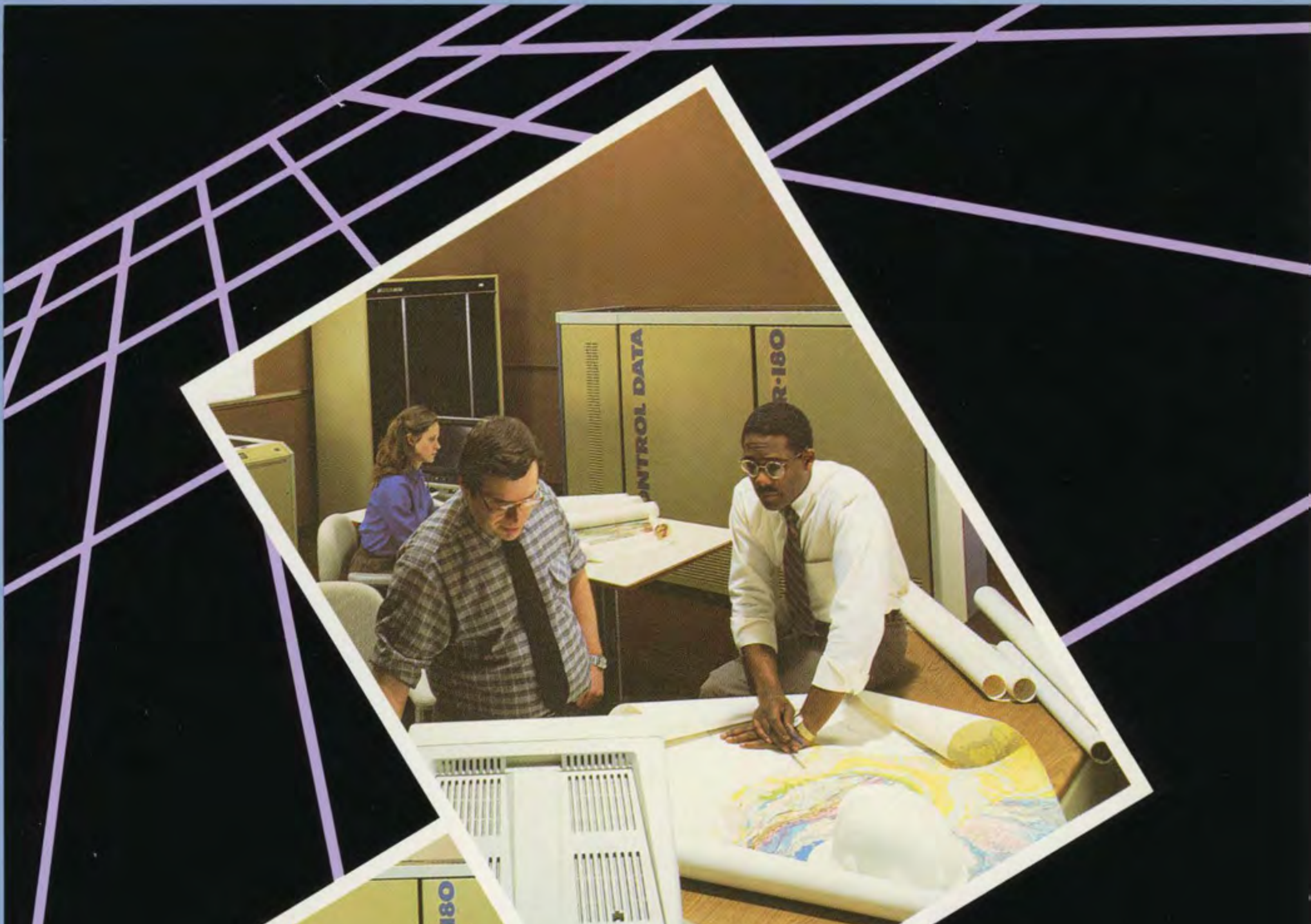
NOS Version 2
Security Administrator's Handbook

CONTROL DATA



60460410

NOS Version 2

Security Administrator's

Handbook

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Publication Number 60460410

Manual History

Revision	Description
A (12-31-84)	Manual released; reflects NOS 2.3 at PSR level 617. Features include all NOS multi-level security options.
B (9-23-87)	Revised to reflect NOS 2.5.3 at PSR level 688. This revision adds descriptions for Personal Identification, Password Masking, and Preventing Multiple Concurrent Logins to CDC Security Solutions in section 1. Section 2 includes a Security Needs Assessment Questionnaire, and discussions on Defining Site Security Requirements and Physical Security Requirements. Section 3 now includes recommendations for System Maintenance and recommendations for User Support. The following permission identifiers have been added to the AW parameter of the MODVAL directive: CNRD, COPR, CLTD, COPI, and CACA. An installation procedure for computer systems that have not run NOS has been added to section 4. Appendix F has been added to describe the Security Audit Reduction Tool. Due to extensive changes, change bars are not used, and all pages reflect the latest revision level. This edition obsoletes all previous editions.

Contents

Figures

Tables

About This Manual

This manual describes security features, both standard features for unsecured systems and multi-level security (MLS) features for secured systems, available under the Network Operating System (NOS). A secured NOS system enforces strict mandatory security controls in addition to the security controls present in unsecured NOS systems. A secured NOS system operates on the following CDC systems.

CDC® CYBER 180 Computer Systems

Models 810, 830, 835, 840, 845, 850, 855, 860, 870, 990, and 995

CDC CYBER 170 Computer Systems

Models 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 815, 825, 835, 845, 855, 865, and 875

CDC CYBER 70 Computer Systems

Models 71, 72, 73, and 74

CDC 6000 Computer Systems

Audience and Organization

This manual is a guidebook for the computer site personnel responsible for management and administration of site data security. This manual is organized in the following manner.

Section 1 Describes Control Data's security solutions—the hardware and software features that make our system an all-purpose, trusted computing facility.

Section 2 Describes the responsibilities of the site security administrator: assessing site security needs, defining site security requirements, and establishing site security practices.

Section 3 Contains recommendations for the operation and maintenance of a secured site.

Section 4 Describes the installation options for setting up a secured site.

Appendix A The glossary contains definitions of all terms associated with NOS 2 security features.

Appendix B Lists the user validations necessary for a secured site.

Appendix C Lists the operator commands and utilities that require security administrator privileges.

Appendix D Lists the maintenance commands and utilities that contain parameters for dealing with ranges of file access levels on a secured system.

Appendix E Details the user commands and macros available on a secured system.

Appendix F Describes a security audit reduction tool used to assist the security administrator in auditing security-related user activities.

Submitting Comments

There is a comment sheet at the back of this manual. You can use it to give us your opinion of the manual's usability, to suggest specific improvements, and to report errors. If the comment sheet has already been used, mail your comments to:

Control Data Corporation
Technology and Publications Division ARH219
4201 North Lexington Avenue
St. Paul, Minnesota 55126-6198

Please indicate whether you would like a response.

If you have access to SOLVER, the Control Data online facility for reporting problems, you can use it to submit comments about the manual. When entering your comments, use NS2 as the product identifier. Include the name and publication number of the manual.

If you have questions about the packaging and/or distribution of a printed manual, write to:

Control Data Corporation
Literature and Distribution Services
308 North Dale Street
St. Paul, Minnesota 55103

Or call (612) 292-2101. If you are a Control Data employee, call (612) 292-2100.

CYBER Software Support Hotline

Control Data's CYBER Software Support maintains a hotline to assist you if you have trouble using our products. If you need help not provided in the documentation, or find the product does not perform as described, call one of the following numbers. A support analyst will work with you.

From the USA and Canada:	(800) 345-9903
From other countries:	(612) 851-4131

Related Publications

The following manuals describe in greater detail certain topics covered in this manual.

Control Data Publication	Publication Number
CYBER Initialization Package User's Handbook	60457180
Network Access Method Version 1 Network Definition Language Reference Manual	60480000
NOS Version 2 Administration Handbook	60459840
NOS Version 2 Analysis Handbook	60459300
NOS Version 2 Installation Handbook	60459320
NOS Version 2 Operations Handbook	60459310
NOS Version 2 Reference Set, Volume 3 System Commands	60459680
NOS Version 2 Reference Set, Volume 4 Program Interface	60459690
NOS Version 2 System Programmer's Instant	60459370

Control Data manuals are available through Control Data Sales offices or Control Data Literature and Distribution Services (308 North Dale Street, Saint Paul, Minnesota 55103).

Disclaimer

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed functions or undefined parameters.

CDC Security Solutions 1

CDC Security Solutions 1

Control Data has long been involved with developing solutions to the ever-growing problems of computer security. The Network Operating System (NOS) and the CYBER 170 and 180 series computers represent a culmination of our design expertise and our experience with security-sensitive data processing sites. Our system—both the hardware and the software—incorporates the fundamental security controls necessary for a trusted computing facility. These security controls can be tailored to protect your users' data in a wide variety of time-sharing environments.

The CYBER hardware physically separates most system and user processing. NOS takes full advantage of the separation of functions inherent in the CYBER architecture. In addition, NOS offers a wide range of discretionary and mandatory access controls that allow you to set up a computer security system that is exactly suited to your site's needs.

For example, when NOS is installed as released, it automatically controls access to host computers and protects the privacy of both user data and system resources. If your site handles information which can or must be marked according to its security sensitivity and you need to enforce mandatory access controls for this information, you can install NOS to run in secured mode. When NOS runs in secured mode, it enforces strict multi-level mandatory security controls on all system activities. These multi-level access controls are designed to meet security requirements typically associated with the handling of classified information, but can also be used in a wide range of environments.

This section describes Control Data's security solutions in more detail: system integrity, software controls, NOS standard security features, and NOS multi-level security.

System Integrity

Control Data CYBER 170 and 180 hardware incorporates numerous features to detect and correct errors that ensure the integrity of data and the accuracy of results. These features are known as Reliability, Availability, and Maintainability (RAM).

The unique architecture of the CYBER 170 and 180 computers physically separates the central processor unit (CPU) and central memory from the peripheral processors (PPs) and data channels.

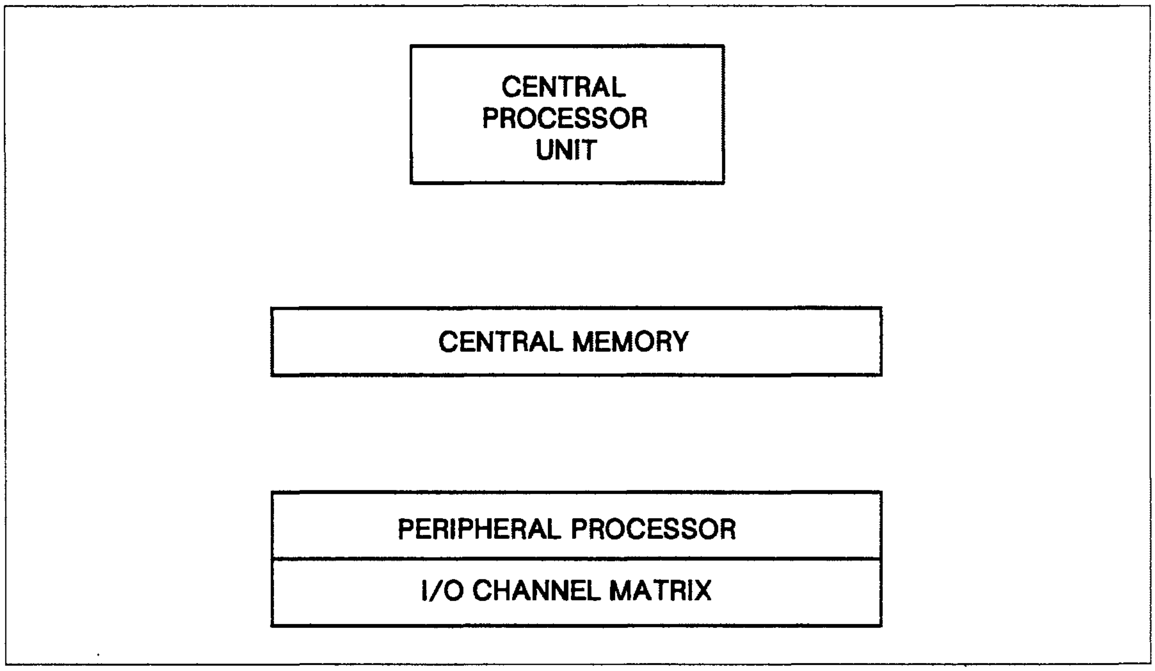


Figure 1-1. CYBER 170 and 180 Architecture

The CPU is designed primarily to provide high-speed computation and manipulation of data in central memory. It has no I/O capabilities and must request this and other services from PPs via requests stored in central memory. PPs, which have limited computational capabilities, perform a wide variety of I/O and system services.

The CYBER supports up to 22 independent processors—up to 2 CPUs and up to 20 PPs. Only PPs can perform I/O, and they have the power to deny I/O requests and to abort jobs. Also, the PPs execute only system software modules. The primary system monitor, called MTR; and the console operator interface, called DSD, execute in dedicated PPs which cannot be interrupted or overloaded by user actions. Although the EXEC modules in central memory participate in the process of managing the system, MTR is the primary controller of the system. This means that in the CYBER machines, the CPU is a high-speed slave to the physically isolated and dedicated PPs.

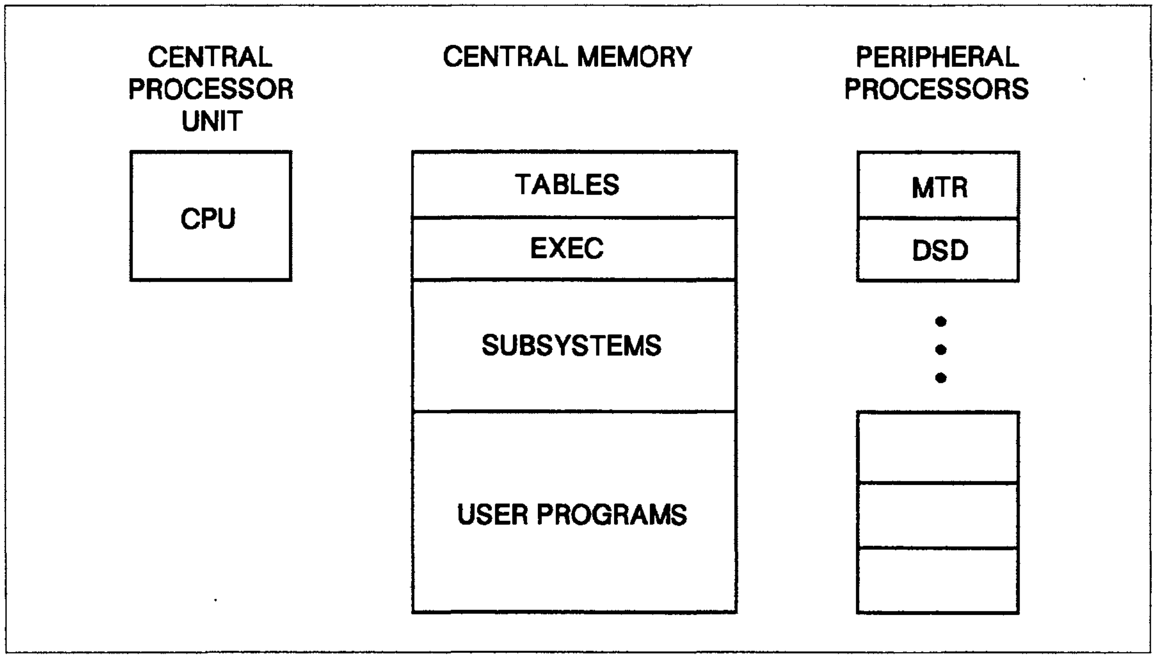


Figure 1-2. System Layout

Because the system software consists primarily of PP modules and PPs cannot execute user programs, the hardware design physically prevents users from subverting system software. User programs, which execute in central memory only, are separated and protected from one another by the CYBER memory protection hardware and the NOS system design, which does not permit the sharing of memory between independent processes.

