3) Create and compile rule sets, with a default access of *none*, for each of the output destinations. The following are sample rule sets to define resources controlling output from JES2:

```
$KEY(JES2) TYPE(WTR)
LOCAL.devicename UID(-) PREVENT
LOCAL.OFF*.JT UID(-) PREVENT
LOCAL.OFF*.ST UID(-) PREVENT
LOCAL.PRT* UID(-) PREVENT
LOCAL.PUN* UID(-) PREVENT
```

```
NJE.nodename UID(-) PREVENT
RJE.devicename UID(-) PREVENT
```

(4) Grant *read* access to authorized users for each of the defined output destinations.

The following is an example of granting operators with a group of *jesopr* permission to off-load SYSOUT data sets into any SPOOL off-load processor, after obtaining permission from the IAO:

```
$KEY(JES2) TYPE(WTR)
LOCAL.OFF*.ST UID(jesopr) SERVICE(READ) LOG
```

(5) The resource definition should be generic if all the resources of the same type have identical access controls (e.g., if all off-load transmitters are equivalent). If all users are permitted to route output to a specific destination, the resource controlling may be defined with a default access of either *none* or *read*. Otherwise it is defined with a default access of *none*.

5.2.4 Security Controls for JES2 SPOOL Data Sets

Use the following controls to protect the JES2 SPOOL data sets. Stricter security may be implemented at the direction of the IAO.

(1) Ensure the JESSPOOL resource class maps to the STIG required resource type SPL as shown in the following example:

```
SET C(GSO)
INSERT CLASMAP.JESSPOOL RESOURCE(JESSPOOL) RSRCTYPE(SPL)
```

(2) Create and compile rule sets. The following are sample rule sets to define resources controlling the JES2 SPOOL data sets:

```
$KEY(localnodeid) TYPE(SPL)
JES2.$TRCLOG.*.*.JESTRACE UID(-) PREVENT
- UID(-) PREVENT
+MASTER+.SYSLOG.*.*.SYSLOG UID(-) PREVENT
jesid.$JESNEWS.*.*.JESNEWS UID(-) PREVENT

$KEY(JES2) TYPE(OPR)
UPDATE.JESNEWS UID(-) PREVENT
```

The following is a sample rule to allow production control personnel with a userid matching *prod* to update the JES News data set:

```
$KEY(JES2) TYPE(OPR)
UPDATE.JESNEWS UID(prod) SERVICE(DELETE) LOG
```

The following are sample rules to allow users matching *usr* to access jobs with a prefix of *pfx* belonging to user *own*:

```
$KEY(localnodeid) TYPE(SPL)
own.pfx*- UID(-) PREVENT
own.pfx*.- UID(usr) SERVICE(READ) LOG
```

The following is a sample rule set to allow external writers with a userid matching ***log*** to archive SYSLOG data sets. Access should be strictly limited to external writers used only to archive SYSLOG.

```
$KEY(localnodeid) TYPE(SPL)
+MASTER+.SYSLOG.*.*.SYSLOG UID(log) SERVICE(READ) LOG
```

The following are sample rules to allow JES2 systems programmers with a userid matching *jesa* and OS/390 systems programmers with a userid matching *mvs* to access JES2 SYSLOG and trace data sets. This access should be strictly limited to the minimum number of necessary personnel.

```
$KEY(localnodeid) TYPE(SPL)
+MASTER+.SYSLOG.*.*.SYSLOG UID(jes) SERVICE(READ) LOG
+MASTER+.SYSLOG.*.*.SYSLOG UID(mvs) SERVICE(READ) LOG
JES2.$TRCLOG.*.*.JESTRACE UID(jes) SERVICE(READ) LOG
```

5.2.5 Security Controls for JES2 Commands

Extended MCS support allows the installation to control the use of JES2 system commands through the ACP. These commands are subject to various types of potential abuse. For this reason, it is necessary to place restrictions on the JES2 system commands that can be entered by particular operators. To control access to JES2 system commands, apply the following recommendations when implementing security:

(1) Define the default access to JES2 system commands as *none*:

```
$KEY(JES2) TYPE(OPR)
UID(-) PREVENT
```

(2) The following are sample rules to define the resources controlling JES2 system commands that display non-sensitive information:

```
$KEY(JES2) TYPE(OPR)
DISPLAY.ACTRMT UID(-) PREVENT
DISPLAY.BAT UID(-) PREVENT
DISPLAY.BATOUT UID(-) PREVENT
DISPLAY.DEV UID(-) PREVENT
DISPLAY.INITIATOR UID(-) PREVENT
DISPLAY.JOBUID(-) PREVENT
DISPLAY.JOBOUT UID(-) PREVENT
DISPLAY.JST UID(-) PREVENT
DISPLAY.JSTOUT UID(-) PREVENT
DISPLAY.PCE UID(-) PREVENT
DISPLAY.QUE UID(-) PREVENT
DISPLAY.REBLD UID(-) PREVENT
DISPLAY.SPOOL UID(-) PREVENT
DISPLAY.STC UID(-) PREVENT
DISPLAY.STCOUT UID(-) PREVENT
DISPLAY.TRACE UID(-) PREVENT
DISPLAY.TSU UID(-) PREVENT
DISPLAY.TSUOUT UID(-) PREVENT
DISPLAY.initstmt UID(-) PREVENT
GDISPLAY.JOB UID(-) PREVENT
```

(3) Grant *read* access to these resources to any operator who requires it. Logging is not required. The following is a sample rule to allow operators matching *opr* to use the $DS command:

```
$KEY(JES2) TYPE(OPR)
DISPLAY.STC UID(opr) SERVICE(READ) ALLOW
```

(4) The following are sample rules to define the resources controlling JES2 system commands that allow the operator to send messages to running jobs and to other members of a multi-access spool (MAS) configuration, and/or to send commands to remote sites:

```
$KEY(JES2) TYPE(OPR)
MSEND.MESSAGE UID(-) PREVENT
NSEND.MESSAGE UID(-) PREVENT
SEND.MESSAGE UID(-) PREVENT
```

(5) Grant *read* to these resources for any operator who is authorized to send such commands and messages. The following is a sample rule to allow operators matching *opr* to use the $DM command:

```
$KEY(JES2) TYPE(OPR)
SEND.MESSAGE UID(opr) SERVICE(READ) LOG
```

(6) The following are sample commands to define the resources controlling routine JES2 system commands that allow the operator to manage the JES2 workload:

```
$KEY(JES2) TYPE(OPR)
BACKSP.DEV UID(-) PREVENT
CANCEL.AUTOCMD UID(-) PREVENT
CANCEL.BAT UID(-) PREVENT
CANCEL.DEV UID(-) PREVENT
CANCEL.JOB UID(-) PREVENT
CANCEL.JST UID(-) PREVENT
CANCEL.STC UID(-) PREVENT
CANCEL.TSU UID(-) PREVENT
FORWARD.DEV UID(-) PREVENT
GCANCEL.JOB UID(-) PREVENT
GMODIFYHOLD.JOB UID(-) PREVENT
GROUTE.JOBOUT UID(-) PREVENT
HALT.DEV UID(-) PREVENT
INTERRUPT.DEV UID(-) PREVENT
MODIFY.AUTOCMD UID(-) PREVENT
MODIFY.BAT UID(-) PREVENT
MODIFY.BATOUT UID(-) PREVENT
MODIFY.JOB UID(-) PREVENT
MODIFY.JOBOUT UID(-) PREVENT
MODIFY.JST UID(-) PREVENT
MODIFY.JSTOUT UID(-) PREVENT
MODIFY.STC UID(-) PREVENT
MODIFY.STCOUT UID(-) PREVENT
MODIFY.TSU UID(-) PREVENT
MODIFY.TSUOUT UID(-) PREVENT
MODIFYHOLD.BAT UID(-) PREVENT
MODIFYHOLD.JOB UID(-) PREVENT
MODIFYHOLD.JST UID(-) PREVENT
MODIFYHOLD.STC UID(-) PREVENT
MODIFYHOLD.TSU UID(-) PREVENT
MODIFYRELEASE.BAT UID(-) PREVENT
MODIFYRELEASE.JOB UID(-) PREVENT
MODIFYRELEASE.JST UID(-) PREVENT
MODIFYRELEASE.STC UID(-) PREVENT
MODIFYRELEASE.TSU UID(-) PREVENT
RELEASE.BATOUT UID(-) PREVENT
RELEASE.JOBOUT UID(-) PREVENT
RELEASE.JSTOUT UID(-) PREVENT
RELEASE.STCOUT UID(-) PREVENT
RELEASE.TSUOUT UID(-) PREVENT
REPEAT.DEV UID(-) PREVENT
RESTART.DEV UID(-) PREVENT
ROUTE.JOBOUT UID(-) PREVENT
START.DEV UID(-) PREVENT
STOP.BAT UID(-) PREVENT
STOP.DEV UID(-) PREVENT
STOP.JOB UID(-) PREVENT
STOP.JST UID(-) PREVENT
STOP.STC UID(-) PREVENT
STOP.TSU UID(-) PREVENT
```

Permit *update* access to these resources for those operators responsible for managing the corresponding type of work. However, if an operator is required to change or cancel automatic commands issued by a different operator, grant the operator *control* access to *JES2*.CANCEL.AUTOCMD and *JES2*.MODIFY.AUTOCMD. The following is a sample rule to allow operators matching *opr* to use the $B command:

```
$KEY(JES2) TYPE(OPR)
BACKSP.DEV UID(opr) SERVICE(UPDATE) LOG
```

(7) The following are sample rules to define the resources controlling JES2 system commands that control JES2 facilities:

```
$KEY(JES2) TYPE(OPR)
HALT.AUTOCMD UID(-) PREVENT
HALT.DEV UID(-) PREVENT
HALT.INITIATOR UID(-) PREVENT
MODIFY.INITIATOR UID(-) PREVENT
MODIFY.LINE UID(-) PREVENT
MODIFY.LOGON UID(-) PREVENT
MODIFY.MEMBER UID(-) PREVENT
MODIFY.NUM UID(-) PREVENT
MODIFY.OFF UID(-) PREVENT
MODIFY.OFFLOAD UID(-) PREVENT
MODIFY.RMT UID(-) PREVENT
MODIFY.SYS UID(-) PREVENT
RESTART.LOGON UID(-) PREVENT
STOP.INITIATOR UID(-) PREVENT
STOP.LINE UID(-) PREVENT
STOP.LOGON UID(-) PREVENT
STOP.RMT UID(-) PREVENT
STOP.SPOOL UID(-) PREVENT
STOP.SYS UID(-) PREVENT
STOP.TRACE UID(-) PREVENT
START.AUTOCMD UID(-) PREVENT
START.INITIATOR UID(-) PREVENT
START.LINE UID(-) PREVENT
START.LOGON UID(-) PREVENT
START.NET UID(-) PREVENT
START.RMT UID(-) PREVENT
START.SPOOL UID(-) PREVENT
START.SYS UID(-) PREVENT
START.TRACE UID(-) PREVENT
VS UID(-) PREVENT
```

Permit *control* access to these resources for those operators responsible for managing the corresponding type of work. The ACF2 equivalent is SERVICE(DELETE). The following is a sample rule to allow operators matching *opr* to use the $ZI command:

```
$KEY(JES2) TYPE(OPR)
HALT.INITIATOR UID(opr) SERVICE(DELETE) LOG
```

(8) The following are sample rules to define resources controlling sensitive JES2 system commands that control JES2 facilities:

```
$KEY(JES2) TYPE(OPR)
ADD.APPL UID(-) PREVENT
ADD.CONNECT UID(-) PREVENT
ADD.DESTID UID(-) PREVENT
ADD.FSS UID(-) PREVENT
ADD.RMT UID(-) PREVENT
HALT.SPOOL UID(-) PREVENT
MODIFY.NODE UID(-) PREVENT
MODIFY.SSI UID(-) PREVENT
MODIFY.initstmt UID(-) PREVENT
RESTART.BAT UID(-) PREVENT
RESTART.MEMBER UID(-) PREVENT
RESTART.SYS UID(-) PREVENT
```

Permit *control* access to these resources for those operators responsible for managing the corresponding type of work. However, access to these commands should be restricted to senior personnel. Policies and procedures should be imposed to ensure that they are only used at the direction of network and systems personnel responsible for JES2. The following is a sample rule to allow operators matching *opr* to use the $ZSPOOL command:

```
$KEY(JES2) TYPE(OPR)
HALT.SPOOL UID(opr) SERVICE(DELETE) LOG
```

5.2.6 Security Controls for Job Submission, Naming, and Control

The TSO SUBMIT and CANCEL commands are controlled via resources in the JESJOBS class.

