



KERNEL SECURITY TOOLS MANUAL

Version 8.0

July 1995

Revised February 2007

Department of Veterans Affairs
VistA Health Systems Design & Development (HSD&D)
Infrastructure and Security Services (ISS)

Revision History

Documentation Revisions

The following table displays the revision history for this document. Revisions to the documentation are based on patches and new versions released to the field.

Date	Revision	Description	Author
07/95	1.0	Initial Kernel V. 8.0 software and documentation release	REDACTED and Kernel Development Team, San Francisco, CA Office of Information Field Office (OIFO)
02/01/05	2.0	<p>Reformatted this document to follow the latest ISS styles and guidelines. No other content updates have been made in regards to released patches at this time.</p> <p>Reviewed document and edited for the "Data Scrubbing" and the "PDF 508 Compliance" projects.</p> <p>Data Scrubbing—Changed all patient/user TEST data to conform to HSD&D standards and conventions as indicated below:</p> <ul style="list-style-type: none">• The first three digits (prefix) of any Social Security Numbers (SSN) start with "000" or "666."• Patient or user names are formatted as follows: KRNPATIENT,[N] or KRNUSER,[N] respectively, where the N is a number written out and incremented with each new entry (e.g., KRNPATIENT, ONE, KRNPATIENT, TWO, etc.).• Other personal demographic-related data (e.g., addresses, phones, IP addresses, etc.) were also changed to be generic. <p>PDF 508 Compliance—The final PDF document was recreated and now supports the minimum requirements to be 508 compliant (i.e., accessibility tags, language selection, alternate text for all images/icons, fully functional Web links, successfully passed Adobe Acrobat Quick Check).</p>	REDACTED

Revision History

08/03/06	2.1	Updates: <ul style="list-style-type: none">• Reformatted document to the latest ISS SOP Guidelines.	Oakland, CA OIFO: <ul style="list-style-type: none">• Project Manager: REDACTED• Developers: REDACTED• Technical Writer: REDACTED
02/12/07	2.2	Changed Kernel document title references to: <ul style="list-style-type: none">• <i>Kernel Developer's Guide</i> (previously known as the <i>Kernel Programmer Manual</i>).• <i>Kernel Systems Management Guide</i> (previously known as the <i>Kernel Systems Manual</i>).	Oakland, CA OIFO: <ul style="list-style-type: none">• Project Manager: REDACTED• Developers: REDACTED• Technical Writers: REDACTED & REDACTED

Table i. Documentation revision history

Patch Revisions

For the current patch history related to this software, please refer to the Patch Module on FORUM.

Contents

Revision History	iii
Figures and Tables	vii
Orientation	xi
1. Introduction	1-1
Overview	1-1
Software Management.....	1-2
2. User Security.....	2-1
Kernel Security During User Sessions	2-1
Device Check.....	2-1
User Identification	2-1
Menus and Options	2-2
Kernel Security Codes.....	2-4
Access/Verify Codes.....	2-4
Electronic Signatures	2-6
Tips and General Advice for Users	2-8
Reviewing Users.....	2-8
General Information About Users.....	2-9
User's Access to VA FileMan Files	2-11
3. Menu Manager Security	3-1
Examining Menus and Options	3-1
Secure Menu Delegation	3-4
4. Kernel Audit Features.....	4-1
IRM's Responsibility	4-1
Initiating Audits.....	4-2
System Access Audits	4-2
Old Access and Verify Codes	4-2
Sign-on Log	4-3
Failed Access Attempts.....	4-4
Option and Server Usage Audits	4-9

Programmer Mode Log.....	4-9
Option Audit	4-10
Server Audit.....	4-15
VA FileMan Audits	4-19
5. Software Integrity.....	5-1
Program Integrity Checker Option	5-2
Verify Program Integrity Option.....	5-3
Checking Programs Received via Network Mail/PackMan	5-4
Checking Secured Programs Received via Network Mail/PackMan	5-7
Glossary	Glossary-1
Appendix A—VistA Security Forms.....	A-1
Index	Index-1

