GUIDELINES FOR THE DEVELOPMENT OF APPLICATION SECURITY

CONTENTS

PAGE

1. OVERVIEW
   - A. Information Protection Concepts 2
   - B. SW912 Terminology 4
   - C. SW912 Administration 4

2. PURPOSE AND SCOPE
   - A. Security Training 5
   - B. Application Programming 5
   - C. Reviewing Systems for Security 5
   - D. Documenting Controls 6

3. SECURITY RESPONSIBILITIES
   - A. Project Manager 6
   - B. Project Leader 7
   - C. System and Security Administrators 8
   - D. Client Department 8
   - E. Computer Security Administration Group (CSAG) 8
   - F. Interdepartmental Security Forum (ISF) 8

4. SECURITY REQUIREMENTS
   - A. SWBT Security Strategies 9
   - B. Security Control Processes 9
   - C. Access Control - General Requirements 11
   - D. Identification - Userids 12
   - E. Authentication - Passwords/Tokens/Biometrics 13
   - F. Authorization - Access to Resources 14
   - G. Monitoring and Audit Mechanisms 16

5. APPLICATION DEVELOPMENT PROCEDURES
   - A. Life Cycle Concerns 16
   - B. Determining System Risk 17
   - C. Designing Controls 18
   - D. Testing Controls 20
   - E. Implementing a System 21
   - F. Documenting Security Features 21

6. CONTINUITY OF SERVICE
   - A. Preparing for a Recovery 22
   - B. Recovery Procedures 22

1. OVERVIEW

A. Information Protection Concepts

1.01 IMPORTANCE OF ELECTRONIC INFORMATION: Electronic Information (EI) is defined in Operating Practice No. 113 (OP113), Protection of Electronic Information, as any data in an electronic format. Southwestern Bell Telephone Company (SWBT) relies heavily on EI processing to meet operational, financial, and informational requirements. Consequently, the continued integrity, availability, and confidentiality of software systems, databases, and data networks are critical to the Company. Corruption, unauthorized disclosure, or theft of EI could have a disruptive effect on the business operations of SWBT, cause financial and legal ramifications, and lead to the loss of public confidence.

1.02 AUDIENCES FOR SECURITY DOCUMENTATION: OP113 is written to provide a broad overview of Information Security concepts and policies for SWBT. The various Company practices which begin with the designation "SW" provide specific information for selected groups in SWBT. This practice, SW 007-590-912 (SW912), Guidelines for the Development of Application Security, applies to ALL individuals in SWBT doing application development on any platform as well as those performing acceptance testing and analysis on centrally developed or vendor supplied software. Section 1.07 describes some of the related Company documentation for specific areas of security, system operations, and user groups.

1.03 PERSONAL ACCOUNTABILITY: SW912 describes the specific requirements to protect electronic information using software programming. In accordance with the Code of Business Conduct and OP113, Section 2.3:

> ***"Fulfilling protection of electronic information responsibilities is MANDATORY and a condition of continued employment."***

1.04 SECURITY FEATURES AND MECHANISMS: The process of providing the software, from beginning to end, must take security considerations into account. SW912 consists of both security feature requirements (i.e., those for software) and development cycle security requirements (i.e., those which support the process of software production and maintenance). Features and mechanisms must be properly conceived, designed, implemented, tested, installed, documented, and maintained or a false sense of security is achieved.

1.05 DESIGNING IN SECURITY: SWBT's experiences have shown that application security is extremely difficult and expensive to add after an application is in operational use. To minimize the development cost, ensure standards are met, and properly train all users from the beginning, the elements of application

security ***MUST*** be included in the initial design, testing, and documentation effort for all systems.

1.06 REQUIREMENTS BY PLATFORM: Individual requirements by hardware/software platforms ***MUST*** be in accordance with this document. These requirements must be documented and maintained by the appropriate programming and/or development groups and be incorporated into training programs (see OP113 for associated documentation and training available for various platforms). In order to standardize security software and minimize programming and support costs, application groups ***SHOULD*** use a SWBT Standard Operating Environment (SOE) approved software package for basic security services (e.g., MVS applications should use RACF, etc.). Platform standards groups ***MUST*** document these packages and provide instructions on their use.

1.07 RELATED DOCUMENTATION: The concepts in SW912 are in accordance with OP113 and SWBT's Strategic Systems Architecture (SSA) Application Developer Guidelines (see Appendix 2). Related concepts are discussed in Appendix 1, DOD Trusted Computer System Evaluation Criteria ("The Orange Book"), and under development in Bellcore's Minimum Security Functional Requirements (MSFR). SW912 is based on Bellcore's generic security document, Bellcore Operations Systems Security Requirements (BOSSR). Functionally, OP113 is SWBT's "umbrella" practice for Electronic Information Security with other Company documents providing specific procedures by security area: physical, hardware/software platforms, departmental operations, and contingency planning. Some examples are:

- SW 007-590-904 (SW904), Off-Premise Storage;
- SW 007-590-905 (SW905), Computer Facility Physical Security;
- SW 007-590-906 (SW906), Disaster Recovery/Computer Facility Contingency Planning;
- SW 007-590-907 (SW907), Computer Security-Mainframes;
- SW 007-590-908 (SW908), Computer Security-UNIX Platforms;
- SW 007-590-909 (SW909), Documenting Minicomputer Recovery Plans;
- SW 007-590-910 (SW910), Computer Security-Networks (under development for issuance in 1993);

- SW 007-590-911 (SW911), Microcomputer Security Procedures Manual; and

- SW 007-590-913 (SW913), Computer Security-Central Office Switches (to be issued in 1993).