(1) Ensure the JESJOBS resource class maps to the STIG required resource type JOB as shown in the following example:

```
SET C(GSO)
INSERT CLASMAP.JESJOBS RESOURCE(JESJOBS) RSRCTYPE(JOB)
```

(2) Define the following with a default access of *none*:

```
$KEY(CANCEL) TYPE(JOB)
UID(-) PREVENT

$KEY(SUBMIT) TYPE(JOB)
UID(-) PREVENT
```

(3) Permit *alter* access to CANCEL.*localnodeid.userid.jobname* for those users allowed to cancel the job. The ACF2 equivalent is SERVICE(ADD). Use generic profiles (wild cards) as much as possible for this purpose. The permissions granted should take into account installation conventions for job names.

```
$KEY(CANCEL) TYPE(JOB)
localnodeid.userid.jobmask UID(-) PREVENT
localnodeid.userid.jobmask UID(opr) SERVICE(ADD) LOG
```

(4) Permit *read* access to SUBMIT.*localnodeid.jobname.userid* for those users allowed to submit the job. Use generic profiles (wild cards) as much as possible for this purpose. The permissions granted should take into account installation conventions for job names.

```
$KEY(SUBMIT) TYPE(JOB)
localnodeid.jobmask.userid UID(-) PREVENT
localnodeid.jobmask.userid UID(opr) SERVICE(READ) LOG
```

5.2.7 Security Controls for Surrogate Users

Apply the following recommendations when implementing security for surrogate users:

(1) Allowing a user to be a surrogate for another user gives that user indirect access to the resources available to the execution user. For this reason, grant surrogate permission to the minimum number of personnel required for running production jobs.

(2) Ensure the SURROGAT resource class maps to the STIG required resource type SUR as shown in the following example:

```
SET C(GSO)
INSERT CLASMAP.SURROGAT RESOURCE(SURROGAT) RSRCTYPE(SUR)
```

(3) Define a resource rule set for each user *executionuserid* on behalf of which a surrogate submits jobs. The default access is *none*, and logging is be required. Grant *read* access to *executionuserid*.SUBMIT for each authorized surrogate user.

```
$KEY(executionuserid) TYPE(SUR)
UID(-) PREVENT
SUBMIT UID(profile) SERVICE(READ) LOG
```

5.2.8 Security Controls for Remote Processing

Implement the following controls for JES2 Remote Processing:

(1) Enable ACP control of NJE nodes and RJE workstations by defining resources in the FACILITY class:

```
$KEY(NJE) TYPE(FAC)
nodename UID(-) PREVENT

$KEY(RJE) TYPE(FAC)
workstation-id UID(-) PREVENT
```

Access rights do not need to be granted to these resources. JES2 only tests to ensure they are active.

(2) Ensure the NODES resource class maps to the STIG required resource type NOD as shown in the following example:

```
SET C(GSO)
INSERT CLASMAP.NODES RESOURCE(NODES) RSRCTYPE(NOD)
```

(3) Define profiles in the NODES class in accordance with installation policy. A later revision of this *STIG* specifies guidelines for those policies when more data is available. The following is a sample command to allow entry of jobs with a userid of *userid* from NJE node *nodename*. The jobs have explicit userids and passwords.

```
$KEY(nodename) TYPE(NOD)
UID(-) PREVENT
USERJ.userid UID(-) SERVICE(READ) LOG
```

5.3 RACF

5.3.1 Userids for Remote Processing

The following paragraphs identify the minimum security controls for Remote Processing userids. Stricter controls may be specified by the IAO.

(1) Define userid RMT*nnnn* for each RJE workstation, where *nnnn* is the number on the RMT statement or $ADD RMT command. Do not define any profile segments or grant any access rights. The password controls in *Section 3.1.3, Password Controls,* will apply in full.

ADDUSER RMT*nnnn* DATA('RJE workstation *nnnn*') -
NAME('RJE workstation *nnnn*') -
PASSWORD(*password*) -
DFLTGRP(*rmtgrp*)

(2) Define userid *nodename* for each NJE node, where *nodename* is the name on the NODE statement or $ADD APPL command. Do not define any profile segments or grant any access rights. The password controls in *Section 3.1.3, Password Controls,* will apply in full.

ADDUSER *nodename* DATA('NJE node *nodename*') -
NAME(‘NJE node *nodename’*) -
PASSWORD(*password*) -
DFLTGRP(rmtgrpacid)

5.3.2 Security Controls for Input

Job and data set input controls are provided via resources in the JESINPUT resource class. This class should already be active and use generic masking, but the sample commands shown below include the relevant SETROPTS commands[21] for the sake of completeness. When protecting the facilities via the JESINPUT class, use the following controls:

(1) Create generic and specific profiles as follows:

SETROPTS GENERIC(JESINPUT)
RDEFINE JESINPUT INTRDR AUDIT(ALL) UACC(NONE)
RDEFINE JESINPUT *nodename* AUDIT(ALL) UACC(NONE)
RDEFINE JESINPUT OFF*n*. AUDIT(ALL) UACC(NONE)
RDEFINE JESINPUT RDR*n* AUDIT(ALL) UACC(NONE)
RDEFINE JESINPUT STCINRDR AUDIT(ALL) UACC(NONE)
RDEFINE JESINPUT TSUINRDR AUDIT(ALL) UACC(NONE)

[21] The SETROPTS REFRESH is only shown once; it must be repeated as necessary.

```
SETROPTS CLASSACT(JESINPUT)
SETROPTS RACLIST(JESINPUT) REFRESH
```

The default access should be *none* except for sources that are permitted to submit jobs for all users. Those sources may be defined as either *none* or *read*.

(2) Grant *read* access to authorized users for each of the defined input sources.

The following is an example of granting operators with a GROUP of *jesopr* permission to restore jobs into any SPOOL off-load processor after obtaining permission from the IAO:

```
PERMIT OFF*.JR CLASS(JESINPUT) ID(jesopr) ACCESS(READ)
```

The resource definition should be generic if all of the resources of the same type have identical access controls (e.g., if all off-load receivers are equivalent). The default access is *none* except for sources that are permitted to submit jobs for all users. Those sources may be defined as either *none* or *read*.

5.3.3 Security Controls for Output

Job and data set output controls are provided via resources in the WRITER resource class. This class should already be active and use generic masking, but the sample commands shown below include the relevant SETROPTS commands[22] for the sake of completeness. When protecting the facilities via the WRITER class, use the following controls:

(1) Define *JES2*.** with a default access of *none*:

```
SETROPTS GENERIC(WRITER)
RDEFINE WRITER JES2.** AUDIT(ALL) UACC(NONE)
SETROPTS CLASSACT(WRITER)
SETROPTS RACLIST(WRITER) REFRESH
```

(2) Define resources for each of the output destinations:

```
RDEFINE WRITER JES2.LOCAL.devicename AUDIT(ALL) UACC(NONE)
RDEFINE WRITER JES2.LOCAL.OFF*.JT AUDIT(ALL) UACC(NONE)
RDEFINE WRITER JES2.LOCAL.OFF*.ST AUDIT(ALL) UACC(NONE)
RDEFINE WRITER JES2.LOCAL.PRT* AUDIT(ALL) UACC(NONE)
RDEFINE WRITER JES2.LOCAL.PUN* AUDIT(ALL) UACC(NONE)
RDEFINE WRITER JES2.NJE.nodename AUDIT(ALL) UACC(NONE)
RDEFINE WRITER JES2.RJE.devicename AUDIT(ALL) UACC(NONE)
```

[22] The SETROPTS REFRESH is only shown once; it must be repeated as necessary.

(3) Grant *read* access to authorized users for each of the defined output destinations.

The following is an example of granting operators with a group of *jesopr* permission to off-load SYSOUT data sets into any SPOOL off-load processor after obtaining permission from the IAO:

PERMIT *JES2*.LOCAL.OFF*.ST CLASS(WRITER) ID(*jesopr*) ACCESS(READ)

The resource definition should be generic if all of the resources of the same type have identical access controls (e.g., if all off-load transmitters are equivalent). If all users are permitted to route output to a specific destination, the resource controlling it may be defined with a default access of either *none* or *read*. Otherwise it is defined with a default access of *none*.

5.3.4 Security Controls for JES2 SPOOL Data Sets

The following are sample commands to define resources controlling the JES2 SPOOL data sets:

RDEFINE JESSPOOL *localnodeid*.** AUDIT(ALL) UACC(NONE)
RDEFINE JESSPOOL *localnodeid*.*JES2*.$TRCLOG.*.*.JESTRACE AUDIT(ALL) UACC(NONE)
RDEFINE JESSPOOL *localnodeid*.+MASTER+.SYSLOG.*.*.SYSLOG AUDIT(ALL) UACC(NONE)
RDEFINE JESSPOOL *localnodeid*.*jesid*.$JESNEWS.*.*.JESNEWS UACC(READ)
RDEFINE OPERCMDS *JES2*.UPDATE.JESNEWS AUDIT(ALL) UACC(NONE)

The following is a sample command to allow production control personnel with a group of *prod* to update the JES News data set:

PERMIT *JES2*.UPDATE.JESNEWS CLASS(OPERCMDS) ID(*prod*) ACCESS(CONTROL)

The following are sample commands to allow users with a group of *usr* to access jobs with a prefix of *pfx* belonging to user *own*:

RDEFINE JESSPOOL *localnodeid*.*own*.*pfx**. AUDIT(ALL) UACC(NONE)
PERMIT *localnodeid*.*own*.*pfx**.** CLASS(JESSPOOL) ID(*usr*) ACCESS(READ)

The following is a sample command to allow external writers with a group of *log* to archive SYSLOG data sets. Access should be strictly limited to external writers used only to archive SYSLOG.

PERMIT *localnodeid*.+MASTER+.SYSLOG.*.*.SYSLOG CLASS(JESSPOOL) ID(*log*) ACCESS(READ)

The following are sample commands to allow JES2 systems programmers with a group of *jes* and OS/390 systems programmers with a group of *mvs* to access JES2 SYSLOG and trace data sets. This access should be strictly limited to the minimum number of necessary personnel.