Figures and Tables

Table i. Documentation revision history	iv
Table ii. Documentation symbol descriptions.....	xii
Table 2-1. Security access types	2-2
Figure 2-1. Entering a <i>valid</i> Access/Verify code pair—Sample user dialogue	2-5
Figure 2-2. Entering an <i>invalid</i> Access/Verify code pair—Sample user dialogue	2-6
Figure 2-3. List users option—Sample user dialogue.....	2-9
Figure 2-4. List users option—Sample report.....	2-9
Figure 2-5. User Inquiry option—Sample user dialogue and report.....	2-10
Figure 2-6. User Status Report option—Sample user dialogue and report.....	2-11
Figure 2-7. Find a user option—Sample user dialogue.....	2-11
Figure 2-8. Inquiry to a User's File Access option—Example of a user with access to two files.....	2-13
Figure 2-9. Inquiry to a User's File Access option—Sample report of a user with access to two files ...	2-13
Figure 2-10. Inquiry to a User's File option—Example of a user with programmer access to all files ...	2-13
Figure 2-11. Inquiry to a User's File option—Sample report of a user with programmer access to all files.....	2-13
Figure 2-12. List Access to Files by File number option—Sample user dialogue.....	2-14
Figure 2-13. List Access to Files by File number option—Sample report	2-14
Figure 2-14. Print Users Files option—Listing file access for multiple users, sample user dialogue	2-14
Figure 2-15. Print Users Files option—Sample report.....	2-15
Figure 3-1. Option Access By User option—Sample user dialogue.....	3-1
Figure 3-2. Option Access By User option—Sample report.....	3-2
Figure 3-3. Inquire option—Sample user dialogue.....	3-2
Figure 3-4. Inquire option—Sample report.....	3-2
Figure 3-5. Print Options File option—Sample user dialogue.....	3-3
Figure 3-6. Print Options File option—Sample report.....	3-3
Figure 3-7. Show a Delegate's Options option—Sample user dialogue	3-4
Figure 3-8. Show a Delegate's Options option—Sample report	3-4
Figure 3-9. List Delegated Options and their Users option—Sample user dialogue	3-4
Figure 3-10. List Delegated Options and their Users option—Sample report.....	3-4
Figure 3-11. Print All Delegates and their Options option—Sample user dialogue	3-5
Figure 3-12. Print All Delegates and their Options option—Sample report.....	3-5
Figure 4-1. Purge Log of Old Access and Verify Codes option—Sample user dialogue.....	4-3

Figure 4-2. Print Sign-on Log option—Sample user dialogue	4-3
Figure 4-3. Print Sign-on Log option—Sample report	4-4
Figure 4-4. Establish System Audit Parameters option—Sample user dialogue setting audit parameters	4-5
Figure 4-5. Display the Kernel Audit Parameters option—Sample user dialogue and report	4-6
Figure 4-6. Failed Access Attempts Log option—Sample user dialogue	4-6
Figure 4-7. Failed Access Attempts Log option—Sample report	4-7
Figure 4-8. Device Failed Access Attempts option—Sample user dialogue	4-7
Figure 4-9. Device Failed Access Attempts option—Sample report	4-7
Figure 4-10. User Failed Access Attempts option—Sample user dialogue	4-8
Figure 4-11. User Failed Access Attempts option—Sample report	4-8
Figure 4-12. Failed Access Attempt Log Purge option—Sample user dialogue	4-8
Figure 4-13. Display of Programmer Mode Entry List option—Sample user dialogue	4-9
Figure 4-14. Display of Programmer Mode Entry List option—Sample user dialogue	4-10
Figure 4-15. Programmer Mode Entry Log Purge option—Sample user dialogue	4-10
Figure 4-16. Set Parameters for Audit	4-12
Figure 4-17. Display the Kernel Audit Parameters option—Sample user dialogue	4-13
Figure 4-18. Display the Kernel Audit Parameters option—Sample report	4-13
Figure 4-19. Audited Options Log option—Sample user dialogue	4-13
Figure 4-20. Audited Options Log option—Sample report	4-14
Figure 4-21. Option Audit Display option—Sample user dialogue	4-14
Figure 4-22. Option Audit Display option—Sample report	4-14
Figure 4-23. User Audit Display option—Sample user dialogue	4-14
Figure 4-24. User Audit Display option—Sample report	4-15
Figure 4-25. Audited Options Purge option—Sample user dialogue	4-15
Figure 4-26. Adding the Postmaster and the XQSRV namespace to the list of audit parameters to monitor servers	4-16
Figure 4-27. Display the Kernel Audit Parameters option—Sample user dialogue	4-17
Figure 4-28. Display the Kernel Audit Parameters option—Sample report	4-17
Figure 4-29. Server Audit Display option—Sample user dialogue	4-17
Figure 4-30. Server Audit Display option—Sample report	4-18
Figure 4-31. Audited Options Purge option—Sample user dialogue	4-18
Figure 5-1. Program Integrity Checker option—Sample user dialogue and report	5-2
Figure 5-2. Verify Package Integrity option—Sample user dialogue and report	5-3
Figure 5-3. Sample mailbox entries, including a sample PackMan message	5-4