B. SW912 Terminology

1.08 DEFINITIONS IN SW912: Definitions are not intended to be exact technical descriptions, but rather are general descriptions for the use of SWBT employees in implementing information security mechanisms. When first used, significant items will be defined and also cross-referenced in Appendix 4, Information Security Index.

1.09 BASIC DEFINITIONS: The following are definitions of basic concepts used in this document.

a. DATA: In accordance with OP113, any information in an electronic format whether stored, processed, or transmitted.

b. GUIDELINES: Rules which are highly recommended. Guidelines may be recognized by the use of the words ***SHOULD*** or ***MAY***.

c. STANDARDS: Rules which ***MUST*** be followed. Standards may be recognized by the use of the word ***MUST*** and ***SHALL***.

d. SYSTEM: For the purposes of this document, "system" will be used in the place of the expression "computer system and/or application."

C. SW912 Administration

1.10 SW912 MAINTENANCE: SW912 is maintained by Information Services-Support (CSAG), see Section 1.11 for CSAG contact information. This is the initial release of this document (in the future, this section will include a brief overview of previous and current releases).

1.11 SW912 QUESTIONS/COMMENTS: Questions or comments on this practice should be referred to the Computer Security Administration Group (CSAG). Contact CSAG through the SWBT GHQ directory (blue pages) under Computer Security Administration or via E-mail to user "csag" on the "isoa" system.

2. PURPOSE AND SCOPE

A. Security Training

2.01 TRAINING COURSES: Initial training courses for project managers and programming staff ***MUST*** include an overview of the security concepts and concerns documented in SW912. Information Services' Computer Security Administration Group (CSAG) coordinates and reviews the content of the security elements of these courses.

B. Application Programming

2.02 DEVELOPING SECURE SOFTWARE: The SW912 guidelines are designed to provide an overview of how software developers and programmers should prevent and detect unauthorized use, disclosure, modification, or destruction of application software and data. These guidelines provide a set of requirements that specify both secure procedures during development and security features for live processing. However, these requirements ***MUST NOT*** be used blindly as the absolute and complete security requirements for a specific application. Rather, they are a template to which additional security features ***MAY*** need to be added based on the specific nature of an application's data and its criticality to SWBT operations.

2.03 PROCESSING PLATFORM: These guidelines apply to applications on ALL processing platforms and in any phase of the development cycle. In some environments, the network or operating environment supporting the application ***MAY*** be used to provide selected security features.

2.04 SYSTEM DEVELOPMENT: As a system is developed (or during the maintenance change cycle), SW912 guidelines ***MUST*** be reviewed for applicability by the project manager and the development team (Section 3 describes the responsibilities for developing secure software). Required controls ***MUST*** be implemented in accordance with the specific procedures within the software/hardware environment. These controls ***MUST*** be tested during the testing phase of development. If a control is not available in a specific environment, alternative procedures ***MUST*** be developed and documented by the development team to provide adequate security.

C. Reviewing Systems for Security

2.05 APPLICATION SECURITY CHECKLIST: Appendix 3, Application Security Checklist, is a generic checklist which summarizes the security controls which should be available. This checklist, or an equivalent one unique to the platform environment, ***MUST*** guide the selection, testing, review, and implementation of

controls. Bellcore uses a similar document, "Self-help Guide and Workbook for Producing BOSSR Security Analyses," in its reviews (called ISPRs) of their centrally developed Operating Support Systems (OSS).

2.06 SECURITY REVIEWS: In accordance with OP113, operating system environment and application controls ***MUST*** be evaluated against SW912 and specific platform guidelines during design phase, before initial implementation, and when there are relevant changes to the software, operating environment, or procedures. The completed reviews ***MUST*** be kept with other project documentation. Control deficiencies ***MUST*** be documented and reviewed for implementation as soon as possible during the maintenance cycle.

2.07 SYSTEM AUDITS: The Internal Audit Staff will use SW912, the associated platform procedures, and the completed security reviews when auditing applications and the development process.

D. Documenting Controls

2.08 SOFTWARE DOCUMENTATION: Security controls utilized in new or in existing applications ***MUST*** be included as part of the software documentation. The personnel responsible for centrally developed systems ***MUST*** coordinate with the appropriate central developers to ensure that adequate system controls exist and that the controls are properly documented. If the controls are not properly documented by the central developers, it is the responsibility of the SWBT development team (i.e., project manager and leader) to prepare the appropriate documentation.