PERMIT *localnodeid*.+MASTER+.SYSLOG.*.*.SYSLOG CLASS(JESSPOOL) ID(*jes*) ACCESS(READ)

PERMIT *localnodeid*.*JES2*.$TRCLOG.*.*.JESTRACE CLASS(JESSPOOL) ID(*jes*) ACCESS(READ)

PERMIT *localnodeid*.+MASTER+.SYSLOG.*.*.SYSLOG CLASS(JESSPOOL) ID(*mvs*) ACCESS(READ)

5.3.5 Security Controls for JES2 Commands

Extended MCS support allows the installation to control the use of JES2 system commands through the ACP. These commands are subject to various types of potential abuse. For this reason, it is necessary to place restrictions on the JES2 system commands that can be entered by particular operators. To control access to JES2 system commands, apply the following recommendations when implementing security:

(1) Define the default access to JES2 system commands as *none*:

RDEFINE OPERCMDS *JES2*.** AUDIT(ALL) UACC(NONE)

(2) The following are sample commands to define the resources controlling JES2 system commands that display non-sensitive information:

RDEFINE OPERCMDS *JES2*.DISPLAY.ACTRMT UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.BAT UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.BATOUT UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.DEV UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.INITIATOR UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.JOB UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.JOBOUT UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.JST UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.JSTOUT UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.PCE UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.QUE UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.REBLD UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.SPOOL UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.STC UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.STCOUT UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.TRACE UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.TSU UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.TSUOUT UACC(NONE)
RDEFINE OPERCMDS *JES2*.DISPLAY.*initstmt* UACC(NONE)
RDEFINE OPERCMDS *JES2*.GDISPLAY.JOB UACC(NONE)

Grant *read* access to these resources to any operator who requires it. Logging is not required. The following is a sample command to allow operators with a group of *opr* to use the $DS command:

```
PERMIT JES2.DISPLAY.STC CLASS(OPRCMDS) ID(opr) ACCESS(READ)
```

(3) The following are sample commands to define the resources controlling JES2 system commands that allow the operator to send messages to running jobs and to other members of a multi-access spool (MAS) configuration, or to send commands to remote sites:

```
RDEFINE OPERCMDS JES2.MSEND.MESSAGE AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.NSEND.MESSAGE AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.SEND.MESSAGE AUDIT(ALL) UACC(NONE)
```

Grant *read* access to these resources for any operator who is authorized to send such commands and messages. The following is a sample command to allow operators with a group of *opr* to use the $DM command:

```
PERMIT JES2.SEND.MESSAGE CLASS(OPRCMDS) ID(opr) ACCESS(READ)
```

(4) The following are sample commands to define the resources controlling routine JES2 system commands that allow the operator to manage the JES2 workload:

```
RDEFINE OPERCMDS JES2.BACKSP.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.CANCEL.AUTOCMD AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.CANCEL.BAT  AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.CANCEL.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.CANCEL.JOB  AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.CANCEL.JST   AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.CANCEL.STC AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.CANCEL.TSU AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.FORWARD.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.GCANCEL.JOB  AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.GMODIFYHOLD.JOB AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.GROUTE.JOBOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.HALT.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.INTERRUPT.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.AUTOCMD AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.BAT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.BATOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.JOB AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.JOBOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.JST AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.JSTOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.STC AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.STCOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.TSU AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.TSUOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFYHOLD.BAT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFYHOLD.JOB  AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFYHOLD.JST   AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFYHOLD.STC AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFYHOLD.TSU AUDIT(ALL)UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFYRELEASE.BAT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFYRELEASE.JOB AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFYRELEASE.JST AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFYRELEASE.STC AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS ES2.MODIFYRELEASE.TSU AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JE2.RELEASE.BATOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.RELEASE.JOBOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.RELEASE.JSTOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.RELEASE.STCOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.RELEASE.TSUOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.REPEAT.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.RESTART.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.ROUTE.JOBOUT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.BAT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.JOB   AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.JST   AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.STC AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.TSU AUDIT(ALL) UACC(NONE)
```

Permit *update* access to these resources for those operators responsible for managing the corresponding type of work. However, if an operator is required to change or cancel automatic commands issued by a different operator, grant that operator *control* access to *JES2*.CANCEL.AUTOCMD and *JES2*.MODIFY.AUTOCMD. The following is a sample command to allow operators with a group of *opr* to use the $B command:

```
PERMIT JES2.BACKSP.DEV CLASS(OPERCMDS) ID(opr) ACCESS(UPDATE)
```

(5) The following are sample commands to define the resources controlling JES2 system commands that control JES2 facilities:

```
RDEFINE OPERCMDS JES2.HALT.AUTOCMD AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.HALT.DEV AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.HALT.INITIATOR AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.INITIATOR AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.LINE AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.LOGON AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.MEMBER AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.NUM AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.OFF AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.OFFLOAD AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.RMT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.SYS AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.RESTART.LOGON AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.INITIATOR AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.LINE AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.LOGON AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.RMT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.SPOOL AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.SYS AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.STOP.TRACE AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.AUTOCMD AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.INITIATOR AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.LINE AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.LOGON AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.NET AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.RMT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.SPOOL AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.SYS AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.START.TRACE AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.VS AUDIT(ALL) UACC(NONE)
```

Permit *control* access to these resources for those operators responsible for managing the corresponding type of work. The following is a sample command to allow operators in group *opr* to use the $ZI command:

```
PERMIT JES2.HALT.INITIATOR CLASS(OPERCMDS) ID(opr) ACCESS(CONTROL)
```

(6) The following are sample commands to define resources controlling sensitive JES2 system commands that control JES2 facilities:

```
RDEFINE OPERCMDS JES2.ADD.APPL AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.ADD.CONNECT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.ADD.DESTID AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.ADD.FSS AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.ADD.RMT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.HALT.SPOOL AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.NODE AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.SSI AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.MODIFY.initstmt AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.RESTART.BAT AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.RESTART.MEMBER AUDIT(ALL) UACC(NONE)
RDEFINE OPERCMDS JES2.RESTART.SYS AUDIT(ALL) UACC(NONE)
```

Permit *control* access to these resources for those operators responsible for managing the corresponding type of work. However, access to these commands should be restricted to senior personnel. Policies and procedures should be imposed to ensure that they are only used at the direction of network and systems personnel responsible for JES2. The following is a sample command to allow operators in group *opr* to use the $ZSPOOL command:

PERMIT*JES2*.HALT.SPOOL CLASS(OPERCMDS) ID(*opr*) ACCESS(CONTROL)

5.3.6 Security Controls for Job Submission, Naming, and Control

The TSO SUBMIT and CANCEL commands are controlled via resources in the JESJOBS class. This class should already be active and use generic masking, but the sample commands shown below include the relevant SETROPTS commands[23] for the sake of completeness. When protecting the facilities for these commands via the JESJOBS class, use the following controls:

(1) Define the following with a default access of *none*:

RDEFINE JESJOBS CANCEL.** AUDIT(ALL) UACC(NONE)
RDEFINE JESJOBS SUBMIT.** AUDIT(ALL) UACC(NONE)

(2) Permit *alter* access to CANCEL.*localnodeid.userid.jobname* for those users allowed to cancel the job. Use generic profiles (wild cards) as much as possible for this purpose. The permissions granted takes into account installation conventions for job names.

RDEFINE JESJOBS CANCEL.localnodeid.userid.jobmask AUDIT(ALL) UACC(NONE)

PERMIT CANCEL.*localnodeid.userid.jobmask* CLASS(JESJOBS) ID(*opr*)
ACCESS(ALTER)

(3) Permit *read* access to SUBMIT.*localnodeid.jobname.userid* for those users allowed to submit the job. Use generic profiles (wild cards) as much as possible for this purpose. The permissions granted takes into account installation conventions for job names.

RDEFINE JESJOBS SUBMIT.*localnodeid.jobmask.userid* AUDIT(ALL) UACC(NONE)

PERMIT SUBMIT.*localnodeid.jobmask.userid* CLASS(JESJOBS) ID(*opr*)
ACCESS(READ)

5.3.7 Security Controls for Surrogate Users

Apply the following recommendations when implementing security for surrogate users:

[23] The SETROPTS REFRESH is only shown once; it must be repeated as necessary.

(1) Allowing a user to be a surrogate for another user gives that user indirect access to the resources available to the execution user. For this reason, grant surrogate permission to the minimum number of personnel required for running production jobs.

(2) Define a resource profile for each user *executionuserid* on behalf of which a surrogate submits jobs. The default access will be *none* and logging will be required.

RDEFINE SURROGAT *executionuserid*.SUBMIT AUDIT(ALL) UACC(NONE)

(3) Grant *read* access to *executionuserid*.SUBMIT for the group or user profile of each authorized surrogate user:

PERMIT *executionuserid*.SUBMIT CLASS(SURROGAT) ID(*profile*) ACCESS(READ)

5.3.8 Security Controls for Remote Processing

Implement the following controls for JES2 Remote Processing:

(1) Enable ACP control of NJE nodes and RJE workstations by defining resources in the FACILITY class:

RDEFINE FACILITY NJE.*nodename* AUDIT(ALL) UACC(NONE)
RDEFINE FACILITY RJE.*workstation-id* AUDIT(ALL) UACC(NONE)

Access rights do not need to be granted to these resources. JES2 only tests to ensure they are active.

(2) Define profiles in the NODES class in accordance with installation policy. A later revision of this *STIG* specifies guidelines for those policies when more data is available. The following is a sample command to allow entry of jobs with a userid of *userid* from NJE node *nodename*. The jobs have explicit userids and passwords.

RDEFINE NODES *nodename*.USERJ.*userid* AUDIT(ALL) UACC(READ)

5.4 TOP SECRET

5.4.1 Userids for Remote Processing

The following paragraphs identify the minimum security controls for Remote Processing userids. Stricter controls may be specified by the IAO.

(1) Define user ACID RMT*nnnn* for each RJE workstation, where *nnnn* is the number on the RMT statement or $ADD RMT command. Do not define any profile segments or grant any access rights other than the CONSOLE facility. The password controls in *Section 3.1.3, Password Controls*, will apply in full.

TSS CREATE(RMT*nnnn*) NAME('RJE workstation *nnnn*') -
PASSWORD(*password*,0) FACILITY(CONSOLE) -
DFLTGRP(rmtgrpacid) GROUP(rmtgrpacid)

(2) Define user ACID *nodename* for each NJE node, where *nodename* is the name on the NODE statement or $ADD APPL command. Do not define any profile segments or grant any access rights. The password controls in *Section 3.1.3, Password Controls*, will apply in full.

TSS CREATE(*nodename*) NAME('NJE node *nodename*') -
PASSWORD(*password*,0) -
DFLTGRP(rmtgrpacid) GROUP(rmtgrpacid)

5.4.2 Security Controls for Input

Job and data set input controls are provided via resources in the JESINPUT resource class. The actual owning ACID specified for *deptacid* should be named in accordance with installation recommendations.

(1) The following commands may be used to establish default protection for resources defined to the **JESINPUT** resource class:

TSS ADDTO(*deptacid*) JESINPUT(INTRDR)
TSS ADDTO(*deptacid*) JESINPUT(*nodename*)
TSS ADDTO(*deptacid*) JESINPUT(OFF*n*.)
TSS ADDTO(*deptacid*) JESINPUT(RDR*n*)

(2) Grant *read* access to authorized users for each of the resources defined to the JESINPUT resource class.

The following is an example of granting operators with a profile ACID of *jesopracid* permission to restore jobs into any SPOOL off-load processor after obtaining permission from the IAO:

TSS PERMIT(*jesopracid*) JESINPUT(OFF*.JR) ACCESS(READ) ACTION(AUDIT)

The resource definition should be generic if all of the resources of the same type have identical access controls (e.g., if all off-load receivers are equivalent).

5.4.3 Security Controls for Output

Job and data set output controls are provided via resources in the WRITER resource class. The actual owning ACID specified for *deptacid* should be named in accordance with installation recommendations.

(1) The following command may be used to establish default protection for resources defined to the WRITER resource class:

TSS ADDTO(*deptacid*) WRITER(*JES2*.)

(2) Grant *read* access to authorized users for each of the following WRITER resource class output destinations:

JES2.LOCAL.devicename
JES2.LOCAL.OFF*.JT
JES2.LOCAL.OFF*.ST
JES2.LOCAL.PRT*
JES2.LOCAL.PUN*
JES2.NJE.*nodename*
JES2.RJE.*devicename*

The following is an example of granting operators with a profile ACID of *jesopracid* permission to off-load SYSOUT data sets into any SPOOL off-load processor after obtaining permission from the IAO:

TSS PERMIT(*jesopracid*) WRITER(*JES2*.LOCAL.OFF*.ST) -
ACCESS(READ) ACTION(AUDIT)

The resource definition should be generic if all of the resources of the same type have identical access controls (e.g., if all off-load transmitters are equivalent).

5.4.4 Security Controls for JES2 SPOOL Data Sets

(1) The following command may be used to establish default protection for resources defined to the JESSPOOL resource class:

TSS ADDTO(*deptacid*) JESSPOOL(*localnodeid*.)