Figure 5-4. Sample PackMan message contents	5-5
Figure 5-5. Activating PackMan.....	5-5
Figure 5-6. Comparing PackMan message contents with programs on a Vista system	5-6
Figure 5-7. PackMan and program comparison report	5-6
Figure 5-8. Sample mailbox entries, including a sample encrypted PackMan message.....	5-7
Figure 5-9. Sample encrypted PackMan message contents	5-8
Figure 5-10. Activating PackMan.....	5-8
Figure 5-11. PackMan and program comparison report	5-9
Figure A-1. User Account Notification message	A-1
Figure A-2. Computer Account Access Policy message	A-2

Orientation

The purpose of this manual is to provide instructions for Information Security Officers (ISOs) to review the Veterans Health Information Systems and Technology Architecture (VistA) system. Material is presented with the assumption that the reader has access to a VA VistA computing environment with ISO access to Kernel. It is assumed that the reader may not be familiar with Kernel as a whole but has a basic working knowledge of VA FileMan.

The intent of this manual is to provide a guide to the use of the security features supported by Kernel (i.e., Security Features User's Guide). In essence, this manual, along with the *Kernel Technical Manual* can serve your Information Resource Management (IRM) Service and Information Security Officer (ISO) as a Trusted Facility Manual.

This manual focuses on those security features which make up the System Security options of Kernel. Specific procedural instructions are restricted to examples. Users are expected to use the options described in the manual to enhance the ADP Security operations for their facility. There is no attempt to cover all possibilities that each option may offer. Rather, examples are shown which illustrate the potential use of Kernel's security features.

Specific details regarding the precise steps Kernel executes to provide system security are not discussed. The actual locations on your VistA computer system where Kernel records security-related data are not covered.



REF: To review such information, please refer to the *Kernel Systems Management Guide*.

The reader is encouraged to work closely with the Information Resource Management (IRM) Service to gain an understanding of the parameters and layout of the facility computer system before exercising the options. The more familiar you are with your own computer system, the more information you will gain from the reports produced by these security options.

The reader is also encouraged to become familiar with the security features of other VistA software. Many VistA applications include security features which can assist the Information Security Officer in monitoring system security.

How to Use this Manual

Throughout this manual, advice and instruction are offered about security-related information that Kernel V. 8.0 provides for overall Veterans Health Information Systems and Technology Architecture (VistA) application developers.



CAUTION: To protect the security of VistA systems, distribution of this software for use on any other computer system by VistA sites is prohibited. All requests for copies of Kernel for non-VistA use should be referred to the VistA site's local Office of Information Field Office (OIFO).

Otherwise, there are no special legal requirements involved in the use of Kernel.

This manual uses several methods to highlight different aspects of the material:

- Various symbols are used throughout the documentation to alert the reader to special information. The following table gives a description of each of these symbols:



Symbol	Description
	NOTE/REF: Used to inform the reader of general information including references to additional reading material.
	CAUTION: Used to caution the reader to take special notice of critical information.