3. SECURITY RESPONSIBILITIES

A. Project Manager

3.01 PROJECT MANAGER FUNCTIONS: The Project Manager is the SWBT employee who represents the users and sponsoring organizations as an application is designed, developed, and modified. Generally, this person has overall responsibility for funding, training, operations, and implementation decisions.

3.02 DETERMINING THE LEVEL OF CONTROLS: The Project Manager and the development team (e.g., Project Leader, clients, etc.) are jointly responsible for performing a risk analysis to determine the level of controls which are required (see Section 5.03). Guidance in assessing the sensitivity and vulnerability of system data and implementing controls may be obtained from Information Services' CSAG group (see Section 1.11). As required, CSAG will draw additional assistance from Legal, Asset Protection, and other departments.

3.03 DOCUMENTING CONTROLS: The Project Manager is responsible for ensuring that the appropriate application documentation is completed. In general, user documentation and compliance methodology will be completed by the Project Manager or client department with computer processing documentation completed by the Project Leader. However, even when the direct responsibility for documenting selected control points exists in another group, the Project Manager ***MUST*** still ensure all items are completed.

3.04 REVIEWING CONTROLS

a. OP113 REVIEWS: In accordance with OP113, security controls must be reviewed regularly. SW912 describes the reviews associated with software development. Once a system is operational, a different review is made annually by operations personnel and project management using the checklists described in either OP113 or the specialized platform SW practices (e.g., SW907 for MVS/RACF platforms, etc.). All reviews ***MUST*** cover the appropriateness, completeness, and effectiveness of the controls.

b. SW912 REVIEWS: An SW912 review ***MUST*** be made of the operating system and application security controls which have been identified as a result of the risk analysis (see Section 5.03) during design phase, before initial implementation, and when there are security relevant program or operating environment changes. The operating environment controls ***MUST*** be included in the review since they may be used to supplement the system's (e.g., RACF for MVS applications, etc.). Either Appendix 3 or an equivalent checklist ***MUST*** be used to document these reviews. Reviews ***MUST*** be retained with the project's documentation, reviewed with the departmental ISF member (see Section 3.12), and any remaining deviations reported to the Chairperson of the ISF as indicated in Section 4.10.

B. Project Leader

3.05 PROJECT LEADER FUNCTION: A Project Leader is the SWBT or contract employee who is responsible for developing and testing software. In Information Services this person is usually a programmer who works under the direction of the Project Manager. In some departments, this person may even be the Project Manager and/or operate the resulting system.

3.06 IMPLEMENTING TECHNICAL CONTROLS: The Project Leader is responsible for translating the levels of controls into specific coding techniques and procedures depending on the hardware/software platform in use. As improved technologies become available, they ***SHOULD*** be included in the programming maintenance and development cycles.

3.07 REVIEWING SECURITY: The Project Leader ***MUST*** work closely with the Project Manager to ensure that a review is made of the program security controls during development and whenever significant changes are made to the program or the operating environment. Appendix 3 or an equivalent checklist ***MUST*** be used for this purpose in accordance with Section 3.04b.

C. System and Security Administrators

3.08 SYSTEM OPERATIONS: In most departments, a System Administrator provides daily operational support for computer systems. This individual can be in the user department, as with many departmental systems, or in a central computer facility as in Information Services. Specific responsibilities are documented in OP113 and other SW practices (see Section 1.07).

3.09 SECURITY OPERATIONS: The security elements of daily operations are the responsibility of the Security Administrator. This person may be responsible for the security of multiple systems (e.g., a RACF administrator, multiple UNIX systems, etc.) or a single system, in which case he may also be the System Administrator (see Section 3.08).

D. Client Department

3.10 COMPLIANCE METHODOLOGY: With the assistance of the ISF, operating departments are responsible for implementing and adhering to effective security compliance programs. These programs ***MUST*** ensure that appropriate user and operations controls are in place and being used. Project Managers are responsible for developing and documenting the appropriate review and compliance procedures for the various controls. Project Leaders should mechanize this process whenever feasible and cost effective.

E. Computer Security Administration Group (CSAG)

3.11 SW912 ADMINISTRATIVE SUPPORT: Information Services-Support (CSAG) is responsible for the support and administration of these guidelines (see Section 1.11 for CSAG contact procedures).

F. Interdepartmental Security Forum (ISF)

3.12 SWBT SECURITY POLICIES: The ISF is composed of security representatives from all sixth level groups and meets monthly to evaluate and recommend Company policies on Information Security. The ISF also reviews current control deviations, departmental procedures, and security alerts from Bellcore. Contact the ISF as indicated in the SWBT GHQ directory (blue pages) under Interdepartmental Security Forum or by E-mail under user "isf" on system "isoa."

4. SECURITY REQUIREMENTS

A. SWBT Security Strategies

4.01 STRATEGIC SYSTEMS ARCHITECTURE (SSA): A new document, Strategic Systems Architecture (SSA) Developer Guidelines, is being developed by the Strategic Planning District at Technology Resources (SBC-TRI) during 1992/93. Application developers and managers ***MUST*** use the security guidelines and requirements described in this new document as the criteria for developing or acquiring new applications. Developers ***MUST*** review at least the sections on Contract Authorization and Module-to-Module Authorization. Appendix 2, provides an overview of SSA security considerations.