Due to the protection established with the previous command, the following command should be issued to ensure users are able to access their own spool data:

TSS PERMIT(ALL) JESSPOOL(*localnodeid*.%) ACCESS(ALL)

(2) Grant *read* access to authorized users for each of the following JESSPOOL resources:

```
localnodeid.JES2.$TRCLOG.*.*.JESTRACE
localnodeid.+MASTER+.SYSLOG.*.*.SYSLOG
localnodeid.jesid.$JESNEWS.*.*.JESNEWS
```

The following command example may be used to allow all valid TOP SECRET users *read* access the JES News data set:

```
TSS PERMIT(ALL) JESSPOOL(localnodeid.jesid.$JESNEWS.*.*.JESNEWS) –
ACCESS(READ)
```

The following is a sample command to allow production control personnel with a profile ACID of *prodacid* to update the JES News data set:

```
TSS PERMIT(prodacid) OPERCMDS(JES2.UPDATE.JESNEWS) -
ACCESS(CONTROL) ACTION(AUDIT)
```

NOTE: Refer to *Section 5.4.5, Security Controls for JES2 Commands*, for information on JES2 command control requirements.

The following is a sample command to allow users with a profile ACID of *usracid* to access jobs with a prefix of *pfx* belonging to user *ownacid*:

```
TSS PERMIT(usracid) JESSPOOL(localnodeid.ownacid.pfx*.) -
ACCESS(READ) ACTION(AUDIT)
```

The following is a sample command to allow external writers with a profile ACID of *logacid* to archive SYSLOG data sets. Access is strictly limited to external writers used only to archive SYSLOG.

```
TSS PERMIT(logacid) JESSPOOL(localnodeid.+MASTER+.SYSLOG.*.*.SYSLOG) -
      ACCESS(READ) ACTION(AUDIT)
```

The following are sample commands to allow JES2 systems programmers with a profile ACID of *jesacid* and OS/390 systems programmers with a profile ACID of *mvsacid* to access JES2 SYSLOG and trace data sets. Strictly limit this access to the minimum number of necessary personnel.

```
TSS PERMIT(jesacid) JESSPOOL(localnodeid.+MASTER+.SYSLOG.*.*.SYSLOG) -
ACCESS(READ) ACTION(AUDIT)

TSS PERMIT(jesacid) JESSPOOL(localnodeid.JES2.$TRCLOG.*.*.JESTRACE) -
ACCESS(READ) ACTION(AUDIT)

TSS PERMIT(mvsacid) JESSPOOL(localnodeid.+MASTER+.SYSLOG.*.*.SYSLOG) -
```

ACCESS(READ) ACTION(AUDIT)

5.4.5 Security Controls for JES2 Commands

Extended MCS support allows the installation to control the use of JES2 system commands through the ACP. These commands are subject to various types of potential abuse. For this reason, it is necessary to place restrictions on the JES2 system commands that can be entered by particular operators. To control access to JES2 system commands, the following recommendations will be applied when implementing security:

(1) The following command may be used to establish default protection for JES2 system commands defined to the OPERCMDS resource class:

TSS ADDTO(*deptacid*) OPERCMDS(*JES2*.)

(2) The following OPERCMDS resources can be used to control access to JES2 system commands that display non-sensitive information:

JES2.DISPLAY.ACTRMT
JES2.DISPLAY.BAT
JES2.DISPLAY.BATOUT
JES2.DISPLAY.DEV
JES2.DISPLAY.INITIATOR
JES2.DISPLAY.JOB
JES2.DISPLAY.JOBOUT
JES2.DISPLAY.JST
JES2.DISPLAY.JSTOUT
JES2.DISPLAY.PCE
JES2.DISPLAY.QUE
JES2.DISPLAY.REBLD
JES2.DISPLAY.SPOOL
JES2.DISPLAY.STC
JES2.DISPLAY.STCOUT
JES2.DISPLAY.TRACE
JES2.DISPLAY.TSU
JES2.DISPLAY.TSUOUT
JES2.DISPLAY.*initstmt*
JES2.GDISPLAY.JOB

Grant *read* access to these resources to any operator who requires it. Logging is not required. The following is a sample command to allow operators with a profile ACID of *opracid* to use the $DS command:

TSS PERMIT(*opracid*) OPRCMDS(*JES2*.DISPLAY.STC) ACCESS(READ)

(3) The following OPERCMDS resources can be used to control access to JES2 system commands that allow the operator to send messages to running jobs and to other members of a multi-access spool (MAS) configuration, or to send commands to remote sites:

JES2.MSEND.MESSAGE
JES2.NSEND.MESSAGE
JES2.SEND.MESSAGE

Grant *read* access to these resources for any operator who is authorized to send such commands and messages. The following is a sample command to allow operators with a profile ACID of *opracid* to use the $DM command:

TSS PERMIT(*opracid*) OPERCMDS(*JES2*.SEND.MESSAGE) -
ACCESS(READ) ACTION(AUDIT)

(4) The following OPERCMDS resources can be used to control access to JES2 system commands that allow the operator to manage the JES2 workload:

JES2.BACKSP.DEV
JES2.CANCEL.AUTOCMD
JES2.CANCEL.BAT
JES2.CANCEL.DEV
JES2.CANCEL.JOB
JES2.CANCEL.JST
JES2.CANCEL.STC
JES2.CANCEL.TSU
JES2.FORWARD.DEV
JES2.GCANCEL.JOB
JES2.GMODIFYHOLD.JOB
JES2.GROUTE.JOBOUT
JES2.HALT.DEV
JES2.INTERRUPT.DEV
JES2.MODIFY.AUTOCMD
JES2.MODIFY.BAT
JES2.MODIFY.BATOUT
JES2.MODIFY.JOB
JES2.MODIFY.JOBOUT
JES2.MODIFY.JST
JES2.MODIFY.JSTOUT
JES2.MODIFY.STC
JES2.MODIFY.STCOUT
JES2.MODIFY.TSU
JES2.MODIFY.TSUOUT
JES2.MODIFYHOLD.BAT
JES2.MODIFYHOLD.JOB
JES2.MODIFYHOLD.JST

JES2.MODIFYHOLD.STC
JES2.MODIFYHOLD.TSU
JES2.MODIFYRELEASE.BAT
JES2.MODIFYRELEASE.JOB
JES2.MODIFYRELEASE.JST
JES2.MODIFYRELEASE.STC
JES2.MODIFYRELEASE.TSU
JES2.RELEASE.BATOUT
JES2.RELEASE.JOBOUT
JES2.RELEASE.JSTOUT
JES2.RELEASE.STCOUT
JES2.RELEASE.TSUOUT
JES2.REPEAT.DEV
JES2.RESTART.DEV
JES2.ROUTE.JOBOUT
JES2.START.DEV
JES2.STOP.BAT
JES2.STOP.DEV
JES2.STOP.JOB
JES2.STOP.JST
JES2.STOP.STC
JES2.STOP.TSU

Permit *update* access to these resources for those operators responsible for managing the corresponding type of work. However, if an operator is required to change or cancel automatic commands issued by a different operator, grant that operator *control* access to *JES2*.CANCEL.AUTOCMD and *JES2*.MODIFY.AUTOCMD. The following is a sample command to allow operators with a profile ACID of *opracid* to use the $B command:

```
TSS PERMIT(opracid) OPERCMDS(JES2.BACKSP.DEV) -
ACCESS(UPDATE) ACTION(AUDIT)
```

(5) The following **OPERCMDS** resources can be used to control access to JES2 system commands that control JES2 facilities:

JES2.HALT.AUTOCMD
JES2.HALT.DEV
JES2.HALT.INITIATOR
JES2.MODIFY.INITIATOR
JES2.MODIFY.LINE
JES2.MODIFY.LOGON
JES2.MODIFY.MEMBER
JES2.MODIFY.NUM
JES2.MODIFY.OFF
JES2.MODIFY.OFFLOAD
JES2.MODIFY.RMT

JES2.MODIFY.SYS
JES2.RESTART.LOGON
JES2.STOP.INITIATOR
JES2.STOP.LINE
JES2.STOP.LOGON
JES2.STOP.RMT
JES2.STOP.SPOOL
JES2.STOP.SYS
JES2.STOP.TRACE
JES2.START.AUTOCMD
JES2.START.INITIATOR
JES2.START.LINE
JES2.START.LOGON
JES2.START.NET
JES2.START.RMT
JES2.START.SPOOL
JES2.START.SYS
JES2.START.TRACE
JES2.VS

Permit *control* access to these resources for those operators responsible for managing the corresponding type of work. The following is a sample command to allow operators with a profile ACID of *opracid* to use the $ZI command:

TSS PERMIT(*opracid*) OPERCMDS(*JES2*.HALT.INITIATOR) -
ACCESS(CONTROL) ACTION(AUDIT)

(6) The following OPERCMDS resources can be used to control access to JES2 system commands that control JES2 facilities:

JES2.ADD.APPL
JES2.ADD.CONNECT
JES2.ADD.DESTID
JES2.ADD.FSS
JES2.ADD.RMT
JES2.HALT.SPOOL
JES2.MODIFY.NODE
JES2.MODIFY.SSI
JES2.MODIFY.*initstmt*
JES2.RESTART.BAT
JES2.RESTART.MEMBER
JES2.RESTART.SYS

Permit *control* access to these resources for those operators responsible for managing the corresponding type of work. However, restrict access to these commands to senior personnel. Impose policies and procedures to ensure that these commands are only used at the direction of network and systems personnel responsible for JES2. The following is a sample command to allow operators with a profile ACID of *opracid* to use the $ZSPOOL command:

TSS PERMIT(*opracid*) OPERCMDS(*JES2*.HALT.SPOOL) -
ACCESS(CONTROL) ACTION(AUDIT)

5.4.6 Security Controls for Job Submission, Naming, and Control

The TSO SUBMIT and CANCEL commands are controlled via resources in the JESJOBS class. Name the actual owning ACID specified for *deptacid* in accordance with installation recommendations.

(1) The following commands may be used to establish default protection for resources defined to the **JESJOBS** resource class:

TSS ADDTO(*deptacid*) JESJOBS(CANCEL.)
TSS ADDTO(*deptacid*) JESJOBS(SUBMIT.)

(2) Permit *alter* access to CANCEL.*localnodeid.userid.jobname* for those users allowed to cancel the job. Use generic profiles (wild cards) as much as possible for this purpose. The permissions granted will take into account installation conventions for job names.

TSS PERMIT(*opracid*) JESJOBS(CANCEL.*localnodeid.userid.jobmask*) -
ACCESS(ALTER) ACTION(AUDIT)

(3) Permit *read* access to SUBMIT.*localnodeid.jobname.userid* for those users allowed to submit the job. Use generic profiles (wild cards) as much as possible for this purpose. The permissions granted will take into account installation conventions for job names.

TSS PERMIT(*opracid*) JESJOBS(SUBMIT.*localnodeid.jobmask.userid*) -
ACCESS(READ) ACTION(AUDIT)

5.4.7 Security Controls for Surrogate Users

Apply the following recommendations when implementing security for surrogate users:

(1) Allowing a user to be a surrogate for another user gives that user indirect access to the resources available to the execution user. For this reason, grant surrogate permission to the minimum number of personnel required for running production jobs.

(2) The following command may be used to establish default protection for resources defined to the SURROGAT resource class:

```
TSS ADDTO(deptacid) SURROGAT(executionuserid.SUBMIT)
```

(3) Grant *read* access to *executionuserid*.SUBMIT for the profile ACID of each authorized surrogate user:

```
TSS PERMIT(profileacid) SURROGAT(executionuserid.SUBMIT) -
ACCESS(READ) ACTION(AUDIT)
```

5.4.8 Security Controls for Remote Processing

Implement the following controls for JES2 Remote Processing:

(1) Enable ACP control of NJE nodes and RJE workstations by defining resources in the IBMFAC resource class:

```
TSS ADDTO(deptacid) IBMFAC(NJE.nodename)
TSS ADDTO(deptacid) IBMFAC(RJE.workstation-id)
```

Access rights do not need to be granted to these resources. JES2 only tests to ensure they are active.

(2) Define profiles in the NODES resource class in accordance with installation policy. A later revision of this *STIG* will specify guidelines for those policies when more data is available. The following are sample commands to allow entry of jobs with a user ACID of *userid* from NJE node *nodename*. The jobs will have explicit userids and passwords.

```
        TSS ADDTO(deptacid) NODES(nodename.USERJ.userid)
TSS PERMIT(ALL) NODES(nodename.USERJ.userid) ACCESS(READ)
        ACTION(AUDIT)
```

6. SESSION MANAGERS

6.1 General Considerations

The main function of any Session Manager is to secure access to both the network and the applications running inside the network. The Session Manager acts as a welcome application and front-end security system for the network. It gathers and checks all security-related information about any user who tries to access the network. In general, users are required to provide a userid and password. However, additional information (e.g., LU name, account number, etc.) may also be required.