The memory protection mechanism uses hardware registers to store the boundaries of central memory and extended memory assigned to each user job and does not permit a job to access memory outside of the established boundaries. Because these registers are inaccessible to CPU programs and are managed entirely by system modules, CPU programs are completely isolated from one another.

NOS builds on the CYBER hardware-based job isolation to ensure that jobs cannot interact except via system-mediated mechanisms. In fact, unless a user job specifically requests communication with another cooperating job (or common resource subsystem), there is no direct inter-job communication.

Software Controls

In addition to supporting the hardware mechanisms that separate system and user programs, NOS system design also provides security features that meet the requirements of these fundamental security principles:

- Identification and Accountability
- Access Controls
- Auditability and Surveillance
- Security Countermeasures

Identification and Accountability

Every action in a NOS system is traceable and attributable to an individual who is accountable for that action. Before being granted access to the computer, each user must enter a user name and a password. The user name uniquely identifies the user to the system, and that user name is associated with all processes initiated by, or on behalf of, that user. The user password is protected by an internal one-way data encryption mechanism.

Access Controls

Each user on the system is associated with a comprehensive set of authorizations. These authorizations control user access to system capabilities, resources, and privileges. When properly set, these authorizations prevent users from creating denial-of-service or deadlock situations.

Further access controls are provided by the NOS Permanent File (PF) system. The NOS PF system allows users to control access to their files by using discretionary file access controls. For example, a user can grant another user permission to access a file by explicitly identifying the user by user name. When a file owner grants another user permission to access a file, the owner can also specify the mode of access, such as read or write mode. The file owner can further protect files by assigning file passwords.

If your site requires additional access controls, you can enable the NOS multi-level security (MLS) feature during installation. When MLS is enabled, NOS imposes strict access controls on users, information, and devices. The access controls are based on hierarchical access levels and discrete access categories. To gain access to data or a process, the user must be authorized for the security level of the object. The user must also be assigned the same or a superset of the object's security categories.

Auditability and Surveillance

Each access, resource, or privilege control is based on user authorizations. Thus, when a control decision is made, it is recorded along with the identification of the responsible user. NOS automatically collects and records an audit trail of all control decisions, along with resource usage, system statistics, and other security-related data. This audit trail and other system logs are protected by the system and may be audited on-line via the system console or off-line via text editors or programs. The SECART utility, described in appendix F, can be used to help in the audit process.

Security Countermeasures

There are a number of automatic countermeasures embedded in user access controls. For example, the system keeps track of the number of times a user attempts to perform an action that would violate system security. Each attempt decrements the user's security count. When the user's security count is zero, the user is denied access to the system. In addition, the system aborts any job that attempts to perform an action that would result in a security conflict.

As a further countermeasure, NOS limits the number of invalid login attempts per terminal and the frequency of system-wide illegal login attempts to prevent brute-force penetration methods. The site security administrator can intervene with an executing batch job or user terminal session and can take remedial action by performing on-line modifications to user identifications, passwords, and authorizations, if needed.

NOS Standard Security Features

The standard NOS system offers the following security features:

- Control of Access to Host Computer
- Privacy of User Data
- Protection of System Resources

Control of Access to Host Computer

NOS controls access to the host computer by means of a user name and password validation system. Each user must enter a valid user name and password before accessing the host computer. The user name uniquely identifies each user to the system; the password authenticates the user identification. Thus, the user password is the key to user authorizations and must be protected to ensure user privacy and accountability. NOS protects user passwords using several mechanisms:

- Password Encryption

  All user passwords are stored on the system validation file VALIDUS in encrypted form. Whenever a user's password is examined by NOS, it is processed by a one-way encryption algorithm and compared with the encrypted form on the validation file. Thus, the system does not store users' passwords in plain text anywhere in the computer.

- Secure Login Feature

  You can enable the secure login feature to protect user login information from masquerade programs. (Masquerade programs execute at a terminal and imitate the system login dialogue. In this manner, the masquerade program intercepts the user's login information.) When this feature is enabled, a terminal user can enter a secure login character sequence to ensure a connection to the network login program and thus terminate any potential masquerade program.

- Personal Identification

  During installation, you can assign certain user names a personal identifier. These users must enter their personal identifier in addition to their family name, user name, and password in order to access the system.

- Separate Batch and Interactive Passwords

  During installation, you can select an option to provide separate user login passwords for batch and interactive jobs. The batch password is used for all batch jobs, including jobs initiated with the Remote Host Facility (RHF); the interactive password is used only during interactive login.

- User Password Expiration Dates

  You can enter a password expiration date for a given user's batch password, interactive password, or both. Thus, certain users (for example, students of a computer class) can be validated to use computer resources for only a specific period of time. Users may also be granted permission to set their own password expiration dates. The system automatically issues a password expiration warning message to the user dayfile of all interactive and batch jobs. This message is also displayed on the screen of interactive users at login time.

- Changing Passwords

  If you grant a user permission to do so, that user may change his or her user password. The user may change batch, interactive, or both passwords at any time.

- Password Masking

  To ensure the security of a user password during interactive login, all fields in the line containing the password are masked. For example, on a single line login the mask overwrites the family name, user name, password, personal identifier, and application. If each value is entered following its respective prompt, only the password line is overwritten.

- Preventing Multiple Concurrent Logins

  You can prevent certain user names from logging into more than one terminal at a time. If a restricted user is already logged in and attempts to log into another terminal, the system aborts the second login attempt and issues an error message.

- Recording and Processing Invalid Login Attempts

  NOS records a message in the account dayfile each time an invalid login sequence is entered from a terminal. The account dayfile message includes the terminal name associated with the attempt. By auditing the account dayfile, you can detect patterns of invalid login attempts from particular communication lines. These repeated invalid logins may indicate that an unauthorized user is attempting to access the computer. You can set the number of attempts permitted before the login process is automatically terminated and the terminal is disconnected. This feature increases the difficulty of gaining access to the computer by entering random login sequences.

- Security Count

  Each user is validated with a security count that defines the number of security violations the user can attempt before being denied further access to the system. For example, when a user attempts to submit or route a job with an invalid USER command, the system aborts the job and decrements the user's security count. When the user's security count is zero, the user is denied access to the system.

Privacy of User Data

NOS protects the privacy of user data in three ways.

1. **Memory Clearing**

2. **File Overwrite Option**

3. Discretionary File Access Controls

Memory Clearing

NOS automatically clears central memory and extended memory by filling them with a continuous pattern of zeros before assigning them to a user job. This feature prevents a user from accessing data left in memory by another user.

NOS provides an additional memory clearing option which can be enabled during installation. When this feature is enabled, the system clears all central memory and extended memory associated with a job whenever it is released from the job. This means that whenever a job rolls out, has its field length reduced, or completes, no data from that job is left in memory. This reduces the risk of memory data scavenging by maintenance personnel in the event of a system failure.

File Overwrite Option

NOS maintains control over all I/O operations and can thus ensure that a user cannot read mass storage space unless previously written as part of the file. NOS also provides a user-callable utility to overwrite the mass storage space occupied by a file before it is released. This reduces the risk of off-line access to the mass storage media.

Discretionary File Access Controls

With NOS, users can specify who can access their files and how those files can be accessed. These discretionary file access controls are accomplished via file catalog types, file permits and modes, and file passwords.

- File Catalog Types

  All NOS permanent files are assigned one of three catalog types: private, semiprivate, or public. A private file (system default) can be accessed only by the file owner or by users explicitly granted permission to access the file by the file owner. Each user may be granted different file access modes.

  Public and semiprivate files may be accessed by all other users in the owner's family if they know the owner's user name, the file name, and (if specified) the file password. Semiprivate files may have permits (as with private files), and permit records are created for each user who accesses the file. Permits cannot be created for public files.

- File Permits and Modes

  For private and semiprivate files, the owner can permit other users to access the file and specify the mode of access for the file (for example, read access or write access). In addition, the owner can specify an expiration date for the file access permission.

  For semiprivate and public files, there is a single access mode specified which applies to all users without an explicit file permit.

- File Passwords

  A user can further control access to a file by specifying a password for the file and an expiration date for the file password. This password must be specified by all other users who attempt to access the file.

Protection of System Resources

Under NOS, system resources are controlled by the system executive program CPUMTR, which resides in central memory, and by other system programs that execute either in central memory or in peripheral processors (PPs). NOS protects these operating system programs from potential user abuse through a combination of hardware and software mechanisms.

Hardware registers in CDC computers establish the boundaries of central memory and extended memory assigned to each user job; these values are part of the exchange package of a job. During each memory read or write instruction, the address control hardware checks whether the address to be read or written is within the boundaries of the job's field length. If it isn't, the hardware causes an interrupt and aborts the instruction. When this interrupt occurs, control of the processor is given to the CPUMTR program, which initiates appropriate action to issue an error message to the job, set error flags, and abort the job step or the job. Since NOS does not share central memory between users and the system or between independent processes, this mechanism protects other users' data as well as those portions of the operating system that reside in central memory.

The operating system programs that execute in PPs are further isolated from users by the physically separate memories. Users can interact with these programs only under the control of the operating system, using the NOS command interface, the NOS program interface, or a request entered in address RA+1 of the user job's field length. Users have no way of reading, writing, or otherwise altering PP programs in PP memories.

PP programs and overlays are loaded by a PP-resident program either from the system portion of central memory or from the system library on mass storage. Central Memory-resident PP programs are protected by the hardware memory protection mechanisms described above and the system library is protected via the system control of all I/O (by PP programs) and the NOS file system.

The potential for denial of service by abuse of system resources such as CPU execution time and file space is controlled by PP programs which monitor and control access to resources and have the capability not only to ensure appropriate sharing, but also to terminate user jobs which exceed resource authorizations.

NOS Multi-Level Security

If your site requires additional security features, you can enable the NOS multi-level security (MLS) feature during system installation. When MLS is enabled, the system is referred to as a secured system.

A secured NOS system enforces strict mandatory security controls over all system activity based on sensitivity marking for information and clearances for users and devices. NOS provides two marking schemes:

- 8 hierarchical access levels
- 32 non-hierarchical access categories

NOS controls access to and the flow of information in the system based on the system access levels and categories and user authorizations for those levels and categories.

Access Levels

During installation, you can select your own names for the eight hierarchial access levels. The released system has default names of LVL0 through LVL7. The lowest access level is associated with the least sensitive information; each higher level is associated with more sensitive information. Your site can establish its own policy regarding the degree of sensitivity associated with each level.

At deadstart time, you can define a range of access levels for the entire system by specifying lower and upper access level limits. The system then allows users to run jobs only within this range. You can also define a range of access levels for each job in a secured system. The system allows users to run jobs only within the access level limits for their job origin type.

You can assign an upper level limit for each communication line connected to the computer. When you assign limits for communication lines, these limits restrict the user's clearance to the limit of the communications line. This feature makes it possible for sites to allow access to particularly sensitive information only in areas where physical security permits that information to be handled.

In addition to setting up access level ranges for the system, job origin types, and communication lines, you must also assign a range of access levels for each physical device attached to the system. Only files whose access levels are within this range may be processed by the device.

Finally, you must validate each user to use a set of access levels. Normally this is a contiguous range of access levels, such as LVL0 through LVL4. A user can never access a file on a secured system outside of this set of access levels.

Job Access Level

Every job in a secured system has a set of access levels and a current access level associated with it. NOS bases the job access level set on the intersection of the user's validated set of access levels, the current system access levels, the job orgin type access levels, and, if applicable, the communication line access levels.

The job's initial access level is the lowest level in its access level set. The job access level is automatically raised when the user job reads a file that has a higher access level than the job's current access level, provided the file is within the job's access level set. A user job can never access a file whose access level is not within the job's access level set. The user job can also raise the job access level (within the job access level set) using a command or macro.

Only users who have special validation may lower the access level of their job. This means that a user's job access level will typically rise during the job's execution, and then remain at a high level.

A user can select an upper access level limit for a batch job on the Job command. This limit must not exceed the limits the system imposes on the job.

File Access Level

Each local, permanent, and queued file in a secured system has an associated access level. This access level is an indication of the sensitivity of the information contained in the file and is assigned by the owner when the file is created. Unless the file owner has special permission, the access level of a file can be raised but never lowered; this is to prevent the unauthorized downgrading of information. A file's access level can be changed only if the new access level is valid for the device on which the file resides.

In order to write to a file, its access level must be at or above the job access level. If a write is to be permitted, a lower level file will be automatically raised to the current job access level. The access level of attached direct access files and magnetic tape files can never be changed.

Although access levels can be assigned to files on unsecured systems, they are not used to control file access. This means that files transferred from a secured system to an unsecured system lose this protection.

Access Categories

An access category is a classification marking assigned to a permanent file by the file owner to restrict access of the file to a particular group of users. NOS supports up to 32 access categories that have the default names of CAT00 through CAT31. At installation, your site can assign names to the categories and establish its own policy regarding the meaning associated with each category.

Unlike access levels, access categories are not hierarchical; no category is intrinsically higher or lower than another. Also, a permanent file may have one, several, or no access categories, while any file must always have exactly one access level.

At deadstart time, you can specify a set of access categories for the entire system. The set of system access categories can consist of some, all, or none of the defined access categories. The system then allows users to access only files in this set.

You can validate each user to use a set of access categories. This set can consist of some, all, or none of the defined categories. A user can never access a file on a secured system without being validated for all of the access categories of the file.

Job Access Categories

Every job in a secured system has a set of access categories associated with it. NOS bases the job access category set on the intersection of the user's validated set of access categories and the set of system access categories.

When a permanent file is created, it is automatically assigned all of the access categories in the creating job's access category set. The permanent file owner can also explicitly assign any, all, or none of those categories to the file using a command or macro.

Flow of Information on the System

A job may have simultaneous access to files of different access levels. However, a secured system restricts the flow of information between files of different access levels:

- A job may not read a file whose access level is greater than that of the job. This restriction prevents a user who is not authorized for a certain access level from reading information at that level. For example, a job running at LVL2 could read a file at LVL0, but could not read a file at LVL4. The normal case is that the job's access level is automatically raised to the level of the file, but if the file's access level is greater than the highest access level permitted for the job, the read operation is not allowed.