4.02 SOURCES OF SECURITY: The appropriate security requirements specified in SW912 ***MAY*** be supplied by the underlying operating environment or, if ***NOT***, ***MUST*** be implemented in the application. The security requirements presented in this document do ***NOT*** take into account the unique security risks and vulnerabilities of any specific application. They are meant to be used as input to the requirements phase and successively refined to address an application's unique risks. This process will entail refinement of some requirements with application specific details, and, with proper approval, elimination of others due to limitations of the operating environment. When a specific control cannot be implemented, alternative manual procedures ***MAY*** need to be documented and used to maintain appropriate security levels.

4.03 BASIC SECURITY FEATURES: The basic security features any software (by combining the application and operating system) ***MUST*** provide are identification, authentication, authorization/access control, data integrity, user accountability via an auditing mechanism, and continuity of service. These features, as discussed in later sections, will appear to some degree in any secure application, independent of the corporate policy or the specific services provided.

B. Security Control Processes

4.04 DEVELOPMENT TEAM: The SW912 guidelines are intended to be used by all members of the development team. This includes project managers, developers, auditors, standards organizations, and the users. The guidelines include concepts and controls to assist team members in defining control requirements and to assist security administrators in administering these controls. The control guidelines ***MUST*** be used by the individuals responsible for the approval or review of the design as well as the development and implementation of a system. In addition, pertinent security features ***MUST*** be considered in the funding and approval process as documented in Operating Practice No. 71 (OP71), Information Systems.

4.05 GENERIC CONTROLS: System designers and other development team members should select the appropriate SW912 concepts and procedures to be used in their systems. In most cases, the techniques described in SW912 represent typical controls, or examples of various classes of controls. Rarely will it be possible to incorporate them directly into a system without some work in customizing the controls to the peculiarities and circumstances of an actual platform environment.

4.06 SPECIFIC CONTROLS: Using the concepts in SW912, designers ***SHOULD*** prepare their own list of controls which satisfy SWBT security guidelines and are especially suited to their particular design areas (i.e., the individual hardware/software platform). This document is ***NOT*** intended to discourage the use of more appropriate controls and control approaches from other sources. Designers ***SHOULD*** improve upon the SW912 controls, or modify them, as needed. Appendix 3 or an equivalent checklist ***SHOULD*** be used to guide the selection of specific controls.

4.07 CONTROLS FROM THE PHYSICAL ENVIRONMENT: See SW905.

4.08 CONTROLS FROM THE ELECTRONIC ENVIRONMENT: While an application is responsible for ensuring an appropriate level of security, all controls do ***NOT*** have to be within a given application. Wherever possible, an application should make use of the basic security features available in the electronic environment (e.g. MicroLinkII's NUI process, RACF with MVS applications, etc.). This will minimize the cost of programming and supporting these controls. However, in more restrictive situations (e.g., unique permissions for selected users, etc.), additional controls may still be required within the application. Likewise, controls over the introduction of changed program versions should use existing change management procedures if possible.

4.09 VALIDATING THE ADEQUACY OF CONTROLS: Regardless of whether the controls are explicitly included in a system's design or whether they are part of the environment external to the system (either physical or electronic), the members of the development team are responsible for ensuring that adequate controls exist and function to protect the system. Application developers ***MUST*** determine that planned external controls are actually included in the external environment and satisfy themselves as to their adequacy. Normally, operational experience with these or similar controls should provide an indication of their adequacy. In any event, Appendix 3 or an equivalent checklist ***MUST*** be used before system implementation to verify compliance with SW912's security controls.

4.10 DOCUMENTING SECURITY DEVIATIONS: Deviation from the security guidelines requires the approval of the system's project management. It is the

responsibility of the Project Manager to review any deviations with the Computer Security Administration Group (CSAG) (see Section 1.11 for contact information). Potential deviations should be discussed with the department's ISF member or the ISF Chairperson (see Section 3.12). Serious deviations ***MUST*** be documented and justified in accordance with OP113, submitted for review to the ISF, and retained for audit purposes.

C. Access Control - General Requirements

4.11 COMPLEMENTARY FUNCTIONS: Identification, authentication, and authorization are separate but complementary functions. Identification (see Section 4.20) is the process by which a system recognizes and tracks a user using a specific access code or identifier (i.e., a userid). Authentication (see Section 4.25) is the process by which the user "proves" he has a right to use this userid (e.g., he knows a specific password, has a known "token," has a specific voice-print, etc.). Authorization is the process by which the system agrees to specific resources being accessed during an authenticated user's session. These three processes may not always be clearly distinct and may even be integrated with physical access controls. For example, a micro/personal computer (PC) may only require a single password to identify and authenticate the user while simultaneously authorizing complete access to all the PC's resources. The success of protection efforts is dependent on security administrators and users integrating these concepts in light of local conditions.