Many Session Managers are available. IBM's NetView Access Services and CL/GATEWAY for MVS from the Candle Corporation are two popular Session Managers. All Session Managers are capable of performing I&A validation, and of regulating to which applications a user is granted access. Most are capable of interfacing with a system ACP. This STIG has directed that all terminals that are not directly attached use a Session Manager for access.

Apply the following recommendations to every network Session Manager in use on DOD Mainframe systems:

(1) The Session Manager interfaces with the system ACP (e.g., ACF2, RACF, TOP SECRET) via the SAF interface to perform security I&A validation.

(2) Only valid users identified to the ACP are granted access to the network, and access to log on to applications in the network. Security information registered within the Session Manager's own internal security control feature should not be used for I&A validation.

(3) The Session Manager restricts each individual user's access only to the applications each user is authorized to use as defined in the user's security profile. Only those authorized applications should be displayed to the user.

(4) Wherever possible, access to applications should be arbitrated based on information maintained by the system ACP. The security profile maintained within the network Session Manager should only be used to arbitrate application access where use of the ACP is impossible.

(5) The Session Manager will generate SMF records that will then be used to provide audit trails and accounting reports relative to user logon/logoff activity.

- *(ZVTM0015: CAT II) The Systems Programmer and IAO will ensure that the Session Manager generates SMF records for audit trail and accounting reports***.**

(6) The Session Manager should display a logon banner to the user according to the requirements mandated.

It is important to emphasize that in networks controlled by VTAM, the Session Manager plays only a part in network security because it mainly performs user verification. Any full implementation includes other VTAM-based facilities such as the following:

- VTAM Definitions (e.g., LOGAPPL, USSTAB)
- VTAM Session Management Exit (SME)

These facilities are needed to control the initial network entry points, and to control the SNA session verification and authorization at the VTAM level. Refer to *Section 4.1, Virtual Telecommunications Access Method (VTAM)*, for a more comprehensive discussion on VTAM security.

6.2 CL/SUPERSESSION

CL/SUPERSESSION is the Session Manager offering from the Candle Corporation.

Use the following recommendations to secure CL/SUPERSESSION:

(1) I&A validation should be accomplished through the supplied interface to the ACP. External security should be installed as outlined in the specific Access Control Product sections that follow.

(2) The product is available both as a stand-alone purchase and bundled with the OMEGAMON suite of System Performance Monitors. The individual ACP sections that follow make reference to installing the appropriate ACP interface exit. The ACP exit is only installed once for the entire Candle suite. If the OMEGAMON suite already is installed, it is not necessary to re-install the exit for this product.

(3) The CL/SUPERSESSION Administrator is defined to the product during installation. This individual is responsible for setting and changing numerous installation options. The Administrator function should be the responsibility of the appropriate systems software group. Because this privilege is not arbitrated by the ACP, its assignment and use should be closely monitored by the IAO.

By default, the first user to log on to the product is assigned Administrator authority. This authority should have to be reviewed and updated as necessary to ensure that Administrator authority is delegated to the appropriate individuals (i.e., not necessarily to the first system programmer to log on to CL/SUPERSESSION). In addition, the product should have the system ***emergency userids*** (refer to *Section 3.1.2.6, Emergency Userids,* for further information) assigned as Administrators to recover from **No Administrator defined** situations.

(4) Users may be allowed to switch an active CL/SUPERSESSION session from one VTAM terminal to another at the discretion of the site.

(5) VTAM sessions are not allowed to remain active when the associated CL/SUPERSESSION session terminates.

(6) Users should not be allowed to change the **hot keys** associated with their sessions. The ability to change **hot key** definitions can allow a user to change the dialog name associated with a trigger, and can result in access to restricted dialogs.

(7) Many options can be set both globally and at the group or user level. Several of the options have significance relating to security, and appear on various customization screens in the product. Those options are listed below with the STIG required values:

- *(ZCLS0011: CAT II) The IAO will ensure that CL/SUPERSESSION profile options are set to the values specified in the following table:*

Table A-56. REQUIRED COMMON PROFILE OPTIONS (6.2 a)

REQUIRED COMMON PROFILE OPTIONS		
OPTION	DESCRIPTION	REQUIRED VALUE
Administrator authority	Grants Administrator authority	N
Customized menu	Allows the user to customize the application menu	Y
Add sessions	Allows the user to add VTAM sessions to the application menu	N
NOTE: The above options may be set to **Yes** only for the Administrator(s).		
Resource validation	Resource name (APPLID and/or Session Id) used when calling the ACP for dynamic application lists	A
Timeout interval	Interval after which the user's session should be terminated for inactivity	00:15
Group profile	Associated group profile	Will only be specified in a user level profile

Table A-57. REQUIRED SUPSESS PROFILE OPTIONS (6.2 b)

REQUIRED SUPSESS PROFILE OPTIONS		
OPTION	DESCRIPTION	REQUIRED VALUE
Maintain trigger	Allows the user to save changes to the trigger (*hot-key* string) profile when logging off	N
Add triggers	Allows the user to create trigger definitions and add them to the trigger (*hot-key* string) profile	N
Modify triggers	Allows the user to modify existing trigger definitions in the trigger (*hot-key* string) profile	N
Switch terminals	Allows switching an active CL/SUPERSESSION session to another VTAM terminal	Y
Preserve sessions	Allows VTAM application sessions to remain active (until they time out) if the CL/SUPERSESSION session is terminated for some reason (e.g., switching to another CL/SUPERSESSION at another site or host)	N

(8) Use dynamic application lists to ensure that the ACP arbitrates access to all applications.

(9) A warning banner should be issued from CL/SUPERSESSION.

(10) To provide an audit trail of user activity in CL/SUPERSESSION, configure the Network Accounting Facility (NAF) to require SMF recording of accounting and audit data. Accounting to the journal data set is optional at the discretion of the site. To accomplish this, configure the following NAF startup parameters in the **KLVINNAF** member of the **RLSPARM** initialization parameter library as follows:

DSNAME= *dsname* Name of the NAF journal data set. Required only if the site is collecting accounting and audit data in the journal data set in addition to the SMF data.

MOD If the journal data set is used, this parameter should be set to ensure that logging data in the data set is not overwritten.

SMF=*nnn* SMF record number. This field is mandatory to ensure that data are always written to the SMF files.

- *(ZCLS0012: CAT II) The Systems Programmer and IAO will ensure that the Session Manager generates SMF records for audit trail and accounting reports**.***

6.2.1 ACF2

Use the following controls to implement ACF2 security:

(1) During final product configuration (CICAT), specify ACF2 External Security.

(2) Use the following guidelines to ensure that protection of the data sets used for CL/SUPERSESSION is properly in place:

(a) Write data set rules protecting **SYS2.OMEGAMON.***, **SYS2A.OMEGAMON.***, and **SYS3.OMEGAMON.***, allowing only the KLS logonid and the appropriate Systems staff *read*, *write*, and *allocate* access.

(b) Write data set rules protecting specifically the APF-authorized data sets suffixed with **TLSLOAD**, **TLVLOAD**, and **RLSLOAD**. Limit *read* access to the KLS logonid, the necessary Systems staff, Security personnel, and the Audit staff. Log all *write* and *allocate* access to these libraries.

(3) Ensure member **KLVINNAM** in **SYS3.OMEGAMON.*qualifier*.RLSPARM** is configured to read as follows:

```
DEFAULT DSNAME(SYS3.OMEGAMON.qualifier.RLSNAM)
        EXIT=KLSA2NEV NORACF NODB CLASSES=APPCLASS
```

- *(ZCLS0014: CAT II) The Systems Programmer and IAO will ensure that the parameter options for member KLVINNAM are coded to the above specifications.*

(4) Assemble and link the exit for ACF2 security validation **KLSA2NEV** with **AC=1** and **AMODE=24** into **SYS3.OMEGAMON.*qualifier*.RLSLOAD**.

(5) Ensure member **APPCLASS** in **SYS3.OMEGAMON.*qualifier*.RLSPARM** is configured to specify the ACF2 resource type **APL** that is mapped to the **APPL** resource class as follows:

```
VGWAPLST EXTERNAL=APL
```

- *(ZCLS0016: CAT II) The Systems Programmer and IAO will ensure that the parameter options for member APPCLSS are coded to the above specifications**.***

(6) Define the logonid **KLS** for the CT/Engine started task with the following attributes:

```
INSERT KLS STC MUSASS NO-SMC
```

- *(ZCLS0018: CAT II) The Systems Programmer and IAO will ensure that the started task for CL/SUPPERSESSION is properly defined.*

(7) With dynamic application lists enabled, only applications authorized to the user are displayed on the session menu at the terminal. Ensure that all applications defined to CL/SUPERSESSION are defined to **TYPE(APL)** with a default access of *prevent*. The resource names are VTAM network names (APPLIDs). Permit access to applications based on users' responsibilities and roles. For example:

```
$KEY(applid) TYPE(APL)
- UID(-) PREVENT
UID(authorized-group) ALLOW
```

6.2.2 RACF

Use the following controls to implement RACF security:

(1) During final product configuration (CICAT), specify RACF External Security.

(2) Create a matching **STARTED** resource class profile for the CT/Engine associating it with the userid **KLS.**

```
RDEFINE STARTED KLS.* UACC(NONE) OWNER(admin)
    STDATA(USER(=MEMBER) GROUP(STCOMG) TRUSTED(NO))
```

- *(ZCLSR024: CAT II) The IAO will ensure that CL/SUPPERSESSION is defined to the STARTED resource class with the TRUSTED(NO) attribute.*

(3) Create a **PROTECTED** userid for the CT/Engine started task called **KLS**:

```
AU KLS NAME('CL/SUPERSESSION STC') NOPASSWORD
OWNER(admin) DFLTGRP(STCOMG)
```

- *(ZCLS0018: CAT II) The Systems Programmer and IAO will ensure that the started task for CL/SUPPERSESSION is properly defined.*

(4) Use the following guidelines to ensure that protection of the data sets used for CL/SUPERSESSION is properly in place:

(a) Create a data set profile protecting **SYS2.OMEGAMON.***, **SYS2A.OMEGAMON.***, and **SYS3.OMEGAMON.***, allowing only the KLS userid and the appropriate Systems staff *read*, *update*, and *alter* access. Restrict *control* access to the NAM, VIEWLOG, and TABLEDB VSAM data sets to the KLS userid and systems programming staff.

(b) Write data set rules protecting specifically the APF-authorized data sets suffixed with **TLSLOAD**, **TLVLOAD**, and **RLSLOAD**. Limit *read* access to the KLS logonid, the necessary Systems staff, Security personnel, and the Audit staff. Log all *update* and *alter* access to these libraries.

(5) Modify the security system definition in **SYS3.OMEGAMON.*qualifier*.RLSPARM(KLVINNAM)** to implement RACF:

```
DEFAULT DSNAME(SYS3.OMEGAMON.qualifier.RLSNAM)
RACF NODB CLASSES=APPCLASS
```

- *(ZCLS0014: CAT II) The Systems Programmer and IAO will ensure that the parameter options for member KLVINNAM are coded to the above specifications.*

(6) Ensure **SYS3.OMEGAMON.*qualifier*.RLSPARM(APPCLASS)** is configured to specify the **APPL** resource class name as follows:

```
VGWAPLST EXTERNAL=APPL
```

- *(ZCLS0016: CAT II) The Systems Programmer and IAO will ensure that the parameter options for member APPCLSS are coded to the above specifications.*

(7) Ensure that the **APPL** class is active:

```
SETROPTS CLASSACT(APPL)
```

- *(ZCLSR021: CAT II) The Systems Programmer and IAO will ensure that the APPL class is active.*

(8) With dynamic application lists enabled, only applications authorized to the user are displayed on the session menu at the terminal. Ensure that all applications defined to CL/SUPERSESSION are defined to the **APPL** resource class with a default access of *none*. The resource names are VTAM network names (APPLIDs). For example:

```
RDEFINE APPL applid UACC(NONE) OWNER(admin)
```

(9) Permit access to applications based on users' responsibilities and roles. For example:

```
PE applid CLASS(APPL) ID(authorized-group)
```

6.2.3 TOP SECRET

Use the following controls to implement TOP SECRET security:

(1) During final product configuration (CICAT), specify TOP SECRET External Security.