- A job may not write to a file whose access level is lower than that of the job. This restriction prevents the downgrading of sensitive information. The system attempts to raise the access level of the file under these circumstances, but if this is not possible (for example, if the file is a magnetic tape file or an attached direct access file), the write operation is not allowed. (A user who has a special validation may write to files having a lower access level than the job access level.)

Special Handling of Printed Output

At a secured system site, you can restrict the handling of printed output based on the access level of the output. For example, your site can require that all print files at or above a certain access level remain queued until individually released by the operator for printing. This feature, called output queue special handling, can be enabled during installation or at the system console while the system is running.

Security Conflicts

When a user attempts to perform an operation that could violate system security, the system issues a SECURITY CONFLICT message to the user dayfile and aborts the job step. A security conflict causes the requesting user's security count to be decremented, as in an unsecured system. The following actions create security conflicts in a secured system:

- An attempt to set an invalid access level on a file or job.
- An attempt to access a permanent file from a job not validated for the access level and/or access category set for the file (this is not a security conflict if an alternate user attempts to access the file).
- An attempt to write data to a magnetic tape file or an attached direct access file when the job access level is higher than the access level of the file.
- An attempt to set an access level for a file or job which, although within the user's validation, is lower than the current access level of the file or job.

Limitations of the Secured System

Although a secured system imposes a strict system of mandatory security controls, there are some limitations on the applicability of these controls. Your site must provide other means of protection for the following items.

Restrictions on the Use of Magnetic Tape Files

A secured system requires that the access levels of magnetic tape files be within the user's job access level limits and within the access level limits for the device on which the tape is mounted. The user can assign an access level to a tape file using the ASSIGN, LABEL, or REQUEST command or macro, or allow the system to assign a default access level. The access level of a tape file cannot be changed once the file is assigned to the job.

However, a secured system cannot enforce access restrictions to the tape file if the tape is transported to another system. The ANSI labels used by NOS have no standard fields for security; therefore, NOS has no way to control subsequent accesses to the tape. Sites must therefore ensure that other measures are provided for the security of tapes. Here are two methods you can use:

- Use only ANSI-labeled tapes.
- Use the NOS Tape Management Facility (TMS)

Restrictions on the Use of Products

NOS secured system mode does not support security markings and mandatory access controls within the following Control Data products:

- Remote Host Facility (RHF)
- Transaction Facility (TAF)
- CYBER Data Base Control System (CDCS)
- RECLAIM

A secured system can operate in a shared mass storage environment provided each mainframe accessing the shared mass storage of a secured mainframe is also a secured mainframe. Each system may use different system and job access level limits.

Security Administrator Responsibilities 2

Security Administrator Responsibilities 2

A site security administrator is responsible for the overall security of a computer site and must perform a variety of tasks. For example, the security administration of a computer system includes these processes:

- Assessing Site Security Needs
- Defining Site Security Requirements
- Establishing Site Security Practices and Procedures

This section briefly describes the first two processes in the above list; section 3 describes recommendations for site security practices and procedures.

Assessing Site Security Needs

Before you can define security requirements that are appropriate for your site, you must assess your site's security needs. That is, you must make a detailed list of all the probable threats to security that exist for your site. Each site will have different security needs depending on the physical environment of the computer system, the size of the organization, the sensitivity of the data, and the type of work performed at the site.

There are many resources available to help you determine site security needs. For example, you may want to refer to one of the many reference books on the topics of computer crime and computer security; you could seek advice from those responsible for security at data processing sites that are similar to yours; or your site could seek professional help from a certified security consultant.

The following pages contain a security needs assessment questionnaire to be used as a starting point for determining your site's security needs.

Security Needs Assessment Questionnaire

1. Do computers provide services which are critical to the operations of your organization, and would the loss of this information impact the ability of the organization to meet objectives?

2. Does the information you maintain on your computers represent a significant asset of your organization?

   a. Have you assessed the financial and competitive impact on your organization in the event of the loss or destruction of such information?

   b. Have you assessed the financial, competitive, and political ramifications of disclosure of such information?

3. Can the identity of individuals be determined from the information you maintain on computers?

   a. Is your use of such information regulated by local, state, or federal laws, regulations, or guidelines?

   b. Have you assessed the legal, financial, and political ramifications of disclosure of such information?

   c. To what extent do you have a fiduciary responsibility to ensure the accuracy and currency of such data, and have you assessed the consequences of unauthorized modifications to such data?

4. Does your organization use computers in support of work performed on behalf of a government department or agency?

   a. Is any of the processed information subject to government laws or regulations concerning the handling of such information?

   b. Has the government placed specific security requirements on systems which handle such data?

   c. Is your computer system subject to government security evaluation and certification requirements? If so, which agency will evaluate your system and what evaluation criteria will be used?

5. Do you need to establish controls that limit access to your major computing facilities to authorized persons only?

   a. Do you need to establish physical access controls?

   b. Do you need to establish telecommunications access controls?

   c. Do you permit access via switched telephone or other commercial communications networks?

   d. Have you established procedures for data backup and recovery?

   e. Has your organization defined who shall be held responsible for inadequacies of access controls and the consequences of a breach of those controls?

6. Have you assessed the potential for an authorized user to abuse your computing facilities, systems, and the information stored therein?

    a. Are all authorized computer users employees under contractual controls?

    b. Do authorized users have access to all of the services and information provided by your computers?

    c. Can your user community be grouped by their information assets and/or service and access requirements?

    d. Are you aware of current or previously authorized users who might benefit from use of your computing facilities, access to information, or modification of data?

    e. Are you aware of current or previously authorized users who may be motivated to abuse your computing services and/or information assets to the detriment of the organization?

7. Are you aware of any outside individuals or organizations who might benefit from the use of your computing facilities, access to information, or modification of your data?

8. Are you aware of any outside individuals or organizations who might be motivated to abuse your computing facilities, systems or information assets to the detriment of the organization?

Defining Site Security Requirements

Once you have assessed your site's security needs, you can begin the process of developing requirements to meet these needs. During this process you must determine what types of security controls are required to address each need, and then determine if your site meets that requirement. Typically, some requirements may not be adequately met, if at all. In some instances, the threat probability may be very low and/or the consequences not significant enough to warrant fully satisfying the requirements to meet the threat. A management decision is required to accept the risk of not fully meeting the requirement. After you complete the list of requirements, you should order the items in the list according to their importance to your organization and address the requirements accordingly.

The remainder of this section provides a general outline of the types of security controls required at most sites and specific examples of NOS controls designed to meet those controls which a computer system can be expected to provide.

There are three broad types of security requirements for computer installations:

- Physical Security Requirements
- Personnel Security Requirements
- Computer System Security Requirements

Physical Security Requirements

The physical security policies implemented at your site must include preventive measures for each physical threat to your system. To do so, these policies must cover the following issues:

- Physical location of your facility

  Providing a proper physical location for your computer facility can eliminate many security threats due to the physical environment. Your computer facility should be located such that it is the least susceptible to any physical threats. For example, select an area that is free from congestion (both pedestrian and traffic), and that protects your facility from the natural disasters that are possible in your geographic area. In addition, you should locate your facility either in a separate building or in an area of a building that has been specifically modified to protect the system.

- Physical access control

  Your site policies should dictate who is allowed physical access to the computer system and all sensitive materials at your facility. Your site should establish methods for enforcing controlled admittance policies. These policies should include card-activated locking devices for computer rooms, employee identification badges, and physical access logs. The distribution of keys, badges, and materials should be recorded, and when a person is no longer employed at the facility, all security-related items should be retrieved.

- Natural disaster protection

  In addition to protecting your site from all damages relating to fire, water, and storm damage, your site's security policies should contain measures to protect your facility from every natural disaster that is likely to occur in your geographic area.

- Power protection

  Your site must maintain the reliability of the power source for the computer facility. That is, you must protect your system from the occurrence of transient voltage (a positive or negative voltage peak), blackouts (a complete failure of power), and brownouts (periods of low voltage). To do this, your site should consider using voltage-regulating transformers (VRT), uninterruptible power supplies (UPS), or power generators.

- Emanations

  Electromagnetic and acoustic emanations from hardware devices can be intercepted and interpreted by unauthorized persons who have the technical expertise and hardware to do so. If your site processes sensitive or classified information, your site must consider the threat of emanations interception, and should prevent it by physically separating radiating devices from potential intercept points. In addition, close attention should be given to the location of remote terminals to ensure that they are also located in a secure environment.

- Interference

  Your site may need to protect the computer system from radar, microwave, and magnetic pulse interference depending on the site location. In addition, ensure that the separate components of your computer system and other machines do not cause interference during equipment operation.

- Storage and disposal

  Your site should secure the storage areas for all materials (documents and storage media) that contain sensitive information. In addition, you should establish practices for the safe disposal of these materials.

- Magnetic media management

  Your site should develop strict practices for the handling of all magnetic media to prevent loss of data and to reduce read/write errors. These practices should dictate the mounting and unmounting of disk packs and tapes, and the proper storage for all media when not in use.

Personnel Security Requirements

Each individual directly or indirectly involved in the operation of a secured computer system must be cleared according to the your site's procedures for access to sensitive information. These individuals include site security managers, site security administrators, systems programmers, systems analysts, and operators.

Within the context of a secured NOS system, the term security administrator refers to any person who has been granted security administrator privileges in their user validation record. It is your site's responsibility to determine how many and which persons require security administrator privileges in order to maintain system operation satisfactorily. Because security administrators must be cleared for access to the most sensitive information contained in the system, we recommend that the number of security administrators be kept as small as possible.

The following list of security personnel describes one way that a site could distribute site security responsibilities. Depending on the needs of your site, your site may assign one person to be responsible for more than one of the security tasks described in the list.

- Site Security Manager

  The site security manager is responsible for overall security of a data processing site. These responsibilities include computer systems security, personnel security, and physical security. The site security manager requires security administrator privileges in order to monitor and evaluate the security of a secured system, and must be cleared for access to the most sensitive information contained in the system.

- Site Security Administrator

  The site security administrator maintains the validation file for a family of users in a NOS system (this person is sometimes called the family administrator or the validation file manager). This individual is responsible for adding and removing users' access to the system and for granting and revoking users' privileges. In a secured system, the site security administrator establishes users' security validations and permissions by entering these permissions in a validation file. The site security administrator must have security administrator privileges in a secured system in order to run the MODVAL user validation utility. Therefore, site security administrators must be cleared for access to the most sensitive information contained in the system.

- Systems Programmers

  Systems programmers should be necessary only at those sites that make modifications to the standard operating system. Because operating system modifications can affect the integity of a secured NOS system, systems programmers must be cleared for access to the most sensitive information contained in the system. Because security administrator privileges are required to create a new version of the secured NOS system using the SYSEDIT utility, it is likely that some systems programmers will be given security administrator privileges.

- Systems Analysts

  The systems analyst is responsible for the overall performance of a system and for identifying and correcting software problems. The NOS systems analyst occasionally must perform tasks that require security administrator privileges in a secured system. These functions include setting ENGINEERING mode, setting DEBUG mode, viewing central memory, and creating memory dumps. Systems analysts require security administrator privileges and should be cleared for access to the most sensitive level of information in the system.

- Operators

  Because the console interface for secured NOS systems prevents the operator or other individuals from accessing sensitive information stored in the computer, operators need not be given security administrator privileges. However, many operator functions require that a high degree of trust be placed in the operator of a secured system. These functions include mounting tapes and removable disk packs, interrupting or terminating user jobs, performing recovery deadstarts, releasing queued files for output, and handling printed output.

  At some sites, one or more operators may be entrusted to function as a system security administrator reporting directly to the site security manager.

Computer System Security Requirements

Section 1 of this handbook describes the wide range of computer system security features provided by the CYBER hardware and the NOS operating system. Your site must establish practices and procedures that implement these features in a manner that is appropriate for your site. For example, your site can install NOS in unsecured or secured mode.

- Hardware Configuration Management

  For a secured system, you must assign access level limits for each hardware device connected to the system, including communication lines between devices. The policies your site establishes for assigning these limits should depend on the physical location of the devices and the level of security inherent in the physical device. Thus, at a secured NOS site, the hardware configuration of your system must include a mapping of the access limits to the hardware devices.

- Software Configuration Management

  Your site must establish policies regarding how the system is to be installed and the manner in which your site will allow modifications to the system after installation. For a secured site, you must set up policies that dictate how access level limits and categories are assigned for both individuals and information, how access level limits and categories can be modified, and when modifications can be made. In short, your site must establish policies for implementing all software security features. In addition, your policies should include a comprehensive set of practices and procedures for file system backup and recovery. Section 3 of this handbook gives recommendations for operation and maintenance practices; section 4 describes installation security options.

Guidelines for System Operation and Maintenance 3

Guidelines for System Operation and Maintenance 3

This section contains recommendations for site security practices and procedures concerning system operation and maintenance. These recommendations are intended to be used only as a starting point for developing a set of security practices and procedures that are appropriate for your site.

This section does not describe how to operate a NOS system, nor does it describe standard NOS maintenance procedures. The NOS 2 Version Operations Handbook and the Administration Handbook contain complete information about the commands and utilities used for system operation; the NOS Version 2 Analysis Handbook contains all the information necessary for system maintenance.

Recommendations for System Operation

General guidelines for system operation are:

Designate an operations security manager.

The operations security manager is responsible for enforcing site security practices that relate to system operations. Also, this individual is accountable for any operations security violations.

Secure the operations room.

Secure the operations room by implementing a card-activated locking system capable of recording each access to the room. In addition, you should require that users display an ID before being admitted to the room. Within the operations room, position the console operator area such that the operator is visible to management at all times, and establish separate tape and unit record storage areas.

Keep the system console in security lock status.

On an unsecured system, the console should be kept in LOCK status; on a secured system, the console should be kept in SECURITY LOCK status. You should put the console in security unlock status only when absolutely necessary.

Discourage job submissions via local batch card decks.

Discourage this method of job submission and encourage interactive and remote job entry. If local batch card decks are used, you should require the Job and Validation cards to be color-coded. Then, always destroy validation cards immediately after use.

Use the output queue file special handling (OQSH) feature to control output files that contain sensitive information.

Specify the access level for which output files should receive special handling. Set the access level during initial deadstart by placing the OQSH entry in the IPRDECK.

Establish a security practice for releasing output files.

This security practice should detail how the operator should release queued output files.

Hand deliver packs, tapes, cards, and printed output.

If it is not practical for your site to hand deliver printed output, use visible output bins without access, separate materials by type, and color-code the printouts.

Tape Management

Install and use the NOS Tape Management System (TMS)

TMS manages magnetic tape usage by establishing a database of tape owners; the TMS database keeps track of which user owns specific tapes and enforces the security options that tape owners can select.

Use ANSI-labeled tapes.

If possible, use only ANSI-labeled tapes.

Develop and enforce a site procedure for labeling tapes.

Require the operator to ensure that correct and unique volume serial numbers (VSNs) are used. If possible, require that the operator always set the OWNER parameter; also, require that the VA=x (volume accessibility) parameter is set to lock the VSN and OWNER. Refer to the BLANK command, found in the NOS Version 2 Reference Set, Volume 3, System Commands.