4.12 WARNING MESSAGE: A "no-trespassing" message ***MUST*** be displayed ***BEFORE*** the logon process begins to notify potential users of their responsibilities. The SWBT Legal department has approved the following message:

> ***"This is a Southwestern Bell Telephone Company system restricted to Company official business and subject to being monitored at any time. Anyone using this system expressly consents to such monitoring and to any evidence of unauthorized access, use, or modification being used for criminal prosecution."***

4.13 CONTROLLING ACCESS BY PERIOD OF TIME: Systems ***SHOULD*** have a mechanism to permit or limit access by a specified range of time (e.g., time-of-day, day-of-week, calendar day).

4.14 AUTHENTICATION IS REQUIRED: Access to resources ***MUST*** be based on authenticated userids (see Section 4.32).

4.15 RULE OF LEAST PRIVILEGE: In accordance with OP113, users ***SHOULD*** have ***ONLY*** the ***MINIMUM*** privileges necessary to access the information and tools needed to perform their jobs and nothing more.

4.16 DEFAULT ACCESS FOR NEW DATA: As a general rule, default access for new files or databases (i.e., permissions) ***MUST*** restrict access to the creator of the data. This is in accordance with the Rule of Least Privilege and requires ***EXPLICIT*** action by the creator in order for others to have access to the data (either as individuals or groups). For example, a system used exclusively for public information ***MAY*** choose to have more open access rules during live operations (e.g., permit read access by the general user, etc.).

4.17 ACCESS TO SENSITIVE DATA: The application should enforce proper separation of duties and provide dual controls in areas that are critical such as monetary disbursements and handling of highly sensitive information. In some situations, the Project Manager ***MAY*** decide to maintain the data in an encrypted form.

4.18 TERMINAL TIMEOUT: After a reasonable period of user inactivity (e.g., 15 minutes), a terminal screen ***MUST*** be blanked and either the keyboard locked-up (i.e., the user's password ***MUST*** be used to regain access) or the terminal ***MUST*** be disconnected (logged off). This mechanism ***SHOULD*** be available for user activation (e.g., via a "hot key," etc.).

4.19 TERMINATING SESSIONS: When a session is interrupted (e.g., users hangs-up or turns power off without normal logoff), the application or operating environment ***MUST*** terminate the session and prevent the same or another user from reconnecting without re-logging on.

D. Identification - Userids

4.20 USERID UNIQUENESS: To ensure individual accountability, a userid ***MUST*** be assigned to a SINGLE user. There ***MUST*** be a mechanism to associate individual information with each userid (e.g., user's name, organization, phone number, etc.). Since the userid is public information, it ***MAY*** be used for communication (e.g., through E-Mail, etc.) or in routing/distributing documents. An application ***MUST NOT*** be coded to require or encourage the sharing of userids (sometimes refereed to as a group userid).

4.21 USERID FORMATS: SWBT standard userid formats are documented in OP113, Section 6.2 (SWBT employee userids are displayed on the PHONE/ CORINET system). SWBT employees ***MUST*** use their standard userid on ALL SWBT systems. Non-SWBT employee, machine/application, and other special userids will be assigned and tracked by the system's security administrator. The System Userid Tracking System (SUITS) provides a mechanized file of all current SWBT employee userids (other types of userids will be added and centrally maintained when practicable). Questions on PHONE or SUITS can be referred to

the CORINET help desk (314 235-3412). Existing applications that do not use the standard formats ***MUST*** convert as soon as practical.

4.22 PRIVILEGED/SYSTEM USERID: If a privileged userid (e.g., "root" on UNIX systems, "superuser" on MVS/RACF systems, or an individual whose userid has been granted equivalent rights, etc.) is required to operate or administer the application, it ***MUST*** be distinct for that application and rigorously controlled.

4.23 TRACKING ACTIVE USERS: The application or operating environment ***MUST*** internally maintain the identity of all currently active users (i.e., logged on users cannot change or mask their identity).

4.24 VALIDATING USERIDS: The application or operating environment ***MUST*** have a method to list all userids that are permitted access. Security Administrators ***MUST*** regularly provide (e.g., quarterly) a list of users and their valid activities/permissions for revalidation by an appropriate SWBT district level or higher manager (this could be the user's manager).

E. Authentication - Passwords/Tokens/Biometrics

4.25 AUTHENTICATION METHODS: Sophisticated authentication methods based on tokens (e.g., smartcards), cryptography (e.g., Kerberos), and biometrics (e.g., voice recognition) are becoming more common. However, passwords are generally the most cost effective authentication mechanism for system access and will continue to be used on many systems.

4.26 PERFORM ENTIRE AUTHENTICATION PROCESS: The system ***MUST*** appear to perform the entire user authentication procedure even if the userid that was entered is ***NOT*** valid. ***NO*** error feedback will be given to indicate which part of the authentication information is incorrect.

4.27 INITIAL ENTRY/CHANGING PASSWORD: Before a password can be changed, the user ***MUST*** fully authenticate himself. When a password is initially assigned or it is necessary to administratively change it, a change-on-first-use password ***SHOULD*** be used which requires the user to immediately change it to a personal password upon the first access.