(2) Use the following guidelines to ensure that protection of the data sets used for CL/SUPERSESSION is properly in place:

(a) Permit access to **SYS2.OMEGAMON.***, **SYS2A.OMEGAMON.***,and **SYS3.OMEGAMON.***, allowing only the **KLS** ACID and the appropriate Systems staff *read*, *update*, *scratch*, or *all* access.

(b) Permit access specifically to the APF-authorized data sets suffixed with **TLSLOAD**, **TLVLOAD**, and **RLSLOAD**. Limit *read* access to the **KLS** ACID, the necessary Systems staff, Security personnel, and the Audit staff. Log all *update* and *scratch* access to these libraries.

(3) Define the CT/Engine started task name **KLS** as a Facility to TOP SECRET in the Facility Matrix Table using the following example:

```
* KLS    CL/SUPERSESSION
FACILITY(USERxx=NAME=KLS)
FACILITY(KLS=MODE=FAIL,ACTIVE,SHRPRF)
FACILITY(KLS=PGM=KLV,NOASUBM,NOABEND,NOXDEF)
FACILITY(KLS=ID=xx,MULTIUSER,RES,LUMSG,STMSG,WARNPW,SIGN(M))
FACILITY(KLS=NOINSTDATA,NORNDPW,AUTHINIT,NOPROMPT,NOAUDIT)
FACILITY(KLS=NOTSOC,LOG(INIT,SMF,MSG,SEC9))
```

- *(ZCLST022: CAT II) The Systems Programmer and IAO will ensure that KLS started task name is defined in the Facility Matrix Table.*

(4) Create an ACID called **KLS** for the CT/Engine STC:

```
TSS CRE(KLS) DEPT(Dept) NAME('CL/SUPERSESSION')
  FAC(STC) MASTFAC(KLS) PASSWORD(password,0)
  SOURCE(INTRDR)
```

- *(ZCLS0018: CAT II) The Systems Programmer and IAO will ensure that the started task for CL/SUPPERSESSION is properly defined.*

(5) Add the **KLS** started task ACID to the Started Task Table:

```
TSS ADD(STC) PROCNAME(KLS) ACID(KLS)
```

- *(ZCLST024: CAT II) IAO will ensure that CL/SUPPERSESSION is defined to the started task table.*

(6) Modify the security system definition in **SYS3.OMEGAMON.*qualifier*.RLSPARM(KLVINNAM)** to implement TOP SECRET:

```
DEFAULT DSNAME(SYS3.OMEGAMON.qualifer.RLSNAM)
EXIT=KLSTSNEV RACF  NODB CLASSES=APPCLASS
```

- *(ZCLS0014: CAT II) The Systems Programmer and IAO will ensure that the parameter options for member KLVINNAM are coded to the above specifications***.**

(7) Assemble and link the exit for TOP SECRET security validation **KLSTSNEV** with **AC=1** and **AMODE=24** into **SYS3.OMEGAMON.*qualifier*.RLSLOAD**.

(8) Ensure **SYS3.OMEGAMON.*qualifer*.RLSPARM(APPCLASS)** is configured to specify the CT/Engine Facility name **KLS** as follows:

```
VGWAPLST EXTERNAL=KLS
```

- *(ZCLS0016: CAT II) The Systems Programmer and IAO will ensure that the parameter options for member APPCLSS are coded to the above specifications***.**

(9) To implement dynamic application lists , add the resource **KLS** to the TOP SECRET RDT. **KLS** is the value specified in the **EXTERNAL=** parameter in Step 8:

```
TSS ADD(RDT) RESCLASS(KLS) RESCODE(xx)
```

(where **xx** is an unused hex value)

- *(ZCLST026: CAT II) The IAO will ensure that the KLS resource class is defined to the RDT.*

(10) With dynamic application lists enabled, only applications authorized to the user are displayed on the session menu at the terminal. Ensure that all applications defined to CL/SUPERSESSION are defined to the new **KLS** resource class. The resource names are VTAM network names (APPLIDs). Permit access to applications based on users' responsibilities and roles. For example:

```
TSS ADD(dept_acid) KLS(applid)
TSS ADD(authorized-group) KLS(applid)
TSS PERMIT(authorized-group) KLS(applid)
```

6.3 NC-PASS Authenticator

NC-PASS Authenticator, marketed by CKS, is a product designed to protect users from unauthorized use of userids and passwords. This product was purchased as a remedy to the security exposure introduced by network Sniffers. These Sniffers are programs and/or hardware devices that have the capability to monitor communication links and intercept a user's userid and password without detection. The implementation of this product is used in conjunction with the

SecurID card from Security Dynamics to implement a check for a pseudo random number (PRN) that is unique to each card and that changes every minute. This affords an additional level of user authentication to complement the use of the userid and password.

The design structure for this particular implementation ensures that the user has authenticated with their SecurID card within the previous fifteen (15) minutes of an attempt to log on to an interactive application. An exit has been placed in each ACP at the point on initial user entry validation. The exit program queries NC-PASS to determine if the user has been appropriately validated. If the user has not, the logon attempt is terminated by the exit. If the user has, then the normal ACP initialization processing is allowed to proceed. This same logic is used within all three ACPs. However, the mechanics of the implementations are unique to each ACP.

A seed record is loaded into the NC-PASS database using the product's PSBATCH job stream for each SecurID card to be used. The key to the record is the serial number of the actual card itself. User profiles are created within NC-PASS that identify the actual userid for a given user and the SecurID card assigned to that user.

As each user profile is defined to NC-PASS, the Administrator specifies the appropriate password validation type based on the installed security system as follows:

ACF2	-	Type A
RACF	-	Type R
TSS	-	Type T

A trigger mechanism is set in the ACP for each userid that requires the extended authentication. The ACP exit checks the trigger mechanism for the user logging on and, if set, calls NC-PASS to determine if they have validated with NC-PASS within the required five-minute interval. The logon is permitted or denied based on the results of this check.

In addition to the ACP exit programs, an additional revalidation utility program has been developed to allow the user to revalidate to NC-PASS from the user’s TSO session. This utility (**NCPAUTH**) prompts the user for a current PRN, obtains the userid from OS/390 control blocks, and performs re-validation to NC-PASS. This capability currently is only available from TSO. A companion utility for ROSCOE is available on the SSO Mechanicsburg Web site, and one for CL/SUPERSESSION is being considered.

DISA has directed that NC-PASS extended authentication be implemented on all domains. All users with *update* and *alter* access to sensitive system-level data sets and resources, or who possess special security privileges, are required to use NC-PASS for extended authentication. Typical personnel required to use NC-PASS include, but are not limited to, systems programming, security, operations, network/communications, storage management, and production control.

6.3.1 NC-PASS for ACF2

The ACF2 implementation of NC-PASS involves the use of the ACF2 System Entry pre- and post-validation exit points, **SEVPRE** and **SEVPOST,** to perform the necessary processing. The

SEVPRE exit **(SEVPRE01)** is a dummy exit, required by ACF2 to be defined along with the **SEVPOST** exit, and performs no processing. The **SEVPOST** exit **(SEVPST01)** performs all the necessary processing. The **AUTHSUP1** privilege identifies users requiring validation by NC-PASS. This privilege field was selected because it is a standard ACF2 field, currently unused by any site. This avoids the necessity of further modifying ACF2 to define an addition field. The use of this field requires the specification of an Extended User Authentication **(EUA)** exit. This **EUA** exit **(AUTHXNCP)** is a dummy exit that performs only the minimal processing, setting a required control block flag and issuing the appropriate return code. The following steps outline the security definitions required to install and utilize the NC-PASS Authenticator product in an ACF2 environment.

(1) Create a valid STC userid and define the appropriate access rules for the NC-PASS distribution and installation libraries.

(2) Create the logonid for the NC-PASS STC:

```
INSERT NCPASS STC MUSASS MUSUPDT NO-SMC
```

(3) Create/modify access rule sets:

```
$KEY(DISA01)
- UID(user) R(A) W(A) A(A) E(A)              Systems and Security only
- UID(*) R(P) W(P) A(P) E(P)                 All others

$KEY(SYS2)
NCPASS.LOAD UID(NCPASS) R(A) W(P) A(P) E(A)      NC-PASS Started Task
NCPASS.LOAD UID(user) R(A) W(L) A(L) E(A)        Systems and Security only
NCPASS.LOAD UID(user) R(A) W(P) A(P) E(P)        CA-EXAMINE
Users/Auditors
NCPASS.LOAD UID(*) R(P) W(P) A(P) E(P)           All others
*
NCPASS.- UID(NCPASS) R(A) W(A) A(P) E(P)         NC-PASS Started Task
NCPASS.- UID(user) R(A) W(A) A(A) E(A)           Systems and Security only
NCPASS.- UID(user) R(A) W(P) A(P) E(P)           Auditors
NCPASS.- UID(*) R(P) W(P) A(P) E(P)              All others

$KEY(SYS3)
NCPASS.LOAD UID(NCPASS) R(A) W(P) A(P) E(A)  NC-PASS Started Task
NCPASS.LOAD UID(user) R(A) W(L) A(L) E(A)     Systems and Security only
NCPASS.LOAD UID(user) R(A) W(P) A(P) E(A)    CA-EXAMINE Users/Auditors
NCPASS.LOAD UID(*) R(A) W(P) A(P) E(A)       All others
*
NCPASS.- UID(NCPASS) R(A) W(A) A(P) E(P)     NC-PASS Started Task
NCPASS.- UID(xxxxxx) R(A) W(A) A(A) E(A)     Systems and Security only
NCPASS.- UID(xxxxxx) R(A) W(P) A(P) E(P)     Auditors
NCPASS.- UID(*) R(P) W(P) A(P) E(P)          All others
```

```
$KEY(SYS3A)
NCPASS.LPALIB UID(xxxxxx) R(A) W(L) A(L) E(A)    Systems and Security only
NCPASS.LPALIB UID(xxxxxx) R(A) W(P) A(P) E(P)    CA-EXAMINE Users/Auditors
NCPASS.LPALIB UID(*) R(P) W(P) A(P) E(P)         All others (incl. NC-PASS)
*
NCPASS.- UID(NCPASS) R(P) W(P) A(P) E(P)         NC-PASS Started Task
NCPASS.- UID(xxxxxx) R(A) W(A) A(A) E(A)         Systems and Security only
NCPASS.- UID(xxxxxx) R(A) W(P) A(P) E(P)         Auditors
NCPASS.- UID(*) R(P) W(P) A(P) E(P)              All others
```

(4) Enable APF authorization for the following data sets by adding them to **SYS1.PARMLIB** member **IEAAPFxx** or **PROGxx** as appropriate:

```
SYS2.NCPASS.LOAD
SYS3.NCPASS.LOAD
SYS3A.NCPASS.LPALIB
```

(5) Add/change the GSO **EXITS** record to specify the System Entry Validation exits. It will include the following:

```
SEVPRE(SEVPRE01) SEVPOST(SEVPST01)
```

(6) Add a GSO **AUTHEXIT.001** record as follows:

```
LIDFIELD(AUTHSUP1) PROCPGM(AUTHXNCP) NOINFOSTG
```

(7) To ensure that a user requires NC-PASS validation, the appropriate trigger is set for the userid. For each user who requires extended authentication, perform the following action:

Add the **AUTHSUP1** field to the user's logonid record:

CHANGE *xxxxxxx* AUTHSUP1

- *(ZNCP0011: CAT II) The Systems Programmer and IAO will ensure that the EXIT for NC-PASS is properly installed and defined.*

- *(ZNCP0020: CAT II) The IAO will ensure that the started task for NC-PASS is properly defined.*

- *(ZNCP0030: CAT II) The IAO will ensure that sensitive users are proprerly validated to NC-PASS.*