Require that the operator drop any request without a VSN.

This recommendation has three benefits: it frees the control point, it discourages the use of unlabeled tapes, and it encourages the use of labeled tapes and automatic assignment of tapes.

User Validations

Designate a Validations Security Manager.

The validations security manager is responsible for the integrity of the system validation files. This individual should make sure that there is a valid rationale for raising users' access level limits.

Install user validations at the system console during deadstart.

Subsequent modifications should be allowed only when authorized by the validations manager. When modifications are made, input the directives for the validation file manager utility via permanent files.

Assign user names to individuals.

To ensure that every action is traceable and attributable to a specific individual, you must assign user names only to individuals. In addition, develop a meaningful scheme for selecting user names.

Assign user indices carefully.

Because user indices (UIs) determine permanent file residency, and forcing UIs can allow sharing permanent files, you should assign user indices carefully. In addition, you should consider user index mapping for public packs shared between families.

Require all users to select a login password.

In addition to requiring that all users select a login password, you should also encourage users to change their passwords frequently. It is also recommended that all users maintain separate batch and interactive passwords. Discourage the use of single-line login sequence and encourage the use of step-by-step login sequence to allow password masking and echoplex suppression.

Restrict the system capabilities allowed for each user.

That is, restrict users to the minimum capabilities necessary for their job. In addition, assign low tolerance levels for the number of tapes to be used, the number of jobs to be submitted, the amount of CPU time, and the amount of mass storage. Also, set up the system so that it defaults to minimum access permissions.

Add locally created applications, such as payroll, student grades, etc., to the list of applications that require validation privilege.

Accounting

Designate an accounting security manager.

The accounting security manager is responsible for project accounting (tracking system resource use and maintaining the PROFILE file).

Use PROFILE project accounting facilities.

Use all three levels of these job accounting facilities: charge number, project number, and user name. In addition, PROFILE sets the limits for SRU usage, time in and time out, expiration for charge and project numbers, accumulated SRUs, and system resources.

Install accounting controls at the system console during deadstart.

Subsequent modifications should be allowed only when authorized by the accounting manager. When modifications are made, input the directives for the PROFILE utility via permanent files.

Have the accounting manager establish charge numbers.

For each charge number, the accounting manager should assign a master user (or specify the default master user as the accounting manager). As mentioned above, set SRU limits, expirations dates, and other available limits.

Have the master user establish project numbers.

For every project number, the master user should always define the authorized users, otherwise the project is open to all users. Again, set SRU limits, expiration dates, and use time controls when appropriate.

Require master user sign-off on usage reports.

It is suggested that the master user sign off all usage reports and keep all project accounting records secured.

Surveillance

Prohibit other uses of the system console.

During normal system operation, the system console should be used only by the console operator to perform operating procedures. For example, do not allow programming from the system console.

Develop an application that generates an exception report.

This application should be run as a routine part of your accounting procedures. The application should scan account and job dayfiles and list security-related messages. The program could also look for specific patterns in security conflicts. See the description of SECART in appendix F.

Recommendations for System Maintenance

General guidelines for system maintenance are:

Designate an individual responsible for system maintenance.

The individual responsible for system maintenance should be the only user who is permitted system origin privileges. This individual is responsible for all system modifications, creating the official (and only) deadstart tape, maintaining complete source code libraries, and controlling access to source libraries.

Keep the systems programming staff separate from the troubleshooting staff.

Permanent File Integrity

Separate functional groups of users by family.

Functionally separate groups should not have access to other groups' files. To prevent such access, organize the groups into separate families.

Manage device use.

For example, aggregate users by user index; assign users to logical devices using device and secondary masks; change user indices if necessary to level allocations; use private packs for very large or very private files; use public packs for files shared among families; use multi-spindle devices for greater capacity.

Reload system devices on a weekly basis using permanent file dumps.

Reloading system devices on a weekly basis gives you the opportunity to reorganize users and devices, and automatically cleans up file allocations.

Backup and Recovery

Perform daily incremental dumps.

Dump all files that have been modified since the last full dump. Dump the files on a device-by-device basis; use two sets of tapes for a two-week recovery cycle.

Perform weekly full dumps at the end of each week.

Dump the files on a device-by-device basis; implement a four-week short term backup cycle and a six-month long term backup cycle.

Manage quick recovery packs.

Remove the packs after each full dump and replace with an alternate set of packs. Reload the alternate set from the dump tapes.

Develop procedures for disaster recovery.

Maintain off-site hard storage for recovery files. Store the most recent week-old dump tapes plus the most recent end-of-week incremental dump tapes. This system provides two sets of up-to-date dumps and enables you to verify the usability of the dump tapes. To recover from device failure, swap packs and reload from the most recent incremental dump tapes.

Recommendations for User Support

General guidelines for user support are:

Your site should provide user support facilities.

Establish a hotline office for user assistance; establish a problem mailbox with utilities; provide online system bulletins; alert users by using MESSAGE and WARN commands.

Maintain a controlled, up-to-date library of manuals.

Maintain a LIBRARY catalog of user support files.

The LIBRARY catalog should include the following: systems bulletins and documentation (read mode); non-standard application programs (execute mode); special procedure libraries (read mode); special data files (read mode); and mailboxes (append mode).

Secured Programs

Manage program storage.

Store programs in permanent files in the owner's catalog. They should be stored as direct access, semiprivate or private files (with permits) in execute-only mode.

Provide user controls.

Programs should test for job origin type, and should verify whether the user name is available. Programs can also require additional data before allowing access. In addition, programs should provide logoff or termination capabilities, and should keep track of program usage.

Provide execution protection.

Secure memory using the SETSSM macro; provide error control with MODE, REPRIEVE, and EREXIT; disable terminal control using the DISTC macro; provide recovery processing.

Protect data files.

Store data files in a user catalog separate from the program, and make permanent file access internal to the program. In this manner, there is no user dayfile record, the passwords are protected by the program, and the user name of the files is concealed. In addition, use auto file flushing (GETLOF, SETLOF), and auto file release (PROTECT).

User Guidelines

Protect your password.

We recommend the following practices for password protection:

- If you must give out your password, change it immediately afterward.
- Change your password frequently.
- Choose passwords that are personal; personal passwords are more difficult for other users to guess. For example, using the first letters of words in an easily remembered phrase is an excellent method of selecting a password.
- Never write down your password.
- Use the /USER option for submitting batch jobs.
- If you must use cards, guard the user card carefully.
- Use the password randomization feature.

Protect your permanent files.

We recommend the following practices for permanent file protection.

- Avoid using the catalog type PUBLIC unless the file contains completely inconsequential information and you really don't care who accesses it.
- Use catalog type SEMIPRIVATE when practical.
- When you grant alternate user access with file permissions, always specify the least mode of access needed.
- For extra protection of your file, specify a file password.
- Do not make your files visible to other users by using the AC=Y option (Alternate Catlist feature).

Beware of programmers bearing gifts of helpful procedures and utilities.

The gift of a procedure or utility can be like that of the Trojan Horse. That is, the procedure or utility may contain more than you suspect. For example, the procedure could contain code to intercept and store your user login information, or it could contain code to access your private files.

Installing NOS in Secured Mode 4

Installing NOS in Secured Mode 4

To set up a system that runs NOS in secured mode, you must specify a variety of security options while you are installing NOS and its product set. This section gives the general procedures you should follow and describes the following software installation options for secured systems:

- CMRDECK Entries
- EQPDECK Entries
- IPRDECK Entries
- Network Configuration File Entries
- Secure Login Feature
- Access Level and Category Names

Because this section describes security installation options only, you should use this section as a supplement to the NOS Version 2 Installation Handbook. Also, during the installation process, you'll be performing a variety of tasks that may, depending on your knowledge of NOS, require that you refer to other NOS manuals:

- When you create, modify, or initialize validation files, you may need to refer to the NOS Version 2 Administration Handbook. This handbook contains complete information about NOS validation files and the system utilities you need to create and maintain the file (GENVAL and MODVAL).
- When you deadstart the system, you may need to refer to the NOS Version 2 Operations Handbook. You may also need to refer to the CYBER Initialization Package (CIP) User's Guide. These manuals contains complete information about deadstarting the system.
- When you create and modify deadstart decks or use the PFDUMP and PFLOAD utilities, you may need to refer to the NOS Version 2 Analysis Handbook. This handbook contains complete information about the deadstart decks and details the PFDUMP and PFLOAD utilities.

Installation Procedures

The NOS Version 2 Installation Handbook describes two methods for installing NOS: standard installation and customized installation. If you need to enable the secure login feature or define your own names for access levels and categories, you must perform a customized installation.

If you are installing a secured system on a computer that is currently running NOS, follow the procedures detailed in *Computer Systems Currently Running NOS.* If you are installing a secured system on a computer that has not previously run NOS, follow the procedures detailed in *Computer Systems That Have Not Run NOS.*

Computer Systems Currently Running NOS

To set up a secured system on a computer that is currently running NOS, perform these steps:

1. Follow the directions in the NOS Version 2 Installation Handbook for preparing for installation. After you have gathered together all the materials necessary for installation, begin the installation process.

    - If you do not want to enable the secure login feature or name access levels and categories, and you do not need to customize NOS or any products, go to the section that describes the standard installation process.

    - If you want to customize your system (that is, you want to enable the secure login feature, name access levels and categories, or change NOS or its products in any manner), go to the sections that describe the customized installation process and follow the directions for preparing for a customized installation.

2. Perform a level 0 deadstart of the system in unsecured mode using the released deadstart tape. (That is, do not include the CMRDECK entry that defines the system as a secured system.)

3. If you have a validation file that you want to use on the secured system, you must convert the file to match the PSR level of the released system. In addition, you may want to do the following:

    - If you want to transfer the entire contents of your validation file to the secured system, make a source version of the file using MODVAL and specify the OP=S option. Then, use the PFDUMP utility to make a backup copy of the file on tape. Here is an example of the type of job you should run to create a source version of the validation file:

      ```
      X.DIS.
      SUI,377777.
      DEFINE,SOURCE
      MODVAL,OP=S,FA.
      DROP.
      ```

    - If you want to use the SYSGEN utility to aid in the installation process, modify your validation file so that it contains the default user names and passwords for the products you are installing. (Rather than modify the validation file, you can modify the SYSGEN procedure so that it will use your user names and passwords. Refer to the NOS Version 2 Installation Handbook for details about using SYSGEN during installation and for a list of default user names and passwords).

4. Install NOS and its products. That is, modify the system software that is released on the deadstart tape, and then create a new deadstart tape of the modified system. Follow the steps in the NOS Version 2 Installation Handbook for installing NOS and its products. While you are installing your system, do the following:

   - When you create your deadstart decks, include the security entries necessary for your site. Refer to CMRDECK Entries, EQPDECK Entries, and IPRDECK Entries in this section.

   - When you install CCP, select security characters if you want to enable the secure login feature.[1]

   - When you create or modify your Network Configuration File (NCFFILE), include entries to define the access limits for the Network Processing Unit (NPU) communication lines.[2]

   - When you run the NOS installation procedure, define your own names for security levels and categories.

   After you have run all the procedures necessary to install NOS and its products, create a new deadstart tape of the modified system.

5. Perform a level 0 deadstart of the system in secured mode using the new deadstart tape. That is, specify a CMRDECK that contains an entry to define the system as a secured system.

6. Create and initialize a validation file using the source version of your validation file. To do so, run a job similar to the following:

   ```
   X.DIS.
   ASSIGN,pfdumptape,VSN=vsn,D=den,LB=KL.
   PFLOAD.
   GENVAL.
   ISF.
   USER,SYSTEMX,SYSTEMX.
   PURGE,VALIDUS,VALINDS.
   ATTACH,SOURCE.
   MODVAL,OP=C,I=SOURCE,N=VALIDUS.
   RETURN,VALIDUS,VALINDS.
   ```

   At this point, this must be the only job running in the system. Drop MAGNET and any other subsystem that may be running.

   ```
   ISF,R=VALIDUS.
   ISF.
   DROP.
   ```

   This job uses the PFLOAD utility to load the source version of your validation file. Then, GENVAL creates and initializes a default validation file. Next, the job creates a validation file from the source copy of the former validation file, and initializes the new validation file.

[1] CDCNET networks do not require special installation options to enable the secure login feature.

[2] It is not possible to define access limits for communication lines in a CDCNET network.

7. Assign security administrator privileges to some user other than SYSTEMX. To do this, run a job similar to the following:

```
X.DIS.
USER,SYSTEMX,SYSTEMX.
MODVAL,OP=Z./username,SAC=ALL,SAV=ALL.
DROP.
```

8. Assign new passwords and expiration dates to the default user names and delete the security administrator privilege from user name SYSTEMX. Because the default user names and passwords are public (that is, they appear in system and user documentation), they must be changed to site-selected names on a secured system.

9. Assign security privileges to users of the secured system. To do so, you must use the MODVAL utility to specify values for the SAC, SAL, and SAV parameters for user names. MODVAL input parameters are described in appendix B.

Computer Systems that Have Not Run NOS

To set up a secured system on a computer that has not run NOS, perform these steps:

1. Follow the directions in the NOS Version 2 Installation Handbook for preparing for installation. After you have gathered together all the materials necessary for installation, begin the installation process.

   - If you do not want to enable the secure login feature or name access levels and categories, and you do not need to customize NOS or any products, go to the section that describes the standard installation process.

   - If you want to customize your system (that is, you want to enable the secure login feature, name access levels and categories, or change NOS and its products in any manner), go to the section that describes the customized installation process and follow the directions for preparing for a customized installation.

2. Perform a level 0 deadstart of the system in unsecured mode using the released deadstart tape. (That is, do not include the CMRDECK entry that defines the system as a secured system.)

3. Install NOS and its products. That is, modify the system software that is released on the deadstart tape, and then create a new deadstart tape of the modified system. Follow the steps in the NOS Version 2 Installation Handbook for installing NOS and its products. While you are installing your system, do the following:

   - When you create your deadstart decks, include the security entries necessary for your site. Refer to CMRDECK Entries, EQPDECK Entries, and IPRDECK Entries in this section.

   - When you install CCP, select security characters if you want to enable the secure login feature.[3]

   - When you create or modify your Network Configuration File (NCFFILE), include entries to define the access limits for the Network Processing Unit communication lines.[4]

   - When you run the NOS installation procedure, define your own names for security access levels and categories.

   After you have run all the procedures necessary to install NOS and its products, create a new deadstart tape of the system.

4. Perform a level 0 deadstart of the system in secured mode using the new deadstart tape.

[3] CDCNET networks do not require special installation options to enable the secure login feature.

[4] It is not possible to define access limits for communication lines in a CDCNET network.

5. During the installation process, the SYSGEN procedure created a default validation file for the purpose of loading permanent files onto the system. By default, user SYSTEMX is the only user with security administrator privileges. Assign security administrator privileges to some user other than SYSTEMX. To do this, run a job similar to the following:

```
X.DIS.
USER,SYSTEMX,SYSTEMX.
MODVAL,OP=Z./username,SAC=ALL,SAV=ALL.
DROP.
```

6. Assign new passwords and expiration dates to the default user names and delete the security administrator privilege from user name SYSTEMX. Because the default user names and passwords are public (that is, they appear in system and user documentation), they must be changed to site-selected names on a secured system.