4.28 PASSWORD REQUIREMENTS: The following requirements provide the basis of secure mechanisms for password authentication and access control.

a. INDIVIDUAL PASSWORDS: Each userid ***MUST*** require an individual password or token authenticator. The application ***MUST NOT*** provide a mechanism whereby a single password can be used by multiple userids.

b. DISPLAYING PASSWORDS: Clear-text representation of the password on the data entry device ***MUST*** be suppressed or blotted out.

c. ACCESS TO PASSWORDS: Internal password files ***MUST*** have restricted access and, if possible, be encrypted.

d. PASSWORD FORMAT: Passwords ***SHOULD*** be 6-8 characters in length; ***NOT*** blank or a repeat of the userid; and contain at least one letter and one number/special character (at least one number/special character ***MUST*** be in a position ***OTHER*** than the first or last one).

4.29 PASSWORD AGING: The application or operating environment ***MUST*** provide a mechanism to age passwords, notify the user on-line of an expiring password, and allow the user to change his password on-line.

4.30 PASSWORD REUSE: The system ***MUST*** provide a mechanism to prevent the same user from reusing a password within a specified period of time (e.g., 6 months).

4.31 ADDITIONAL SECURITY: Critical applications with especially sensitive information may require more sophisticated password processors (e.g., one-time password devices, tokens, etc). These requirements ***SHOULD*** be developed during the Risk Analysis phase of system development (see Section 5.03).

F. Authorization - Access to Resources

4.32 INITIAL IDENTIFICATION: In general, an application ***MUST*** require users (e.g., individuals, remote systems/machines, etc.), to request access by using their assigned userid before any actions are permitted. Situations where this may not be appropriate (e.g., Mechanized Employee Locator, public "help" systems, etc.) ***MUST*** be carefully reviewed and monitored to ensure other Company systems are not at unreasonable risk.

4.33 AUTHORIZING ACCESS BY GROUP: For administrative efficiencies in granting access to system resources, a system ***MAY*** refer to a set of userids with a collective name when all userids have the same privileges or authorities. For example, a system's security manager assigns a set of files to the common identifier "OUR-DATA" and sets-up a collective name with the same label. Therefore, any userid which is a member of the group "OUR-DATA" is permitted access to all these files (this does ***NOT*** remove the need to continue to tract actual activities by individual userid).

4.34 PROTECTION OF ACCESS CONTROL FILES: The access control mechanism's data files and tables (e.g., password file in UNIX or MVS/RACF, etc.) ***MUST NOT*** allow access by unauthorized users.

4.35 ACCESS TO OPERATING SYSTEMS: Non-privileged users ***MUST NOT*** be allowed access to the underlying operating environment (e.g., "rc" file in UNIX, MVS system files in MVS, etc.).

4.36 SIMULTANEOUS USE OF THE SAME USERID: A single userid ***MUST NOT*** be permitted to simultaneously access a system(s) unless there is assurance that the ***SAME*** user is using the userid. If this cannot be done but there is a business need for simultaneous access, then real-time notification ***MUST*** be made to the involved users. The system ***MUST*** also require user acknowledgement of the notification (i.e., a hanging message which requires a user response, such as pressing the enter key) to ensure the userid is ***NOT*** being used by an unauthorized person.

4.37 BACKDOOR DOCUMENTATION: Programming backdoors into an application (i.e., hidden entry mechanisms, such as a special userid or password) is ***NOT*** a standard SWBT technique. When this must be done, then the backdoor ***MUST*** be fully documented and coordinated with the Project Manager.

4.38 INACTIVE USERIDS: A mechanism ***MUST*** be available to recognize, remove, or deactivate inactive userids (e.g., those which have no activity for 90 days).

G. Monitoring and Audit Mechanisms

4.39 TRACKING AND IDENTIFYING USERS: The userid associated with any system request or activity ***MUST*** be maintained and passed on to any other connected systems (or tracked) so that the initiating user action can be traceable for the lifetime of the request or activity.

4.40 AUDIT LOG: The application or its environment ***MUST*** generate an audit log that contains information sufficient for after-the-fact investigation of loss or impropriety, appropriate management response and personnel actions, and pursuit of legal remedies. The audit log must provide capabilities to analyze events with user accountability for all significant events (significant events ***SHOULD*** be defined during the requirements phase) and ***MUST*** be retained as a paper or mechanized copy for a reasonable period (e.g., 90 days).

4.41 ACCESS TO THE AUDIT LOG: The audit log control mechanisms and data ***MUST*** be protected from unauthorized access.

4.42 INVALID LOGON ATTEMPTS: After a third consecutive invalid userid/password combination is tried, the system ***MUST*** end the session (e.g., disconnect), log the event, and notify the Security Administrator if possible.

4.43 DISPLAYING LAST & INVALID LOGINS: ***AFTER*** successful authentication, each session ***SHOULD*** display the last logon date/time and the number of unsuccessful logon attempts. The user is responsible for reporting discrepancies to the appropriate Security Administrator.

5. APPLICATION DEVELOPMENT PROCEDURES

A. Life Cycle Concerns

5.01 MISCONCEPTIONS: A common mistake when addressing security concerns in software development is to focus solely on providing operations security mechanisms. Such mechanisms are only one element of providing a secure application. Security issues ***MUST*** also be considered during each phase of the development cycle. This section provides the baseline requirements for secure development cycle considerations.