- *(ZNCP0040: CAT II) The IAO will ensure that Data set access authorization restricts UPDATE and/or ALLOCATE access to systems programming personnel and/or security personnel justification for any other access must be documented.*

6.3.2 NC-PASS for RACF

The RACF implementation of NC-PASS involves the use of the RACF pre- and post-initialization exit programs, **ICHRIX01** and **ICHRIX02**, to perform the necessary processing. The **SECURID** connect group identifies users requiring validation by NC-PASS. The following steps outline the security definitions required to install and utilize the NC-PASS Authenticator product in a RACF environment:

(1) Create a STC userid for the NC-PASS started task. The userid will be defined as a **PROTECTED** userid. For example:

AU NCPASS NAME('NCPASS, STC, SMRTCRD') NOPASSWORD
OWNER(*admin*) DFLTGRP(STC)

(2) Ensure the NC-PASS STC has a matching profile defined to the **STARTED** resource class. The following command can be used to define the profile:

RDEFINE STARTED NCPASS.* UACC(NONE) OWNER(*admin*)
STDATA(USER(=MEMBER) GROUP(STC) TRUSTED(NO))

(3) Add the Group name for the tape data set profile for product distribution:

```
AG DISA01 OWNER(ADMIN) SUPGROUP(ADMIN)
```

(4) Add the rule for the tape data set profile:

```
AD 'DISA01.**' UACC(NONE)
```

(5) Add the rule for NC-PASS non-customized data sets:

```
AD 'SYS2.NCPASS.**' UACC(NONE)
```

(6) Add the rule for NC-PASS customized data sets:

```
AD 'SYS3.NCPASS.**' UACC(NONE)
AD 'SYS3A.NCPASS.**' UACC(NONE)
```

(7) Grant permissions to the NC-PASS data sets to personnel as required:

Systems and Security:

PE 'DISA01.**' ID(*xxxxxx*) ACC(ALTER)
PE 'SYS2.NCPASS.**' ID(*xxxxxx*) ACC(ALTER)
PE 'SYS3.NCPASS.**' ID(*xxxxxx*) ACC(ALTER)
PE 'SYS3A.NCPASS.**' ID(*xxxxxx*) ACC(ALTER)

NC-PASS Started Task:

```
PE 'SYS2.NCPASS.**' ID(NCPASS) ACC(UPDATE)
PE 'SYS2.NCPASS.LOAD' ID(NCPASS) ACC(READ)
PE 'SYS3.NCPASS.**' ID(NCPASS) ACC(UPDATE)
PE 'SYS3.NCPASS.LOAD' ID(NCPASS) ACC(READ)
```

CA-EXAMINE Users/Auditors:

PE 'SYS2.NCPASS.LOAD' ID(*xxxxxx*) ACC(READ)
PE 'SYS3.NCPASS.LOAD' ID(*xxxxxx*) ACC(READ)
PE 'SYS3A.NCPASS.LPALIB' ID(*xxxxxx*) ACC(READ)

All Others:

PE 'SYS3.NCPASS.LOAD' ID(*xxxxxx*) ACC(READ)

(8) Add the Group that will be checked to determine if a user requires extended authentication:

```
AG SECURID OWNER(ADMIN) SUPGROUP(ADMIN)
DATA('GROUP CONTAINING USERS REQUIRING USE OF A CARD')
```

(9) Enable APF authorization for the following data sets by adding them to **SYS1.PARMLIB** member **IEAAPFxx** or **PROGxx** as appropriate:

```
SYS2.NCPASS.LOAD
SYS3.NCPASS.LOAD
SYS3A.NCPASS.LPALIB
```

(10) To ensure that a user requires NC-PASS validation, a trigger is set for the userid. For each user who requires extended authentication perform the following action:

Connect the user to the **SECURID** Connect Group:

CO *xxxxxxx* GROUP(SECURID)

- *(ZNCP0011: CAT II) The Systems Programmer and IAO will ensure that the EXIT for NC-PASS is properly installed and defined.*

- *(ZNCP0020: CAT II) The IAO will ensure that the started task for NC-PASS is properly defined.*

- *(ZNCP0030: CAT II) The IAO will ensure that sensitive users are proprerly validated to NC-PASS.*

- *(ZNCP0040: CAT II) The IAO will ensure that Data set access authorization restricts UPDATE and/or ALLOCATE access to systems programming personnel and/or security personnel justification for any other access must be documented.*

- *(ZNCPR050: CAT II) The IAO will ensure that the started task for NC-PASS is defined to the STARTED resource class.*

6.3.3 NC-PASS for TOP SECRET

The TOP SECRET implementation of NC-PASS involves the use of the post-initialization **(POSTINIT)** entry point of the TOP SECRET exit program **(TSSINSTX)**. Code is inserted at this entry point to invoke the DISA exit program **(NCPPOSTI)** to perform the necessary processing. The code inserted into **TSSINSTX** terminates or continues the initialization process based on the return code from the **NCPPOSTI** program. Access to log on to NC-PASS for validation is controlled by access to the **NCPASS FAC**ility. Requiring users to log on through NC-PASS is enforced by access to the **SECURID ABS**tract. The following steps outline the security definitions required to install and utilize the NC-PASS Authenticator product in a TOP SECRET (TSS) environment:

(1) Create a valid STC userid and define the appropriate access rules for the NC-PASS distribution and installation libraries.

(2) Define a **NCPASS** to the Facilities Matrix Table. Include other data as pertinent to the site.

```
**** NCPASS
FACILITY(USERxx=NAME=NCPASS)
FACILITY(NCPASS=PGM=NCS,ID=NP)
FACILITY(NCPASS=ACTIVE,NOABEND,NOASUBM,AUTHINIT)
FACILITY(NCPASS=MULTIUSER,NORNDPW,NOTSOC,NOXDEF)
FACILITY(NCPASS=LOG(INIT,SMF,MSG,SEC9))
FACILITY(NCPASS=DOWN=GLOBAL,LOCKTIME=00,DEFACID(*NONE*))
*
```

(3) Create an ACID for the started task called **NCPASS** with **FAC**ility STC, and a Master **FAC**ility of **NCPASS**. Include other data as pertinent to the site.

```
TSS CRE(NCPASS) DEPT(Dept) NAME('Descriptive name')
    FAC(STC) MASTFAC(NCPASS) PASSWORD(password,0)
    SOURCE(INTRDR)
```

The following is a *sample* NC-PASS ACID definition:

```
ACCESSORID        = NCPASS     NAME= *STC* - NCPASS/SMARTCARD
TYPE        = USER        SIZE   = 512  BYTES
FACILITY          = STC
DEPT ACID         = STCDEPT    DEPARTMENT = STC DEPT
DIV ACID    = TECHDIV   DIVISION      = TECH SUPPORT DIVISION
ZONE ACID   = SOFTZONE  ZONE          = SOFTWARE ZONE
CREATED     = 12/16/96  LAST MOD      = 04/09/97  09:42
LAST USED   = 04/09/97 10:36 CPU(WP37) FAC(STC     ) COUNT(00018)
MASTER FAC        = NCPASS
```

(4) Define the NC-PASS started task to TSS:

```
TSS ADD(STC) PROCNAME(NCPASS) ACID(NCPASS)
```

(5) Define abstract **SECURID** to TOP SECRET:

```
TSS ADD(Dept) ABSTRACT(SECURID)
```

(6) Grant permissions to the NC-PASS data sets to personnel as required:

Systems and Security:

```
TSS PERMIT(profile) DSN(DISA01.) ACCESS(ALL)
TSS PERMIT(profile) DSN(SYS2.NCPASS.) ACCESS(ALL)
TSS PERMIT(profile) DSN(SYS2.NCPASS.LOAD) ACCESS(READ)
TSS PERMIT(profile) DSN(SYS2.NCPASS.LOAD) ACCESS(ALL)
    ACTION(AUDIT)

TSS PERMIT(profile) DSN(SYS3.NCPASS.) ACCESS(ALL)
TSS PERMIT(profile) DSN(SYS3.NCPASS.LOAD) ACCESS(READ)
TSS PERMIT(profile) DSN(SYS3.NCPASS.LOAD) ACCESS(ALL)
    ACTION(AUDIT)

TSS PERMIT(profile) DSN(SYS3A.NCPASS.) ACCESS(ALL)
TSS PERMIT(profile) DSN(SYS3A.NCPASS.LOAD) ACCESS(READ)
TSS PERMIT(profile) DSN(SYS3A.NCPASS.LOAD) ACCESS(ALL)
    ACTION(AUDIT)
```

NC-PASS Started Task:

```
TSS PERMIT(NCPASS) DSN(SYS2.NCPASS.) ACCESS(UPDATE)
TSS PERMIT(NCPASS) DSN(SYS2.NCPASS.LOAD) ACCESS(READ)

TSS PERMIT(NCPASS) DSN(SYS3.NCPASS.) ACCESS(UPDATE)
TSS PERMIT(NCPASS) DSN(SYS3.NCPASS.LOAD) ACCESS(READ)
```

CA-EXAMINE Users/Auditors:

```
TSS PERMIT(profile) DSN(SYS2.NCPASS.LOAD) ACCESS(READ)
TSS PERMIT(profile) DSN(SYS3.NCPASS.LOAD) ACCESS(READ)
TSS PERMIT(profile) DSN(SYS3A.NCPASS.LPALIB) ACCESS(READ)
```

All Others:

```
TSS PERMIT(profile) DSN(SYS3.NCPASS.LOAD) ACCESS(READ)
```

(7) The implementation requires code to be placed in the **POSTINIT** routine of the TOP SECRET Installation Exit (**TSSINSTX**). The local **TSSINSTX** Installation Exit is modified to call the DISA exit program. If the site does not currently have a **TSSINSTX** exit, create one using the sample **TSSINSTX** in the **TSSOPMAT** optional materials data set from CA. If the site does currently have a **TSSINSTX** exit, the existing one should be modified.

(a) Update **TSSINSTX** to enable the **POSTINIT** exit. Change the entry for the post-initialization exit in the Exit Function Activation Table to **AL1(#####YES)** to activate the exit (if not already activated).

(b) Modify **TSSINSTX** by adding the supplied DISA source code to the **POSTINIT** section of the program. Insert the supplied source *before* any existing local **POSTINIT** processing code.

(c) Assemble and Link **TSSINSTX** into **SYS3A.TSS.CAILIB(TSSINSTX)** (or re-link into the current library).

(d) Add **SYS3A.TSS.CAILIB** to the **LNKLST***xx* member in **parmlib**, before the production TOP SECRET Load Library (if not already included in the Linklist).

(8) Enable APF authorization for the following data sets by adding them to **SYS1.PARMLIB** member **IEAAPFxx** or **PROGxx** as appropriate:

SYS2.NCPASS.LOAD
SYS3.NCPASS.LOAD
SYS3A.NCPASS.LPALIB

(9) Ensure that the TOP SECRET Control Option **EXIT** specifies **ON**.

(10) To ensure that a user requires NC-PASS validation, the appropriate trigger is set for the userid. For each user who requires extended authentication, perform the following actions:

(a) Grant all **sensitive** users access to the **NCPASS FAC**ility to authenticate to NC-PASS by logging into the product:

TSS ADD(*user*) FAC(NCPASS)

(b) Ensure that **sensitive** users under the control of NC-PASS are required to authenticate to the product by granting them access to the **SECURID ABS**tract:

TSS PERMIT(*user*) ABS(SECURID)

- *(ZNCP0011: CAT II) The Systems Programmer and IAO will ensure that the EXIT for NC-PASS is properly installed and defined.*

- *(ZNCP0020: CAT II) The IAO will ensure that the started task for NC-PASS is properly defined.*

- *(ZNCP0030: CAT II) The IAO will ensure that sensitive users are proprerly validated to NC-PASS.*

- *(ZNCP0040: CAT II) The IAO will ensure that data set access authorization restricts UPDATE and/or ALLOCATE access to systems programming personnel and/or security personnel justification for any other access must be documented.*

- *(ZNCPT050: CAT II) The IAO will ensure that the TOP SECRET NC-PASS is properly defined to the Facility Matrix Table.*

- *(ZNCPT060: CAT II) The IAO will ensure that the TOP SECRET NC-PASS is properly defined to the Started Task Table.*

7. TERMINAL MONITOR PROGRAMS

7.1 General Considerations

Terminal monitor programs (TMPs) provide users with a program development system capable of accessing and manipulating data residing under the controls of the operating system. TMP products typically execute as an interactive on-line system, but may also run in a batch environment.