7. Assign security privileges to users of the secured system. To do so, you must use the MODVAL utility to specify values for the SAC, SAL, and SAV parameters for user names. MODVAL input parameters are described in appendix B.

CMRDECK Entries

The presence of the OPSECM entry in the CMRDECK defines the system as a secured system. This entry not only sets the security mode for the system, but also determines how, if at all, the system access level limits can be altered during system operation.

The format for the OPSECM entry is:

```
OPSECM=m
```

The possible values for m and the effects of each value are summarized in table 4-1.

Table 4-1. Security Mode Options

m Security Mode	Initial System Access Level Limits	Function of SECURES DSD Command
0 Unsecured	N/A	N/A
1 Secured	SECURES IPRDECK entry or SECURES DSD command.	Set, raise, or lower system access level limits.
2 Secured	SECURES IPRDECK entry only.	Raise system access level limits only.
3 Secured	SECURES IPRDECK entry only.	Command is invalid.

If you do not specify a value for m in the OPSECM entry, or if you neglect to include the OPSECM entry, the system operates in unsecured mode.

EQPDECK Entries

There are two EQPDECK entries that apply to a secured system:

- ACCESS
- INITIALIZE

Include an ACCESS entry and an INITIALIZE entry for each device that will be used on the secured system.

ACCESS Entry

The ACCESS entry sets the range of access levels for a particular device. This entry applies to mass storage, tape, unit record, and multiplexer equipment types.

The format for the ACCESS entry is:

```
ACCESS,lowerlevel,upperlevel,list.
```

lowerlevel

The lower access level limit for the device; this indicates the lowest access level of files that the device can process while the system is running in secured mode.

upperlevel

The upper access level limit for the device; this indicates the highest access level of the files that the device can process while the system is running in secured mode.

list

List of EST ordinals of devices to which the access level limits apply.

The value of the lowerlevel and upperlevel parameters can be LVL0 through LVL7, if your site uses the default access level names. If your site defines its own access level names, use those names for lowerlevel and upperlevel. The value of upperlevel must be equal to or greater than lowerlevel.

INITIALIZE Entry

After each ACCESS entry, add an INITIALIZE entry for the mass storage devices that will be used on the secured system.

The format for the INITIALIZE entry is:

INITIALIZE,AL,est_1,est_2,...,est_n.

est_i

The parameter est_i specifies the EST ordinal of the mass storage device to be initialized.

For more information about the INITIALIZE entry, refer to the NOS Version 2 Analysis Handbook.

IPRDECK Entries

There are four IPRDECK entries that apply to secured systems:

- SECURES
- SECCATS
- OQSH
- MEMORY CLEARING

SECURES Entry

The SECURES entry sets the initial access level limits for the system and for each origin type. Access level limits are the highest and lowest access levels that are allowed for a particular origin type or, in the case of system access level limits, for any job or file on the system.

If you do not include a SECURES entry in the IPRDECK, the default limits for the system and for all origin types is the lowest level (LVL0 or the site-defined equivalent). The default limits for all origin types for which no SECURES entry is made are the system access level limits. If OPSECM=1 has been specified in the CMRDECK, you can enter the initial system access level limits either with a SECURES entry in the IPRDECK or with a SECURES DSD command while the system is running.

The format of the SECURES entry is:

```
SECURES,ot,LA=lowerlevel,UA=upperlevel.
```

ot

Origin type

BC	Batch
RB	Remote batch
IA	Interactive
SY	System

lowerlevel

The lower access level limit for the origin type or for the system.

upperlevel

The upper access level limit for the origin type or for the system.

The value of the lowerlevel and upperlevel parameters can be LVL0 through LVL7, if your site uses the default access level names. If your site defines its own access level names, use those names for lowerlevel and upperlevel. The value for upperlevel must be equal to or greater than the value of lowerlevel.

SECCATS Entry

The SECCATS IPRDECK entry determines which, if any, access categories are to be used in the secured system.

The format of the SECCATS entry is.

SECCATS=cat_1,cat_2,...,cat_n.

cat_i

Name of an access category to be used in the secured system. Default access categories are CAT00 through CAT31; if your site defines its own access category names, you should use those names as parameters.

The following forms of the SECCATS entry have special meanings:

SECCATS=ALL.

Enables all access categories.

SECCATS=NULL.

Disables all access categories.

OQSH Entry

The OQSH IPRDECK entry sets the access level threshold for output files that are to receive special operator handling. Files whose access levels are equal to or greater than the threshold level remain queued until the operator releases them.

The format of the OQSH entry is:

OQSH=level.

level

Access level name (LVL0 through LVL7, or a site-defined access level name). Output files whose access level is equal to or greater than the access level corresponding to this level are to receive special handling. If LVL0 (or the site-defined equivalent) is specified for level, if no value is specified for level, or if no OQSH entry is present, no special handling goes into effect.

The operator can change the output queue special handling threshold while the system is running by using the OQSH DSD command.

MEMORY CLEARING Entry

NOS automatically clears central memory and extended memory before assigning them to a job. The MEMORY CLEARING IPRDECK entry enables or disables an additional memory clearing feature. When memory clearing is enabled, the system clears all central memory and extended memory associated with a job whenever it is released from the job. This means that whenever a job rolls out, has its field length reduced, or completes, no data from that job is left in memory.

This additional memory clearing feature causes system overhead. Thus, because NOS automatically clears memory before assigning it to a job, you must decide whether this extra memory clearing option is necessary for your site.

Here are the formats of the MEMORY CLEARING entry:

```
ENABLE,MEMORY CLEARING.

DISABLE,MEMORY CLEARING.
```

The released default is that memory clearing is disabled.

Secure Login Feature

During the Communications Control Program (CCP) portion of the installation process, you have the opportunity to enable the secure login feature by selecting security characters. The secure login feature guarantees that the terminal user can request a connection to the Network Validation Facility (NVF) regardless of any action by a host program. This prevents a malicious user from illegally acquiring the user names and passwords of other users by means of a login masquerade program that executes from a terminal and imitates the login dialogue of users for the purpose of intercepting their login information.

When the terminal user enters the security character in a specific sequence (refer to the NOS Version 2 Reference Set, Volume 3), CCP terminates any current connection and either reconnects the user to the host computer or prompts the user to select or connect to a host computer.

The security character must be a 7-bit character (specified as a hexadecimal number) that is within the code set of the terminal. The character is restricted to the values $03 through $1F, $21 through $2F, $3A through $40, $5B through $60, and $7B through $7E. The security character must not have the same value as the abort block, backspace, user break 1, user break 2, cancel, network control, end-of-line, or end-of-block character. Refer to the Network Definition Language Reference Manual for the default values for these characters. For any terminal class for which a security character is not specified (that is, if the value equals $00), the secure login feature is not activated.

You must ensure the integrity of the network configuration files (NCF and LCF) to prevent the subversion of the secure login feature.

CDCNET networks do not require special installation options to enable the Secure Login Feature.

The parameters shown in the table 4-2 can be set in deck SECURITY.

Table 4-2. Security Character Parameters

Parameter	Default	Terminal
SCAN2741	$00	Asynchronous non-2741 terminals
SCA2741	$00	Asynchronous 2741 terminals
SCMD4A	$00	Mode 4A terminals
SCHPOST	$00	HASP postprint terminals
SCHPRE	$00	HASP preprint terminals
SCB2780	$00	IBM 2780 terminals
SCB3780	$00	IBM 3780 terminals
SCXPAD	$00	X.25 packet assembly/disassembly (PAD) terminals
SCXUSER	$00	X.25 user-defined terminals

Network Configuration File Entries

While building the system network configuration file (NCFFILE), specify the upper access level limit for each communication line to the network processor unit (NPU). To specify this limit, use the AL=accesslevel parameter on the LINE or GROUP statement for each communication line. This value determines the highest access level for IAF job processing and for remote batch job submission through RBF. Refer to the Network Access Method Version 1 Network Definition Language Reference Manual for complete descriptions of LINE and GROUP statements.

The access level limits for the communication line connected to the two-port multiplexer on 180-class models are not set in this manner. The access level limits for this communication line are set by the ACCESS EQPDECK entry (described earlier in this section) for the EST ordinal of the two-port multiplexer.

It is not possible to define access limits for communication lines in a CDCNET network.

Access Level and Category Names

You can use the default names for access levels (LVL0 through LVL7) and categories (CAT00 through CAT31), or you can define new names that have special meaning for your site. These names are defined in micros in common deck COMSMLS.

To define new names for access levels and categories, create an indirect access permanent file prior to executing the installation procedure NOS. This file should contain Modify directives to edit the micros in deck COMSMLS corresponding to the access level and/or category names you want to define for your site.

The following example shows a file named ACCESS created for a site that has the four access level names UNCLASS, CONFDL, SECRET, and TOP and the four access category names PERSON, BUDGET, PROCURE, and INVEST.

```
*IDENT ACCESS
*DECK COMSMLS
*DELETE 12,15
ALM0 MICRO  1,,$UNCLASS$            ACCESS LEVEL 0
ALM1 MICRO  1,,$CONFDL$             ACCESS LEVEL 1
ALM2 MICRO  1,,$SECRET$             ACCESS LEVEL 2
ALM3 MICRO  1,,$TOP$                ACCESS LEVEL 3
*DELETE 24,27
ACM00 MICRO  1,,$PERSON$            ACCESS CATEGORY 00
ACM01 MICRO  1,,$BUDGET$            ACCESS CATEGORY 01
ACM02 MICRO  1,,$PROCURE$           ACCESS CATEGORY 02
ACM03 MICRO  1,,$INVEST$            ACCESS CATEGORY 03
```

These directives replace the names LVL0 through LVL3 (symbols ALM0 through ALM3) and CAT00 through CAT03 (symbols ACM00 through ACM03) with site-selected names. The names LVL4 through LVL7 and CAT04 through CAT31 remain as default names; this site would deactivate those names using the SECURES and SECCATS IPRDECK entries as follows:

```
SECURES,SY,LA=UNCLASS,UA=TOP.
SECCATS=PERSON,BUDGET,PROCURE,INVEST.
```

In order to incorporate the new access level and access category names into the secured system, specify the Modify directives file in the USERF parameter in the NOS installation procedure call. The format of the NOS installation procedure call for the above example would be:

```
BEGIN,NOS,INSTALL,USERF=ACCESS.
```

The NOS procedure reassembles all decks that call COMSMLS and incorporates the site-defined names.

Appendixes

Glossary A

A

Access Category

A classification marking assigned to a permanent file by the file owner to restrict access of the file to a particular group of users. A secured system supports up to 32 access categories, and each user is authorized to use some, all, or none of those categories. Refer to Job Access Category Set and Security Access Category Set.

Access Level

A property of each file, job, and equipment on a secured system that is used to indicate the sensitivity of information in the file or job, or the sensitivity of information that can be processed by the equipment. On a secured system, there are up to eight access levels corresponding to increasing levels of sensitivity, and each user is authorized to access some or all of those levels. Refer also to Equipment Access Levels, File Access Level, Job Access Level, and System Access Levels.

Access Level Limits

Upper and lower access levels which define a range of levels permitted for a particular user, device, origin type, and an entire system. Refer to Job Access Level Limits.

Alternate User Name

A user name specified in a permanent file request which indicates the action is to be taken on a user name different from the one under which the job is running.

E

Equipment Access Levels

A range of access levels specified for each equipment on a secured system. In order for a file to be stored on, printed on, or copied to a given equipment, the file's access level must be within the equipment access levels for that equipment.

F

File Access Category

A property of a permanent file used by the creator of the file on a secured system to restrict access of the file to a particular group of users. A secured system supports up to 32 access categories, and each user is authorized to use some, all, or none of those categories. Refer also to Job Access Category Set and Security Access Category Set.

File Access Level

A property of each file on a secured system used to indicate the sensitivity of information contained in the file. A file is assigned the current job access level by default when it is created or stored; the file creator may specify any access level for that file that is within the set of access levels valid for the job, the system, the file creator, and (for interactive jobs) the communication line to the host mainframe. Any user who accesses a file on a secured system must be validated for the access level of the file. Refer also to Access Level, Job Access Level, and Job Access Level Limits.

J

Job Access Category Set

On a secured system, a set of access categories is set when the job is initiated. This set is the intersection of the user's set of validated access categories and the system access category set. Refer also to File Access Category and Security Access Category Set.

Job Access Level

On a secured system, each job has an access level. This is the default access level that is assigned to files that are created by the job. A job's initial access level is the lower access level limit for the job. The job's access level is automatically raised to the access level of any file from which information is read. The job access level can also be changed by the user. Refer also to Job Access Level Limits.

Job Access Level Limits

An upper limit and a lower limit that determine the range of access levels that are valid for a particular job on a secured system. All files used in a given job must have an access level within the job's access level limits.

P

Password

1. A character string used to prove a user identification.

2. A character string used to obtain access to a file.

S

Secured System

A system in which the multi-level security (MLS) feature has been enabled during deadstart. A secured system protects information by enforcing restrictions based on access levels and access categories and restricts many sensitive system functions to security administrators.

Security Administrator

A secured system prevents users and operators from performing certain functions that could result in the unauthorized disclosure of information. These functions can be performed only by a person who is designated a security administrator. A security administrator is always authorized to access the highest level of information stored on the system. This person performs functions in the areas of installation, user validation, system operation, and system maintenance.

Security Count

The number of security violations a user has left before being denied access to the system. Each user's security count is stored in the validation file.

Security Unlock Status

This status of the system console applies only to a secured system and must be set by a security administrator. The console must be in security unlock status in order for the security administrator to perform certain restricted functions on a secured system.

Security Access Category Set

On a secured system, a set of access categories is set during a level 0 deadstart. This set may consist of some, all, or none of the 32 possible access categories. While the system is running, users may only use access categories that are within the set of system access categories. Refer also to File Access Category and Job Access Category Set.

System Access Levels

On a secured system, a range of access levels is set during a level 0 deadstart. This range may contain some or all of the eight possible access levels. While the system is running, users, may use only access levels that are within the range of system access levels.

U

Unsecured System

A system in which the optional security mechanism has not been enabled during deadstart. The restrictions based on access levels and access categories are not enforced on an unsecured system.

V

Validation File

A system file containing validation information for all users (user names, passwords, security permissions, resources allowed, and so on).

User Validations B

Validation information for all users of a NOS system is contained on the VALIDUS file. The VALIDUS file is a special system file maintained as a direct access permanent file under user name SYSTEMX (user index 377777). The VALIDUS file is created and managed by MODVAL and can be updated from a system origin job only. In a secured system, security administrator privileges are required to update VALIDUS.

Validation File Manager

The validation file manager, MODVAL, can be executed from the system console (system origin job) or from a batch job. MODVAL can directly update the VALIDUS file only from a system origin job (using input directives or the K display). When run from a batch job, MODVAL cannot access the VALIDUS file; either a copy of the new file or a directive file is established as a local file and processed later by a system origin job to update the VALIDUS file.