5.02 SEPARATION OF DUTIES: In general, duties should be spread among many people to ensure there is an independent person performing each step of the

process. This improves the error trapping and reduces the possibility of fraud. Major areas of concern:

- DESIGN AND CODING: All phases of application design, the coding process, and the creation of test data to provide a complete representation of application data.

- OPERATIONS TEST: The trial, evaluation, and review of an application using test data.

- UPDATING CHANGES: Application changes are designed, coded, tested (including a field operations parallel, if necessary), and implemented.

- COMBINING DUTIES: When it is found necessary to combine some of these duties in a single person, special procedures ***MAY*** be necessary to ensure integrity and security of the application (e.g., a formal structured walk-through with peers and/or project management, another individual is assigned to review the system and its compliance with security requirements before implementation, etc.).

B. Determining System Risk

5.03 RISK ANALYSIS: A risk analysis ***MUST*** be performed and documented to determine what degree of security is required. The requirements must take into account how and where the system will be run, any legal or regulatory limitations on access to data, connectivity of the application, etc. A SWBT Risk Analysis procedure is under development and will be included in a future release of SW912 and formal Project Management training. The following areas ***SHOULD*** be considered during this process:

- SENSITIVITY OF INFORMATION: The first issue is determining whether electronic information requires special protection. It may be critical to Company operations (e.g., quantities and locations of equipment and facilities which ***MUST*** be protected from unauthorized modification, etc.), need to be protected from premature public disclosure (e.g., strategic plans, private legal or regulatory information, etc.), or be the private or personal records of our employees or customers (e.g., payroll, unlisted numbers, customer calling patterns, etc.). In addition, data files must be reviewed to determine if fields which individually are not sensitive may be a security exposure when available together.

- ACCESSING PERSONNEL: Protection features are influenced by who is being given access to a system. Clearly, systems which grant public access (e.g., Mechanized Employee Locator (MEL), 5-Call, etc.) ***MUST***

limit access to public information and ***NOT*** permit unauthorized access to other, interconnected systems.

- ACCESS METHODOLOGY (PHYSICAL): The physical environment where access is granted also influences the level of protection. A SWBT kiosk located in a shopping mall to permit customers to add calling features to their home phones has greater vulnerability than a terminal located behind a controlled-access door or protected by a guard service.

- ACCESS METHODOLOGY (ELECTRONIC): The electronic environment is also a factor. Dial-up lines and access through a public network may add more complex vulnerabilities than access procedures which use dedicated, SWBT lines. In addition, the operating system environment (e.g., MVS/RACF, UNIX, etc.) may provide standard security features which supplement or replace similar ones at the application level.

5.04 USERS & REQUIREMENTS: Based on the risk analysis, high-level security requirements ***MUST*** be determined and documented in collaboration with user organizations. The requirements must be determined by the sensitivity of the data, the connectivity of the application, and the intended services provided by the application. These high-level requirements ***MUST*** be translated into specific application features and procedures by the development group. The resulting security criteria ***MUST*** fulfill the requirements set forth in this document except where agreed to by CSAG and the client/user organizations.

5.05 SECURE TRANSMISSION OF DATA: When sensitive and/or private information is sent over a network that is not limited to point-to-point connections (e.g., utilizes public networks, etc.), consideration should be given to encrypting the information during transmission.

C. Designing Controls

5.06 SECURITY MECHANISMS: Determine security mechanisms that satisfy the functional security requirements defined during application design. The security mechanisms ***SHOULD*** be based on the target operating environment, internal software design, transaction types, input screen layout, data and file structures, specific entry points, etc.

5.07 PLATFORM TECHNIQUES/MECHANISMS: Specific programming techniques ***MUST*** be documented by the standards and/or programming groups responsible

for individual hardware and software platforms. These mechanisms should include but are ***NOT*** limited to:

- Detection of error conditions that might propagate throughout the application.
- Detection of communication errors above a user-specified threshold.
- Applications ***MUST*** be designed and developed to protect data integrity. These ***SHOULD*** include some or all of the following:
  - Proper rule checking on data updates
  - Proper handling of duplicate/multiple inputs
  - Checking of return status
  - Checking of intermediate results
  - Checking of inputs for reasonable values
  - Proper serialization of update transactions.

5.08 USING OPERATING ENVIRONMENT CONTROLS: Design into the application, to the extent possible, the use of standard security features made available by the underlying operating environment. Additional security controls can be included as required.

5.09 DESIGN REVIEWS: Consider both the functional security requirements and the potential for security flaws during all detailed design reviews.

5.10 SOFTWARE COPYING: Illegal and/or unauthorized copying of copyrighted software is a violation of the Code of Business Conduct and could lead to disciplinary action by the Company, civil litigation, and/or criminal prosecution. Unless authorized, employees or authorized Company agents ***MUST NOT***:

- copy software developed or purchased by the Company;
- use copied software; or
- use code-breaking devices which allow copy-protected software to be copied.