Consideration should be given to the logon or sign-on process. When a user is initiating a TMP session, I&A checking should be performed. This is a requirement that provides protection against unauthorized access to a system.

Depending on the TMP product, a System Administrator assigns authorized users specific logon procedures, programs, or profiles. These are used to establish resource availability during a session. General users will not be permitted to modify or change their logon assignments. Permission to do so will be granted only to authorized personnel.

Some TMP products create an individual address space for each authorized user signing on. In this case, standard OS/390 and z/OS safeguards prevent unauthorized access to data and/or resources owned by that address space. However, if the TMP is a multi-user single address space system (MUSASS), all authorized users share the same address space region.

In a MUSASS environment, the MUSASS TMP performs functions (e.g., data retrieval) for the individual users it supports. The TMP itself has total access to all data, as the validation by the ACP should be done against the userid of the MUSASS. In this case, it is the responsibility of the MUSASS TMP product to ensure data integrity and protection within the shared region.

The mechanisms used by the MUSASS to ensure data integrity should be reviewed and evaluated for possible security exposures. It is important to remember that ACP security controls should never be compromised by internal security.

TMP products offer many powerful tools, and controlling access to these tools is crucial. Most of these tools can be accessed by invoking a command, a program, or a facility. These items should be reviewed and protected against unauthorized access. For example, the ability to issue OS/390 and z/OS operator commands should be prohibited from general use and granted only to authorized personnel.

All interfaces should be carefully evaluated for possible security exposures. These include interfaces delivered with the TMP and interfaces offered by other software packages or applications that execute under the TMP, or that use its facilities. These interfaces may offer their own internal security, provide the potential to circumvent ACP security controls, or expose weaknesses in protection. DISA FSO will be included in the evaluation process.

Use the following recommendations when securing access to terminal monitor program systems:

(1) Control access to the software product's data sets, and restrict access only to authorized personnel.

(2) All TMP systems in use at DOD sites perform I&A checking during the logon process. Perform I&A validation using the services of the ACP.

(3) Strictly control logon procedures, programs, or profiles assigned to users. Restrict permission to modify and change user logon assignments only to authorized personnel.

(4) Strictly control access to data sets specified in logon procedures (e.g., panel libraries, **clist** libraries, etc.). Only grant the required level of access to users. Restrict *update* access to those individuals responsible for the maintenance of the product or application with which the library is associated.

(5) In TMP products that execute as a MUSASS, the product's internal security provides data integrity and protection, if it does not compromise ACP security controls.

(6) Restrict user access to commands, programs, and facilities within a TMP session to that necessary for users to accomplish their assigned responsibilities.

(7) Review product and other vendor interfaces for potential security exposures. Document any potential security exposures. Notify IAO and the vendor.

7.2 TSO

Vendor: IBM Corporation

Time Sharing Option (TSO) and its enhanced version, TSO/E, provide facilities to manipulate data residing under the controls of the operating system. An address space is created for each user as the user signs on to TSO so that commands can be processed to manipulate data. When complemented with ISPF, a full menu-driven process is presented to enable users to easily perform maintenance functions.

TSO is a very powerful tool that can pose risks if not properly controlled. System resources can be consumed very quickly, thus reducing the available capacity of the machine for other work. Many vendors provide interfaces with TSO and ISPF. If not properly controlled, this may present data integrity exposures.

The user’s ability to use a certain TSO logon procedure is validated against the **TSOPROC** resource class. Most users should only be capable of executing one standard logon procedure. Limit the ability to specify an alternate logon procedure to those users who have a justified need.

- *(ZTSO0010: CAT II) The IAO will ensure that access to TSO logon procedures is controlled and that access to multiple logon procedures are limited to authorized personnel.*

Controls that specify the authorized procedure name for the TSO session a user may access should be implemented within each ACP. The accounting information should be controlled to verify that proper billing of resource consumption can be accomplished. All TSO attributes are obtained via the ACP TSO information for each user. Entries will not be coded in **SYS1.UADS** for normal usage.

- *(ZTSO0020: CAT II) The IAO will ensure that all USERID entries in SYS1.UADS are justified and documented.*

The TSOAUTH resource class controls sensitive TSO/E commands. The following identifies the TSO/E command to the TSOAUTH resource.

TSO/E COMMAND	TSOAUTH	DESCRIPTION
ACCOUNT	ACCT	Manage the entries in the user attribute data set (SYS1.UADS) and in the broadcast data set
CONSOLE	CONSOLE	Issue MVS system and subsystem commands
MOUNT	MOUNT	Results the mounting of a volume
OPERATOR	OPER	Regulate and maintain TSO/E from a terminal
PARMLIB	PARMLIB	Display and dynamically change the active **IKJTSO*xx*** member

- *(ZTSO0030: CAT II) The IAO will ensure that the MOUNT resource is assigned only on an as needed basis for userids associated with STCs and LOGONIDS that need to execute TSO in batch.*

- *(ZTSO0030: CAT II) The IAO will strictly control and limit access to TSOAUTH resources. Authorization is restricted to authorized personnel; and justification for access is documented.*

The TSO time-out time limit is enforced through the use of the **JWT** parameter in **SMFPRM00**. Refer to *Section 2.1.2.10, SMF Data Collection,* for further information.

7.2.1 ACF2

Access to TSO is controlled by the **TSO** attribute of the logonid record. Use this mechanism to authorize each user requiring access to TSO.

All TSO attributes are taken from ACF2, versus the IBM-supplied default of **SYS1.UADS**. As users are granted the TSO privilege, specify the appropriate default logon procedure (**TSOPROC**) and IAC (Installation Account Code, if applicable) (**TSOACCT**) for each user.

Access to IACs may be controlled through the use of ACF2 resource rules of **TYPE(TAC)**.

Access to TSO logon procedures will be strictly controlled using ACF2 resource rules of **TYPE(TPR)**.

- *(ZTSOA015: CAT II) The IAO will ensure that access to TSO logon procedures is controlled through the use of explicit ACF2 resource rules TYPE(TPR).*

The **PARMLIB** command is an authorized TSO command processor that provides users the ability to display and dynamically change, without an IPL, the active **IKJTSO*xx*** member of **SYS1.PARMLIB**. Control authority to execute this command by permitting user access to the ACF2 **PARMLIB** resource of the ACF2 **TSOAUTH** resource class.

Only systems programmers responsible for supporting TSO/E should be authorized to access the **PARMLIB** command.

The following steps provide examples of ACF2 commands necessary to control the **PARMLIB** command:

(1) Create a three-character resource type to be associated with the **TSOAUTH** resource class. The STIG required ACF2 resource type **TSO** should be used. Use the following parameters when creating a **CLASMAP** record:

```
RESOURCE(TSOAUTH) RSRCTYPE(TSO)
```

(2) Define the **PARMLIB** resource to ACF2 and protect user access to the **PARMLIB** command by using the following ACF2 statements:

(a) Permit a user access to the **PARMLIB** command to display specifications in the active **IKJTSO*xx*** member:

```
$KEY(PARMLIB) TYPE(TSO)
UID(xxxxxxxx) SERVICE(READ) ALLOW
```

(b) Permit a user access to the **PARMLIB** command to display and dynamically change the active **IKJTSO*xx*** member:

```
$KEY(PARMLIB) TYPE(TSO)
UID(xxxxxxxx) SERVICE(READ,UPDATE) ALLOW
```

ACF2 provides for assigning access to TSO/E commands through TSO privileges given to logonids. Controls will be in place to ensure that ACCTPRIV, CONSOLE, MOUNT, and OPERATOR privileges are restricted.

- *(ZTSOA040: CAT II) The IAO will ensure that special privilege MOUNT is assigned only on an as needed basis for LOGONIDS associated with STCs and LOGONIDS that need to execute TSO in batch.*

- *(ZTSOA040: CAT II) The IAO will strictly control and limit the TSO privileges given to logonids. The privileges will be restricted to authorized personnel and justification for the privilege is documented.*

7.2.2 RACF

Attributes for most TSO-level controls can be found in the RACF TSO segment of a user's profile. Use the **ACCTNUM** field to specify the IAC used for Fee-for-Service requirements, if applicable. The **PROC** parameter is used to specify the default TSO logon procedure associated with a user's TSO sign-on.

Validation of the user's IAC, if applicable, should be done against the **ACCTNUM** resource class. The user's ability to use a certain TSO logon procedure is validated against the **TSOPROC** resource class.

- *(ZTSO0010: CAT II) The IAO will ensure that access to TSO logon procedures is controlled and that access to multiple logon procedures are limited to authorized personnel.*

The **PARMLIB** command is an authorized TSO command processor that provides users the ability to display and dynamically change, without an IPL, the active **IKJTSO*xx*** member of **SYS1.PARMLIB**. Control authority to execute this command by permitting user access to the RACF **PARMLIB** resource of the RACF **TSOAUTH** resource class.

Only systems programmers responsible for supporting TSO/E should be authorized to access the **PARMLIB** command.

The following steps provide examples of RACF commands necessary to control the **PARMLIB** command:

(1) Activate the **TSOAUTH** resource class:

```
SETROPTS CLASSACT(TSOAUTH)
```

(2) Define the **PARMLIB** resource to the **TSOAUTH** resource class allowing no user access:

```
RDEFINE TSOAUTH PARMLIB UACC(NONE)
```

(3) Permit a user access to the **PARMLIB** command to display specifications in the active **IKJTSO*xx*** member:

```
PERMIT PARMLIB CLASS(TSOAUTH)  ID(user1) ACCESS(READ)
```

(4) Permit a user access to the **PARMLIB** command to display and dynamically change the active **IKJTSO*xx*** member:

```
PERMIT PARMLIB CLASS(TSOAUTH)  ID(user1) ACCESS(UPDATE)
```

7.2.3 TOP SECRET

The **TSO** facility controls access to TSO. Authorize each user type ACID requiring access to TSO with the appropriate **ADD** command.

```
TSS ADD(acid) FACILITY(TSO)
```

All TSO attributes are taken from TSS, versus the IBM-supplied default of **SYS1.UADS**. As users are granted the facility of TSO, specify a default logon procedure (**TSOLPROC**) and a default account code (**TSOLACCT**) for each user. The specifications for these fields ensure that only TSS is analyzed for TSO profile options.

Authorize access to account codes, if applicable, via the **TSOACCT** resource class with specific authorizations to use an account code being granted.

Access to a TSO procedure name is controlled via the **TSOPROC** resource class. Most users should only be capable of executing one standard logon procedure. Limit the ability to specify an alternate logon procedure to those users who have a justified need.

- *(ZTSO0010: CAT II) The IAO will ensure that access to TSO logon procedures is controlled and that access to multiple logon procedures are limited to authorized personnel.*

The **PARMLIB** command is an authorized TSO command processor that provides users the ability to display and dynamically change, without an IPL, the active **IKJTSO*xx*** member of **SYS1.PARMLIB**. Control authority to execute this command by permitting user access to the TOP SECRET **PARMLIB** entity of the TOP SECRET **TSOAUTH** resource class.

Only systems programmers responsible for supporting TSO/E should be authorized to access the **PARMLIB** command. The IAO is assigned ownership of the **PARMLIB** entity.

The following steps provide examples of TOP SECRET commands necessary to control the **PARMLIB** command:

(1) Assign ownership of the **PARMLIB** entity:

```
TSS ADD(acid) TSOAUTH(PARMLIB)
```

(2) Permit a user access to the **PARMLIB** command to display specifications in the active **IKJTSO*xx*** member:

```
TSS PERMIT(acid) TSOAUTH(PARMLIB) ACCESS(READ)
```

(3) Permit a user access to the **PARMLIB** command to display and dynamically change the active **IKJTSO*xx*** member:

```
TSS PERMIT(acid) TSOAUTH(PARMLIB) ACCESS(UPDATE)
```

This page is intentionally left blank.