NOTE

The SUI command is invalid in a secured system. Use the USER command in place of the SUI command in the job running MODVAL in a secured system.

MODVAL Input Directives

An input directive enters user names under a MODVAL create run and modifies existing user names under an update run. The format of the input directive is:

/username,parameter$_1$=data$_1$,...,parameter$_n$=data$_n$.

where username is the one- to seven-character user name being referenced and parameter$_i$=data$_i$ is a system usage definition for this user.

The following is a list of parameters that deal with user security validations. A complete list of parameters used with MODVAL is provided in the NOS Version 2 Administration Handbook.

Parameter	Description
AW=perm	Access word validation. perm is a four-character identifier that toggles a particular permission bit in the access word. For each bit that is set, a special permission is allowed to the user. The bit is set when the identifier is first encountered and cleared if the identifier is used again. A maximum of 36 entries per record is allowed. Blanks are suppressed.

The following permission bits are defined in the access word.

perm	Bit	Description
CPWC	0	User can change batch and interactive passwords.
CTPC	1	User can use the ACCESS subsystem commands (terminal user only).
CLPF	2	User can create direct access permanent files.
CSPF	3	User can create indirect access permanent files.
CSOJ	4	● User can have system origin capability from any job origin if the debug option is turned on by the operator. ● User can assign a device by specifying its EST ordinal. This does not require that the debug option be turned on. ● User can call the PP hardware diagnostics of the 881/883 pack reformatting utility FORMAT, if engineering mode is enabled.
CASF	5	User can access the file SYSTEM (with the COMMON,SYSTEM command).
CAND	6	User can request nonallocatable devices (for example, magnetic tape units).
CCNR	7	User can use the system without entering a charge or project number.
CSRP	8	User can issue removable auxiliary device commands.
CSTP	9	User has special TAF privileges of updating task libraries online.
CTIM	10	User will not be logged off because of timeout.
CUCP	11	User can access system control point facility.
CSAP	12	User has special accounting privileges.

Parameter	Description
AW=perm	Continued

perm	Bit	Description
CBIO	13	User has BIO subsystem privileges.
CPRT	14	User can use the PROTECT command to preserve extended memory.
CPLK	15	User can transfer permanent files between hosts.
CQLK	16	User can transfer queued files between hosts.
CUST	17	User can specify a logical identifier on Job or ROUTE command.
CNVE	18	User can access the NVE subsystem.
CMNT	19	User can use Remote Diagnostic Facility.
CNOP	20	User can control NPUs (that is, a user who is also validated for the CS application can become a controlling network operator).
CSAF	21	User can specify an alternate family.
CNRD	22	User can specify a charge and project number other than the default (specified by CN and PN parameterts).
COPR	23	User can specify a password without randomization.
CLTD	24	User is validated to specify preferred file residence as locked to disk on the SAVE, DEFINE, and CHANGE commands.
CCPI	25	User can omit personal id.
CACA	26	Concurrent access allowed to this user name.

By default, all new user names are created with CPWC, CLPF, CCNR, CSPF, CSAF, CNRD, COPR, CCPI, and CACA permissions, unless an AW parameter is entered. In this case, the user is created with only those permissions specified. To set or clear all permission bits in the access word, the following can be specified for perm:

ALL	Sets all permission bits in the access word.
NUL	Clears all permission bits in the access word.

EB=password	Encrypted batch password. A 14-digit octal encrypted password to be used for batch and system origin jobs.

Parameter	Description
EI=password	Encrypted interactive password. A 14-digit octal encrypted password to be used for interactive jobs.
PB=password	Batch password. A one- to seven-character password to be used for batch, remote batch, and system origin jobs. This parameter is processed the same as the PW parameter; however, it applies only to the batch password.
PI=password	Interactive password. A one- to seven-character password to be used for interactive jobs (login to any NAM application). This parameter is processed the same as the PW parameter; however, it applies only to the interactive password.
PW=password	A four- to seven-character password (A through Z, 0 through 9). Blanks are not significant. This parameter sets both the batch and interactive passwords. When PW is entered, the associated password expiration dates are set to the default value. The minimum required length for passwords can be changed by setting the MODVAL installation parameter RPWL to a value from 0 to 7. The PW parameter is required with OP=C unless the RP parameter is specified on the MODVAL command or the minimum required password length is zero. If this parameter is not required and is omitted, the user must enter a null password at login.
SAC=category	Security access categories. category is a one- to seven-character symbolic name that toggles a particular bit in the access category field (bits 31 through 0) of the security validation word. For each bit that is set, the corresponding access category is available to the user. Blanks are suppressed. These validations are checked only in a secured system.

The following access category bits are defined in the security validation word.

category	Bit	Description
CATnn	nn	User is validated for access category nn (00 is less than or equal to nn which is less than or equal to 31).

To set or clear all access category bits in the security validation word, the following can be specified for category:

ALL	Sets all access category validation bits.
NUL	Clears all access category validation bits.

The site can redefine the names associated with the access categories. These are default values, which are defined in common deck COMSMLS.

Parameter	Description
SAL=level	Security access levels. level is a one- to seven-character symbolic name that toggles a particular bit in the access level field (bits 43 through 36) of the security validation word. For each bit that is set, the corresponding access level is available to the user. Blanks are suppressed.

In a secured system, a user must be valid for at least one access level in order to use the system. These validations are not checked in an unsecured system.

The following access level bits are defined in the security validation word.

level	Bit	Description
LVL7	43	User is validated for access level 7.
LVL6	42	User is validated for access level 6.
LVL5	41	User is validated for access level 5.
LVL4	40	User is validated for access level 4.
LVL3	39	User is validated for access level 3.
LVL2	38	User is validated for access level 2.
LVL1	37	User is validated for access level 1.
LVL0	36	User is validated for access level 0.

By default, all new user names are created with LVL0 validation. If a SAL parameter is entered, the user is created with only those access levels specified.

To set or clear all access level bits in the security validation word, the following can be specified for level.

ALL	Sets all access level validation bits.
NUL	Clears all access level validation bits.

Parameter	Description
SAV=privilege	Security access valjdation. This parameter sets the privileges that apply to both secured and unsecured systems. privilege is a four-character designation that toggles a particular bit in the privilege field (bits 59 through 52) of the security validation word. For each bit that is set, the corresponding special permission is allowed to that user. A bit is set when the parameter is first encountered and cleared if the parameter is used again. Blanks are suppressed.

The following privilege bits are defined in the security validation word.

privilege	Bit	Description
CSAP	59	User has security administrator privileges. This bit cannot be cleared by its owner; that is, the user executing MODVAL and clearing this permission bit must not be the user whose permission is being cleared. This bit applies to secured systems only.
COLD	58	User can execute on-line diagnostics and other customer engineer operations. This bit applies to both secured and unsecured systems.
CPWX	57	User can assign user password expiration date using the XD or XT parameter on the PASSWOR command. This bit applies to both secured and unsecured systems.
CFPX	56	User can assign permanent file password and permit expiration using the XD or XT parameter on permanent file commands. This bit applies to both secured and unsecured systems.
CLJL	55	User can lower (downgrade) the access level of a job using the SETJAL command or macro. This bit applies to secured systems only.
CLFL	54	User can lower (downgrade) the access level of local or permanent files using the SETFAL or SETPFAL command or macro. This bit applies to secured systems only.
CWLF	53	User can write to or extend files that are at a lower access level than the user's job (write-down privilege). This bit applies to secured systems only.
CULT	52	User can write to unlabeled magnetic tapes. System origin jobs automatically have this permission. This bit applies to both secured and unsecured systems.

Parameter **Description**

SAV=privilege Continued

To clear or set all privilege bits in the security validation word, the following can be specified for privilege:

ALL Sets all privilege validation bits.

NUL Clears all privilege validation bits.

SC=count Security count. This parameter specifies the number of security conflicts allowed before the user is denied access to the system. The security count is decremented by the system when any security conflict occurs. A value of 77_8 indicates an unlimited security count; 0 indicates no access is allowed. If not specified, the default value is SC=50_8. The security count is not included as output from a LIMITS command.

XB=yymmdd Batch password expiration date. This identifier is the same as the XD parameter except that only the batch password expiration date is set.

XD=yymmdd Password expiration date. This identifier sets the password expiration date for the batch and interactive passwords to yymmdd. The default date is site-defined.

XI=yymmdd Interactive password expiration date. This identifier is the same as the XD identifier except that only the interactive password expiration date is set.

XT=term Password expiration date by term. This identifier adds a one- to four-digit expiration term value to the current date to calculate a new batch and interactive password expiration date. The term value can be from 0 to 4095 (7777_8). Decimal is assumed unless the post radix B is specified. The default term value is site-defined and is used to calculate the expiration date when neither XD nor XT is specified when creating a new user name.

XT=0 sets the password expiration date to immediately expire. This is done to temporarily disable a password without deleting it from the validation file.

The password expiration date can be set to nonexpiring by entering XT=4095 or XT=7777B or XT=*.

XTB=term Batch password expiration date by term. This parameter is the same as the XT identifier except that only the batch password expiration date changes.

XTI=term Interactive password expiration date by term. This parameter is the same as the XT parameter except that only the interactive password expiration date changes.

If the user name has security administrator privileges (CSAP security privilege), the expiration dates for both batch and interactive passwords are nonexpiring, and the security count is unlimited. These values are set automatically and cannot be changed.

Operator Commands and Utilities C

The operator interface to a secured system requires that a number of operator functions be performed by, or under the supervision of, a security administrator. This appendix lists and briefly describes NOS operator commands and utilities that should be restricted to users with security administrator privileges. These commands and utilities are discussed in the following subsections:

- Security of the System Console
- Restricted Commands and Utilities
- QDSPLAY Utility
- DIS Operation
- Security of Printed Output
- System Deadstart and Recovery

The NOS Version 2 Operations Handbook and Analysis Handbook contain complete information about the commands and utilities used for system operation.

Security of the System Console

On a secured NOS system, the system console should be kept in security lock status. You should put the console in security unlock status only to perform certain tasks. When you've completed your tasks, always put the console back in the security lock status.

While the system is in security lock status, the system does not process the following DSD commands:

- DEBUG
- DIS
- DISABLE,ENGR
- ENABLE,ENGR
- ENABLE,RDF
- QDSPLAY
- SECUREQ
- SECURES
- All memory entry commands

In addition, the memory displays (C, D, F, G, and M) can be used only to display the central memory resident portion of central memory and the system table area of extended memory.

The left screen display header always indicates the current status of the system console. The following three types of status for the console are possible.

Console Status	Left Screen Header Message	DSD Command to Set This Status
Lock/security lock	None	LOCK.
Unlock/security lock	UNLOCK	UNLOCK.
Unlock/security unlock	SECURITY-UNLOCK	UNLOCK,username,password.

LOCK Command

To put the console in security lock status, enter this command:

```
LOCK.
```

Any operator can enter the LOCK command. This command puts the system console in standard NOS lock status and security lock status.

UNLOCK Command

To put the console in security unlock staus, enter this command:

```
UNLOCK,username,password.
```

The UNLOCK command can be entered only by a user with security administrator privileges. This command puts the system console in standard NOS unlock status and security unlock status.

Restricted Commands and Utilities

The following DSD commands can be used only when the system is in security unlock status.

DEBUG

The DEBUG command toggles the system from debug to nondebug mode. When debug mode is set, the message DEBUG appears in the left screen display header. Debug mode provides system origin privileges to validated users and allows them to make modifications to the running system. Under normal system operation, you should not use debug mode.

DIS,jsn

The DIS,jsn command calls the job display utility (DIS) to the job sequence name jsn. The A and B displays for DIS automatically appear on the left and right console screen, respectively.

DSD Memory Displays

These displays show the contents of central memory (C, D, F, and G displays) and extended memory (M display). For complete descriptions of these displays, refer to the NOS Version 2 Analysis Handbook.

DSD Memory Entry Commands

The DSD memory entry commands are used to change the contents of central memory and extended memory. These commands can change either absolute locations or those relative to a specific job's reference address (RA). For complete descriptions of these commands, refer to the NOS Version 2 Analysis Handbook.

ENABLE,ENGR

This command enables engineering mode. When engineering mode is set, the message ENGR appears in the left screen display header. Engineering mode allows PP/hardware diagnostics and FORMAT/FDP to be executed by users with system origin privileges.

ENABLE,RDF

This command enables the Remote Diagnostic Facility (RDF). For more information refer to the SUBSYST L DISPLAY command found in the NOS Version 2 Operations Handbook.

SECUREQ,est,LA=lowerlevel,UA=upperlevel

This command changes the equipment access level limits for the unit record equipment with EST ordinal est. The parameters LA and UA specify the lower and upper access level limits for the unit record equipment.

SECURES,ot,LA=lowerlevel,UA=upperlevel

This command sets the access level limits for an origin type (SY, BC, RB, or IA). The parameters LA and UA specify the lower and upper access level limits. The default limits for each origin type are the system access level limits.

QDSPLAY Utility

The DSD utility QDSPLAY displays the contents of a queued file listed in the queued file table (QFT). For a complete description of this utility, refer to the NOS Version 2 Analysis Handbook.

DIS Operations

DIS displays information about a single job. In a secured system, the DIS,jsn command can be entered only while the system console is in security unlock status; this means that information about a job belonging to another user is available only to a security administrator. An attempt to display memory outside of a job's field length causes the message *SECURED AREA* to appear in the memory display.

Under DIS, the B display shows the exchange package area for the job. Central memory addresses relative to the job's reference address are used for the data and program displays.

Security of Printed Output

A secured NOS system provides safeguards for handling printed output of different access levels. These features include the following:

- Output files are restricted to devices that are validated for the access level of the files.
- The access level of a printed file is indicated on the banner page.
- The user has the option of printing the access level of printed output on each page of the output.
- The operator provides special handling to all output files whose access level is equal to or above a site-selected value.

The special handling feature for output files is implemented by two operator commands, OQSH and RELEASE. These commands can be entered when the system console is locked or unlocked.

OQSH

This command determines the access level of files that will receive special handling by the operator. The output queue special handling level is set initially during deadstart by the OQSH IPRDECK entry. The OQSH command can be entered at any time from the system console to change the current level. The format of the command is:

```
OQSH=level.
```

level

The name of the level (LVL0-LVL7). Files whose access level is equal to or greater than this number remain in the queue until released by the operator, If level=0 or is not specified, no files are held in the queue.

If the output queue special handling level is changed to a higher value, files whose access level is lower than the new level are released automatically.

RELEASE

This command allows the operator to release a file from the output queue whose access level is equal to or higher than the output queue special handling level. Output queue files and their access levels can be examined by using the DSD Q display. The RELEASE command can be entered at any time from the system console. The format of the command is:

```
RELEASE,jsn.
```

jsn

Job sequence name of the file to be released from the output queue.

Each output file that is released is processed by BIO. The restrictions based on device access levels and file access levels continue to apply.