5.11 BYPASSING SOFTWARE SECURITY: Employees and authorized Company agents ***MUST NOT*** develop, copy, or use any program or code which circumvents or bypasses system security or privilege mechanisms or distorts accountability or audit mechanisms.

D. Testing Controls

5.12 FORMAL TEST PROCEDURE: Develop, document and implement a formal Project Manager/Client acceptance testing procedure in accordance with OP79.

5.13 LEVEL OF ANALYSIS: Ideally, there should be a different person for each step of the process (see Separation of Duties earlier). The amount of analysis of changes and test results depends on the independence of the people involved as well as the amount of code or changes that were made. A small change to a heading line of a report would require little analysis and a minimum amount of testing. A new program or major changes to an existing program would require extensive analysis and testing.

5.14 TESTING FEATURES: All security features ***MUST*** be tested. Tests must include a search for flaws that would allow violation of resource isolation, denial of service, or permit unauthorized access to any data element.

5.15 CORRECTING FLAWS: All flaws discovered during the testing process ***MUST*** be corrected, removed, or neutralized. The application ***MUST*** be retested to demonstrate that flaws have been eliminated and that no new flaws have been introduced.

5.16 TESTING ON PRODUCTION SYSTEMS: In general, development activities or testing with live data ***SHOULD NOT*** take place on a live production system. All new features, patches, or "quick fixes" ***SHOULD*** be tested on a development system or the approved test system. When this is ***NOT*** possible, the changes ***MUST*** be carefully coordinated and documented to allow timely recovery in case of unforeseen problems.

5.17 AUTHORIZING CHANGES: All new code and modifications to existing code ***MUST*** be authorized by the appropriate authority. All changes must be documented and reviewed to ascertain that security has not been compromised. Any changes to the functionality or defaults of security mechanisms in a new release ***MUST*** be documented and the documentation made available to the user prior to the implementation of that release.

E. Implementing a System

5.18 SECURITY CONTACTS: Specific members of the development/support team ***SHOULD*** be identified as the prime and backup contacts for system security issues.

5.19 VERIFYING SOFTWARE RELEASES: Procedures ***MUST*** exist to determine that the application software updates distributed to the processing site are exactly as specified by the master copies. Procedures (e.g., modification dates, permissions, checksums, etc.) must exist that make it possible to verify that the currently installed software has remained consistent with the delivered software, i.e., no unauthorized modifications have been made.

5.20 EMERGENCY CHANGES: In accordance with the Separation of Duties concept, programmers performing development or maintenance duties ***SHOULD NOT*** be custodians of the production versions or be allowed to modify the production versions except under controlled emergency conditions that are logged by operations personnel and approved by project management.

F. Documenting Security Features

5.21 DEFAULT CLASSIFICATIONS: In accordance with OP92, all documentation is considered ***PROPRIETARY*** and must be treated accordingly unless otherwise approved by the appropriate authority.

5.22 SEPARATE INSTRUCTIONS: Instructions and documentation on security considerations should be provided separately for Users, Security Administrators, and Operators.

5.23 SECURITY ADMINISTRATOR GUIDE: An administrator's guide should contain:

- Documentation on the use of all audit tools.
- The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit event.
- Recommendations for using file integrity review utilities on a regular basis.
- Procedures for periodic backup and deletion of audit logs.
- Procedures for checking the free disk space available for the log files.

- Administrator functions related to security, including adding or deleting a user, changing the security characteristics of a user, etc.

- Guidelines on the consistent and effective use of the protection features of the application and how they interact.

5.24 SYSTEM OPERATOR GUIDE: Operator documentation ***MUST*** fully describe the security features available to the operator and how they are to be used (e.g., special passwords, suspected unauthorized access, etc.).

5.25 CLIENT/END-USER PROCEDURES: The end-user documentation should describe the protection mechanisms, their purpose, and provide guidelines on their use. This documentation ***MUST NOT*** contain any information that could jeopardize the security of the application if it is made public. For example, ***NO*** actual passwords should be listed in the documentation.

6. CONTINUITY OF SERVICE

A. Preparing for a Recovery

6.01 LIMITS ON USER ACTIONS: Non-privileged user action, either deliberate or accidental, ***SHOULD NOT*** cause the application/system to be unavailable to other users, other than as specified by the system requirements.

6.02 APPLICATION BACK-UP: Documentation and procedures ***MUST*** be provided for application back-up and preparation for restoration in the event of a disaster or compromise of service.

6.03 PROTECTING MASTER SOFTWARE: A combination of technical, physical, and procedural safeguards ***MAY*** be required to protect the master copy or copies of all routines used to generate the application from unauthorized modification or destruction.

B. Recovery Procedures

6.04 APPLICATION RECOVERY: Procedures and/or mechanisms ***MUST*** be provided to allow recovery after a system failure or other interruption without a protection compromise. These procedures are documented in SW904 and SW906.

6.05 FACILITY OR NETWORK RECOVERY: Disaster recovery documentation should be coordinated with the associated facilities and networks documentation so that processing following an outage can be efficiently accomplished.