System Deadstart and Recovery

Deadstart is the process that makes NOS operational. Deadstart is performed under a variety of circumstances, including the following:

- When initially installing NOS
- When installing site modifications to NOS
- After operating system or hardware failure
- After performing on-line maintenance
- After performing system tests

NOS provides four types of deadstart in order to recover the greatest amount of the operating system, user files, and user jobs possible under given circumstances. The four types of deadstart are called level 0, level 1, level 2, and level 3. The procedures for performing the different types of deadstart are described in detail in the NOS Version 2 Operations Handbook and the CYBER Initialization Package User's Handbook.

Secure Recovery With Level 3 Deadstart

The level 3 deadstart is used most frequently after an equipment or software failure in a production environment. Because this level of deadstart preserves the portion of the operating system resident in central memory and also recovers user jobs, queued files, and permanent files, a level 3 deadstart in a secured system preserves all the security properties of the system. This is the only type of deadstart that can be used for secure recovery of a secured system after a failure.

Level 2 and Level 1 Deadstart

Level 2 and level 1 deadstarts are not intended to be used for system recovery after a failure. A level 2 deadstart is performed in system test situations, while a level 1 deadstart is normally performed following system maintenance procedures. These two levels of deadstart restore the system and the operating environment from checkpoint files and do not preserve the contents of central memory. In addition, these levels of deadstart allow the operator to change some of the properties of the system that could affect the integrity of a secured system (in particular, they allow the operator to logically reconfigure mass storage and other peripheral devices). These two levels of deadstart do not provide secure recovery of a secured system.

Level 0 Deadstart

The level 0 deadstart is used to install a new operating system or to deadstart when a recovery deadstart is not possible. The memory confidence testing performed during a level 0 deadstart destroys the contents of central memory, including the existing copy of the operating system. Level 0 deadstart does not provide secure recovery of a secured system.

This type of deadstart is used to change the characteristics of a secured system. During level 0 deadstart, the operator can enter new values for the following:

- the OPSECM CMRDECK entry
- the SECURES, SECCATS, MEMORY CLEARING, and OQSH IPRDECK entries
- ACCESS and INITIALIZE EQPDECK entries

A level 0 deadstart is the only type of deadstart that allows these entries to be changed.

Maintenance Commands and Utilities D

This appendix lists and describes NOS maintenance commands and utilities that include parameters for dealing with ranges of file access levels on a secured system. In addition, this appendix briefly describes the SYSEDIT and 881/883 pack reformatting utilities. On a secured system, these utilities require that the user running them be a security administrator.

The NOS Version 2 Analysis Handbook contains complete information about all NOS maintenance commands and utilities.

Permanent File Utilities

Five utility processors maintain the NOS permanent file system. They control the dumping and loading of permanent files, the cataloging of permanent files in the system and on backup storage (archive) files, the copying of archived files to a job as local files, and releasing the disk space of files that reside on tape or cartridge alternate storage. Complete descriptions of the permanent file utilities can be found in the NOS Version 2 Analysis Handbook.

Permanent File Utility Parameters

The following parameters allow you to select a range of access levels for files when using the permanent file utilities. Although these parameters may be used on unsecured systems, they are particularly useful for secured systems.

Parameter	Description
LA=level	One- to seven-character name which specifies the lower limit of the range of access levels to process. If this parameter is specified, the UA parameter must also be specified. If neither LA nor UA is specified, the default is that all access levels are selected.
UA=level	One- to seven-character name which specifies the upper limit of the range of access levels to process. If this parameter is specified, the LA parameter must also be specified. If neither LA nor UA is specified, the default is that all access levels are selected.

The utilities described here have special functions or restrictions that apply to secured systems.

PFCOPY

PFCOPY extracts files from an archive file and copies them to one or more files local to the job. The way the files are copied depends upon the parameter options selected.

In a secured system, the access level from the file's PFC entry is used to assign each file to be copied to an appropriate mass storage device. If no device can be found that allows the file to reside there, the PFCOPY utility skips the file and issues a diagnostic message.

PFDUMP

PFDUMP copies (dumps) permanent files to backup storage in archive files. Dumps can be reloaded by the PFLOAD utility and can be accessed by the PFATC and PFCOPY utilities for cataloging and copying. PFDUMP issues messages to the dayfile indicating how many files were dumped and how many files were not dumped due to errors.

In a secured system, PFDUMP determines the maximum range of access levels that can be dumped. If the LA and UA parameters are specified to select access level limits, these limits are used. If no access level limits are specified, PFDUMP uses the device limits determined by taking the lowest lower access limit and the highest upper access limit of all the devices to be dumped. The range of possible access levels must be within the system access level limits or PFDUMP aborts the job and issues a diagnostic message. The range of possible access levels must also be within the equipment access level limits for the equipment (tape or mass storage) assigned to the archive file and verify file (if one is being written). If the levels specified are not within the equipment access level limits, PFDUMP aborts the job and issues a diagnostic message.

PFLOAD

PFLOAD loads archived files produced by the PFDUMP utililty back into the permanent file system. The load can reestablish the permanent file system exactly as it was at the time of the dump, or can load only a desired subset of files on the archive file (as indicated by specified parameter options). PFLOAD issues messages to the dayfile indicating how many files were loaded and how many files with errors were encountered.

In secured systems, PFLOAD verifies before loading a file that the device where the file is to reside (as selected during installation) is appropriate for the file's access level. If the device is not appropriate, PFLOAD skips the file and issues a diagnostic message.

Queue File Utilities

Several utility programs provide control over queued input and output files. Complete descriptions for these utililties can be found in the NOS Version 2 Analysis Hanbook.

Queue File Utility Parameters

The following parameters allow you to select a range of access levels for queued files or to change the access level of queued files when using the queue file utilities. Although these parameters may be used on unsecured systems, they are particularly useful for secured systems.

Parameter	Description
LA=level	One- to seven-character name which specifies the lower limit of the range of access levels to process. If this parameter is specified, the UA parameter must also be specified. If neither LA nor UA is specified, the default is that all access levels are selected.
NAL=level	QALTER can change the access level of a queued file. The new access level must be within the origin type limits for the file and within the device limits of the device on which the file resides. This option is restricted to users with security administrator privileges.
UA=level	One- to seven-character name which specifies the upper limit of the range of access levels to process. If this parameter is specified, the LA parameter must also be specified. If neither LA nor UA is specified, the default is that all access levels are selected.

The utilities described here have special functions or restrictions that apply to secured systems.

QALTER

QALTER lists, displays, and/or alters the routing and other information about active queued files. The utililty selects files for processing according to a variety of user-specified criteria. QALTER purges selected files from the system when the user specifies OP=PR.

The user may alter the access level of selected files using the NAL=level parameter (this requires security administrator privileges).

QDUMP

QDUMP dumps selected queued files from a single device, a family of devices, or all devices on the system. These queued files can be dumped either to tape or to disk. When active queued files are dumped, the QFT is searched to obtain the proper file. When inactive queues are dumped, the IQFT is searched. QDUMP also provides a listing of all files dumped with information about each file processed.

For a secured system, QDUMP determines the maximum range of access levels that can be dumped. If the LA and UA parameters are specified to select access level limits, these limits are used. If no access level limits are specified, QDUMP uses the device limits determined by taking the lowest lower access limit and the highest upper access limit of all the devices to be processed. If this range of access levels is not within the system access level limits, QDUMP aborts the request and issues a diagnostic message.

QLOAD

QLOAD processes the dump files generated by QDUMP or other utilities using the same format. QLOAD can also list the contents of a dump file without loading any files.

For secured systems, the access level for each file selected to be loaded must be within the file's origin type limits and the file must be assigned to an appropriate mass storage device. If a selected file cannot be loaded, QLOAD issues a diagnostic message and skips that file.

QMOVE

QMOVE moves queued files from one mass storage device to another. It also produces a listing of all files moved with information about each file processed.

For secured systems, the destination device for each file selected to be moved must have access level limits that are appropriate to accept the file. If the file is being reactivated as well as moved, its access level must be within the job's origin type limits. If a selected file cannot be moved, QMOVE issues a diagnostic message and skips that file.

QREC

QREC deactivates or activates selected queued files and purges selected inactive queued files.

For secured systems, the access level for any file selected to be reactivated must be within the access level limits for the job's origin type. If a selected file is outside these limits, QREC issues a diagnostic message and skips that file.

SYSEDIT

The SYSEDIT command provides a method of performing modifications to the system library after the system has been loaded. A job containing a SYSEDIT command must either be a system origin job, or the user must be validated for system origin privileges and the system must be in debug mode. In addition, to prevent unauthorized modification of the operating system on a secured system, SYSEDIT is allowed only from jobs with security administrator privileges.

The use of SYSEDIT in a production environment may cause unpredictable results and is not recommended. The system should be idle to ensure predictable results.

For complete information about using the SYSEDIT command, refer to the NOS Version 2 Analysis Handbook.

881/883 Pack Reformatting Utility

FORMAT is a CPU program which operates in conjuction with FDP, a PP program, to maintain and reformat 881/883 disk packs. It is used to perform the following functions:

- Retrieve factory-recorded manufacturing data, factory-recorded flaw data, and utility flaw data from a factory-formatted disk pack.
- Set or clear sector and track flaws on a factory-formatted disk pack.
- Restore address fields of a previously factory-formatted disk pack. (This function is used only in the event that addresses on the pack are lost.)

For jobs other than system origin jobs, engineering mode must be set at the system console in order to run the FORMAT program. On a secured system, engineering mode can be set only if a security administrator has placed the console in security unlock status.

For complete information about the 881/883 pack reformatting utility, refer to the NOS Version 2 Analysis Handbook.

Secured System User Commands and Macros E

A number of commands and macros have been added to the NOS user interface for use in secured systems. Several other commands and macros contain parameters that are for use in secured systems.

Secured System User Commands

The user commands shown in table E-1 are for use in secured systems. With the exception of the SETJAL command, these commands perform the same functions on unsecured systems (for example, assigning an access category as an attribute of a permanent file), although an unsecured system does not enforce the security restrictions based on the access levels and categories included in the commands.

Table E-1. Secured System Commands

Command	Effect in secured System	Effect in Unsecured System
SECHDR	Establishes security header format for printed output.	Same as secured system.
SETFAL	Sets access level of local file.	Same as secured system.
SETJAL	Sets job access level.	No effect.
SETPFAC	Sets security access category for permanent file.	Same as secured system.
SETPFAL	Sets access level for permanent file.	Same as secured system.

User Commands with Access Level Parameters

The following user commands have an optional access level parameter for use on secured systems:

ASSIGN
DEFINE
JOB
LABEL
REQUEST
RESOURC
SAVE

On unsecured systems, the system reads the access level parameter but does not use that parameter to restrict any activity in the user job. Refer to the NOS Version 2 Reference Set, Volume 3, for complete information about these commands.

Secured System User Macros

The macros shown in table E-2 are for use in secured systems. With the exception of the SETJAL macro, these macros perform the same functions on a secured system as on an unsecured system (for example, assigning an access category set as an attribute of a permanent file), although an unsecured system does not base security restrictions on the access levels and categories set by the macro.

The macros defined in common deck COMCMAC can be accessed by specifying alternate systems text PSSTEXT; the macros defined in common deck CPCOM can be accessed either through the system OPL or by specifying alternate systems text SYSTEXT or NOSTEXT.

Table E-2. Secured System Macros

Macro	Common Deck	Effect in Secured System	Effect in Unsecured System
GETJAL	COMCMAC	Returns job's current access level.	Returns zero value.
GETSSM	COMCMAC	Returns system security mode.	Returns zero value.
SETFAL	COMCMAC	Sets access level of local file.	Same as secured system.
SETJAL	COMCMAC	Sets job access level.	Request is ignored.
SETPFAC	CPCOM	Set access category set for permanent file.	Same as secured system.
SETPFAL	CPCOM	Sets access level for permanent file.	Same as secured system.

User Macros with Access Level Parameters

Users can specify a file access level on certain COMPASS macro calls and RA+1 requests. For LFM and PFM macros, an access level is entered or returned in bits 38 through 36 of FET+4; bit 39 of FET+1 must be set prior to the macro call to indicate that an access level is to be found or returned in FET+4. The following LFM and PFM macros make use of the optional access level parameter in a secured system:

ATTACH
DEFINE
GET
LABEL
REQUEST
SAVE
STATUS

On an unsecured system, LFM and PFM process these macros if an access level is specified, but the system does not use that parameter to restrict any activity of the user job. Refer to the NOS Version 2 Reference Set, Volume 4, for complete information about these macros.

Security Audit Reduction Tool F

The NOS Security Audit Reduction Tool, SECART, is provided to assist the security administrator by digesting and reducing the volume of information which must be analyzed to audit security-related user activities.

SECART is designed to process messages recorded in system dayfile and account logs and to produce printouts and files useful for security auditing.

- Printouts include an exception report listing occurrences of selected messages, statistics on account message identifier usage, tabular data which may be used to trace messages and jobs to individual users, and unit job printouts which may be used to audit the activities of individual users.

- Files produced include an annotated composite log containing messages from both dayfile and account logs merged into a near natural cause-and-effect time sequence, a sorted copy of the composite log which groups messages into unit jobs, and a copy of the trace table data.

SECART is intended to be used as part of routine site operation procedures and as needed to assist the security administrator in analyzing activities of interest.

SECART has been designed to be as efficient as practical. As a general rule, it is capable of processing an entire day's volume of account and dayfile log messages in a small fraction of an hour on the same class machine as that which produced the logs, provided the audit directives used do not result in massive printouts. A rerun capability that uses outputs from an initial run is provided, which significantly reduces processing time and allows changing job selection criteria.

SECART Functions

Since a number of decisions, such as selection of jobs for printout, must be deferred until necessary information has been processed and may also involve human judgement, the functions of this utility are performed in a series of distinct processing phases, some of which may be efficiently rerun by saving and reusing outputs from earlier phases. Thus, SECART is designed to accept raw logs and/or files from previous runs and to select at which phase to begin processing based on the combination of files provided and specified. SECART directives used on reruns may be different from an initial run to reflect changed job selection criteria.

Initialization Phase

SECART initialization consists primarily of accepting file names via program parameters, interpreting audit directives and storing arguments to be used in subsequent phases, and loading data from files into internal tables as applicable. The presence/absence of filenames for raw, composite, and sorted logs then determines at which phase processing will begin. If raw log files are provided, SECART assumes that this is an initial run and begins with the raw logs processing phase. If neither of the raw log files is provided, it will attempt to start processing at the job sort phase.

Raw Logs Processing Phase

The main functions of this phase are:

1. To screen the raw logs and collect information in tables needed to control unit job selection and printout.
2. To merge and annotate log messages to facilitate subsequent sorting and selection processes.

The bulk of SECART processing occurs in this phase. Thus, reruns made possible by retention of outputs from this phase require significantly less processing than initial runs with raw logs.

SECART is designed to process a pair of files containing raw dayfile and account log messages produced on the same machine for matching or overlapping time periods. Thus, one of the first functions performed in this phase is verification that the raw log files are indeed from the same system and date. This is accomplished by a preliminary search for the system version, title, and log date recorded in each log and a check for mismatches. The system title and version acquired in this manner will be displayed in printout headers.

Since much of the information in the raw logs is redundant, overlaps, and changes with time, the most efficient method of digesting this information is to process both logs in time sequence (that is, in parallel). However, the time recorded with each message is precise only to the second, which leads to sequence ambiguities between logs. This sequence ambiguity between dayfile and account messages with identical times is resolved by a built-in bias to process most dayfile messages before account messages. Exceptions are those account log messages relating to the creation and recovery of jobs and system logs and deadstart messages which are biased as a special case to be processed before dayfile messages. This generally reflects a cause-and-effect relationship and allows SECART to obtain information from both logs in a near natural sequence. It also makes the composite log and screening outputs of this phase more readable, although messages for a given second will clump together according to source log. To reduce confusion, messages are annotated to indicate source log.

SECART is designed to make maximum use of available information. It is capable of functioning with either log separately or in combination and thus can recover audit data from incomplete or fragmented logs. In circumstances where there is potential for tampering with system logs, this cross-correlation could be used to detect possible gaps or inconsistencies between logs. The primary purpose for parallel processing of raw logs however, is to combine messages from both logs in proper sequence so as to provide a broader view of the recorded events and to better establish the processing context in which the events occurred.

SECART directives may be specified to screen messages from both logs for printout as they are processed in this raw logs processing phase. Account messages may be selected by message identifier codes, dayfile messages by character string, and messages from both logs by time period. This printout of time periods and messages screened from both logs may be used as an exception report to detect messages and jobs which may warrant further investigation either via SECART reruns or other processing. By default, SECART will print out messages for a number of security-related events including dayfile messages for level 0 deadstart, all system console operator dayfile messages, account messages for recovered jobs (ARRQ), changes to a job's JSN (ACSC) or user identification (ABUN, ACUN), and other account messages as defined in the message identifier database (which can easily be changed by the security administrator). SECART will also insert messages in the screening printout for some of these events.

A primary output of the raw logs processing phase is an annotated composite log file which is input to the job sort phase. This file should be retained for potential reruns or other processing (for example, access via text editor to examine other time periods or correlate events involving multiple jobs).

Account Message Statistics

On an initial run, at the end of the raw logs processing phase and following the screening printout, SECART produces a printout of account log message identifier statistics. This printout contains a list of all encountered and defined account message identifiers with a count and usage by job service class.

SECART is designed to automatically accommodate new or undefined account message identifiers. Defined message identifiers are obtained from the message identifier database along with a brief description. The absence of an entry in the message identifier database results in a blank description field in the printout. This printout is also useful for detecting the presence, absence, or unusual volume of message identifiers which may warrant further investigation. It is also a reference for interpreting account log messages in other printouts and for routine auditing of overall system usage patterns.

Account message identifiers selected or defaulted for printout during the raw logs processing phase are flagged in this printout. Selections may be on an ad hoc basis via SECART directives, or on a default basis via flags set in the message identifier database. A summary of selected message identifiers versus overall message usage is also printed at the end of the table.

JSN Trace Table

The key element linking a user identification with a job and all log messages issued on behalf of that job is the job sequence name (JSN), which is recorded in every message issued to either log. Identification of the user responsible for a job is recorded in messages at job initiation and binds the username to the JSN. A primary function of the raw logs processing phase is to extract this information and build a JSN trace table to record user identification and other data associated with each job. The JSN trace table corresponding to a given set of raw logs thus becomes an index to the composite and sorted job logs. Therefore, the trace table file output at the end of the raw logs processing phase should be saved for input to subsequent SECART reruns.

Since JSNs are assigned by the operating system in a linear sequence which is reset with each initial (level 0) deadstart, it is necessary to resolve ambiguities between identical JSNs generated in different operating periods bounded by level 0 deadstarts. SECART resolves these ambiguities by assigning a code letter to represent each operating period and using it as a prefix to JSNs encountered in that period. In the raw logs processing phase, this deadstart-unique code is added to messages written to the composite log file, appears in the screening printout, and is recorded with each JSN in the trace table. Thus, for SECART purposes the JSN is extended to include the deadstart-unique prefix in the format:

    d.jsno

where d is the deadstart-unique code. This extended JSN then becomes the data element which unambiguously relates messages in the merged log to JSN trace table entries and provides a single key for sorting out individual jobs. This is the primary function of the raw logs processing phase.

Using a single letter to represent each deadstart period is much more compact and efficient than associating a date and time with each JSN to ensure uniqueness. Since a JSN needs to be unique only within the context of a SECART run, a single letter is sufficient to handle up to 26 operating periods without replication. To avoid wrap around, the deadstart-unique code letter defaults to the first letter of the alphabet unless a trace table file from a previous run is loaded, in which case SECART picks up the code letter from the last JSN trace table entry loaded. Thereafter, the code letter is incremented whenever SECART recognizes messages which signal the occurrence of a level 0 deadstart. In the composite log printout, a level 0 deadstart is indicated by a blank space and message displaying the code letter assigned to that operating period.

There are two reasons for loading a JSN trace table from a previous SECART run:

1. To allow SECART to link recovered jobs to their original JSN and possibly influence unit job printout selection.

2. For SECART reruns starting at the job sort or job extraction phases.

Recovered job linkage is relevant only where system queues are recovered, thus a trace table file from a previous log period is optional for initial runs. For reruns, a trace table file is required and must correspond to the composite or sorted jobs log provided as input. When a trace table file is provided for an initial run, the deadstart-unique code letter is picked up from the JSN of the last trace table entry loaded and the risk of spanning more than 26 operating periods or exceeding the capacity of the JSN trace table rise accordingly. When no previous trace table is loaded, extended JSNs are unique only with respect to the set of logs processed in that SECART run.

At the end of the raw logs processing phase, after printout of the account log message identifier statistics, SECART writes the contents of the JSN trace table to the trace table file. This file should be saved if there is a potential for rerunning SECART for the same log period or if there are entries for jobs which may be recovered in a subsequent log period.

The JSN trace table is maintained in memory to facilitate fast access via binary search on extended JSNs. The output trace table file will be in JSN sequence and must remain in that sequence to be used in a subsequent SECART run or rerun. After the trace table file is output, the table is printed with appropriate page headers and a final summary. If specified via directive, the printout will be sorted (via Sort/Merge 5) on the user-specified key fields. This printout provides an index to other SECART printouts and is the key to attributing log messages to a user identification. If a different trace table sort sequence is desired, SECART may be rerun with only the trace table file as input and the appropriate sort keys specified to yield the desired sequence.

Job Sort Phase

In an initial SECART run, the composite log produced by the raw logs processing phase is input to the job sort phase, which automatically follows. To rerun SECART beginning at the job sort phase, the composite log and corresponding JSN trace table files must be provided as input, and the raw logs must be omitted.

The job sort phase consists simply of sorting the composite log file keyed on the extended JSN field and retaining the original sequence for matching keys. This groups together all messages for a given job in composite log sequence, ordering jobs by their extended JSN, which also separates jobs by operating periods. Sort/Merge 5 is also used to perform this sorting as efficiently as possible. The sorted jobs log output from this phase should also be saved for possible SECART reruns of the job extraction phase or other processing, for example, accounting programs.

Job Selection

During the raw logs processing phase, SECART checks each new JSN, jobname, user identification, and terminal name found in log messages for job selection values specified via SECART directives. Matches are recorded in the JSN trace table for subsequent unit job extraction and printout. The printout of the JSN trace table at the end of the raw logs processing phase includes a field containing flag characters to identify those jobs selected for unit job printout and the attribute(s) which qualified each for selection. Similarly, when the JSN trace table is loaded from a file, each entry is checked against current SECART directives and matches are recorded in the JSN trace table for unit job extraction. This provides a capability for ad hoc retrieval of unit jobs from the sorted jobs log via SECART reruns with minimal processing requirements.

It should be noted, however, that job selections made in an initial SECART run test all occurrences of each job attribute as encountered in the raw logs, while selections made via reruns operate only on attributes recorded in the JSN trace table, which reflect only the first occurrence of an attribute associated with the job. Therefore, initial run job selections are more thorough than reruns and provide the capability of detecting transient attributes, for example, changing username where permitted.

Job Extraction Phase

The job extraction phase consists of selecting and printing jobs from the sorted jobs log based on job selection flags associated with the job via the JSN trace table retained in memory. Unit jobs are separated with blank lines for readability. To reduce the risk of unintentionally generating massive printouts, a limit on the number of messages to be printed for each job can be specified via SECART directives. An informative message is issued whenever this sample limit is reached. SECART terminates normally when it encounters the end of the sorted jobs log.

SECART Installation and Use

SECART may be installed on the operating system or maintained as a file stored in a library or username catalog accessible to the system security administrator(s). SECART also requires a message identifier database to define and describe account log message identifiers. This database contains a list of all defined account message identifiers, an optional flag for default printout, and a short description of the event associated with that message identifier. It is a small sequential text file which can be updated via a text editor. The list should be augmented to reflect any additional software products installed on the system which issue account log messages and the default flags set as needed. Since SECART and its message identifier database file are not needed by other users, storing it in a library or username catalog is likely to be more convenient.

SECART is intended to be used as an integral part of routine system dayfile log handling procedures and on an ad hoc basis to assist in the analysis of security-related events. A simple FORTRAN application, SECART does not have the capability to directly access the log data it is designed to process. NOS commands needed to dump and/or terminate dayfile and account logs and to produce permanent file copies of these logs must have been executed prior to executing SECART. Those commands may be issued either manually via the system console or as part of a job with the privileges required to obtain the necessary system log access.

SECART Parameters

The SECART program execution command accepts order-independent parameters to specify which files are to be used/created.

Format:

SECART (p_1=value$_1$,...,p_n=value$_n$)

Parameter	Description
A=aclog	Specifies the name of the input file containing raw account log messages. Omitting this parameter or entering A=0 specifies no file. Specifying only A defaults to ACLOG.
C=combo	Specifies the name of the file for the composite log with JSN prefix and source log annotations. This file is an output file for the initial SECART run; an input file for a rerun at the job sort phase when both raw log files are omitted. Omitting this parameter or entering C=0 specifies no file. Specifying only C defaults to COMBO.
D=dflog	Specifies the name of the input file containing raw dayfile log messages. Omitting this parameter or entering D=0 specifies no file. Specifying only D defaults to DFLOG.
I=input	Specifies the name of the file containing the SECART directives. Omitting this parameter or entering only I defaults to INPUT. I=0 specifies no directives.
J=jobs	Specifies the name of the file for the sorted unit jobs log. This file is an output file for the initial SECART run; an input file for a rerun at the job extraction phase when both raw logs and the serial log are omitted. A matching trace table file is required for reruns. Omitting this parameter or entering J=0 specifies no file. Specifying only J defaults to JOBS.
L=output	Specifies the name of the output file required for printouts. Omitting this parameter or entering only L defaults to OUTPUT. L=0 is not permitted.
M=msgid	Specifies the name of the file containing the account log message identifier database. Omitting this parameter or entering M=0 specifies no file. Specifying only M defaults to MSGID.
T=trace	Specifies the name of the JSN trace table file produced by a previous SECART run or created by this run. Since this file will be used for both input and output, the use of an indirect access file or other form of file copy is recommended. Omitting this parameter or entering T=0 specifies no file. Specifying only T defaults to TRACE.

SECART Directives

Each directive must be on a separate input line, in keyword=arglist form, may be in any order, and repeated as needed. Where applicable, multiple arguments (arg,...) may be specified on a single directive. Internal table sizes limit arguments to a maximum of 50 of each type. Directives accepted by SECART include:

Directive	Description
AMSG=msid,msid,...	Specifies the four-character account log message identifiers to be selected for printout during the first processing phase.
DMSG=nn: textstring	Specifies an undelimited character string to select dayfile messages for serial printout during the first processing phase. If nn: is specified, the text string must begin at this position in the message field which corresponds to column numbers for system commands.

JSN=d.jsno,d.jsno,...
or
JSN=jsno,jsno,...
	Specifies one or more job sequence names to be used for selection of unit jobs for printout. The optional prefix d. is the deadstart-unique code letter assigned by SECART during the raw logs processing phase to resolve JSN ambiguities across initial deadstarts.

LIMIT=nnnnn	Specifies a limit for the number of log messages to be printed for each selected job. A message will be printed when this limit has been reached. Maximum limit is 99999.

TIME=hhmmss-hhmmss
or
TIME=hh.mm.ss.-hh.mm.ss.
	Specifies a time interval in which all messages from both logs are to be included in the composite printout from the raw logs processing phase. Separating periods and the second time argument are optional. If omitted, the second argument will default to the same value as the first argument, which selects a one-second interval.

TRACE=key,key,...	Specifies JSN trace table sort keys. Each key must be one of the job attribute names as labelled in the JSN trace table printout. Valid keys are: UJN, JSN, SDT, EDT, FM, UN, and TRM. The sort sequence will be ascending per the display collation sequence.
TRM=termid,termid,...	Specifies one or more terminal identifiers as defined in the CCP network configuration file or the CDCNET system configuration file and printed in the job trace table.

Directive	Description
UJN=jobname,jobname,...	Specifies one or more user jobnames to be used as job printout selection parameters. Note that a job's UJN is recorded only in the first dayfile message, if any.
USER=family,username or USER=*,username or family,*	Specifies the user identification parameters to select jobs for printout. A blank or omitted family specifies the system default family. An asterisk (*) may be specified for either family or username to select all values of that parameter. A user index may be specified for the username and will be recognized if it is used in an SUI command.

Index

Index

Comments (continued from other side)

'lease fold on dotted line;
eal edges with tape only.

FOLD

OLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

CONTROL DATA
Technology & Publications Division
ARH219
4201 N. Lexington Avenue
Arden Hills, MN 55126-6198

NOS Version 2 Security Administrator's Handbook 60460410 B

We value your comments on this manual. While writing it, we made some assumptions about who would use it and how it would be used. Your comments will help us improve this manual. Please take a few minutes to reply.

Who are you?

□ Manager
□ Systems analyst or programmer
□ Applications programmer
□ Operator
□ Other ______________________________

How do you use this manual?

□ As an overview
□ To learn the product or system
□ For comprehensive reference
□ For quick look-up

What programming languages do you use? ______________________________

How do you like this manual? Check those questions that apply.

Yes	Somewhat	No	
□	□	□	Is the manual easy to read (print size, page layout, and so on)?
□	□	□	Is it easy to understand?
□	□	□	Does it tell you what you need to know about the topic?
□	□	□	Is the order of topics logical?
□	□	□	Are there enough examples?
□	□	□	Are the examples helpful? (□ Too simple? □ Too complex?)
□	□	□	Is the technical information accurate?
□	□	□	Can you easily find what you want?
□	□	□	Do the illustrations help you?

Comments? If applicable, note page and paragraph. Use other side if needed.

Would you like a reply? □ Yes □ No

From:

Name ______________________________ Company ______________________________

Address ______________________________ Date ______________________________

______________________________ Phone ______________________________

____

Please send program listing and output if applicable to your comment.

CONTROL DATA