Table ii. Documentation symbol descriptions

- Descriptive text is presented in a proportional font (as represented by this font).
- Conventions for displaying TEST data in this document are as follows:
 - The first three digits (prefix) of any Social Security Numbers (SSN) will begin with either "000" or "666".
 - Patient and user names will be formatted as follows: [Application Name]PATIENT,[N] and [Application Name]USER,[N] respectively, where "Application Name" is defined in the Approved Application Abbreviations document and "N" represents the first name as a number spelled out and incremented with each new entry. For example, in Kernel (KRN) test patient and user names would be documented as follows: KRNPATIENT,ONE; KRNPATIENT,TWO; KRNPATIENT,THREE; etc.
- Sample HL7 messages, "snapshots" of computer online displays (i.e., character-based screen captures/dialogues) and computer source code are shown in a *non*-proportional font and enclosed within a box. Also included are Graphical User Interface (GUI) Microsoft Windows images (i.e., dialogues or forms).
 - User's responses to online prompts will be boldface.
 - References to "<Enter>" within these snapshots indicate that the user should press the **Enter** key on the keyboard. Other special keys are represented within < > angle brackets. For example, pressing the **PF1** key can be represented as pressing <PF1>.
 - Author's comments are displayed in italics or as "callout" boxes.



NOTE: Callout boxes refer to labels or descriptions usually enclosed within a box, which point to specific areas of a displayed image.

- This manual refers in many places to the M programming language. Under the 1995 American National Standards Institute (ANSI) standard, M is the primary name of the MUMPS programming language, and MUMPS will be considered an alternate name. This manual uses the name M.
- Descriptions of direct mode utilities are prefaced with the standard M ">" prompt to emphasize that the call is to be used *only in direct mode*. They also include the M command used to invoke the utility. The following is an example:

```
>D ^XUP
```

- The following conventions will be used with regards to APIs:
 - Headings for programmer API descriptions (e.g., supported for use in applications and on the Database Integration Committee [DBIC] list) include the routine tag (if any), the caret ("^") used when calling the routine, and the routine name. The following is an example:
`EN1^XQH`
 - For APIs that take input parameter, the input parameter will be labeled "required" when it is a required input parameter and labeled "optional" when it is an optional input parameter.
 - For APIs that take parameters, parameters are listed in lowercase. This is to convey that the listed parameter name is merely a placeholder; M allows you to pass a variable of any name as the parameter or even a string literal (if the parameter is not being passed by reference). The following is an example of the formatting for input parameters:
`XGLMSG^XGLMSG(msg_type,[.]var[,timeout])`
 - Rectangular brackets [] around a parameter are used to indicate that passing the parameter is optional. Rectangular brackets around a leading period [.] in front of a parameter indicate that you can optionally pass that parameter by reference.
- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field/file names, and security keys (e.g., the XUPROGMODE key).



NOTE: Other software code (e.g., Delphi/Pascal and Java) variable names and file/folder names can be written in lower or mixed case.

How to Obtain Technical Information Online

Exported VistA M Server-based software file, routine, and global documentation can be generated through the use of Kernel, MailMan, and VA FileMan utilities.



NOTE: Methods of obtaining specific technical information online will be indicated where applicable under the appropriate topic.

REF: Please refer to the *Kernel Technical Manual* for further information.

Help at Prompts

VistA M Server-based software provides online help and commonly used system default prompts. Users are encouraged to enter question marks at any response prompt. At the end of the help display, you are immediately returned to the point from which you started. This is an easy way to learn about any aspect of VistA M Server-based software.

Obtaining Data Dictionary Listings

Technical information about VistA M Server-based files and the fields in files is stored in data dictionaries (DD). You can use the List File Attributes option on the Data Dictionary Utilities submenu in VA FileMan to print formatted data dictionaries.



REF: For details about obtaining data dictionaries and about the formats available, please refer to the "List File Attributes" chapter in the "File Management" section of the *VA FileMan Advanced User Manual*.

Assumptions About the Reader

This manual is written with the assumption that the reader is familiar with the following:

- VistA computing environment:
 - Kernel—VistA M Server software
 - VA FileMan data structures and terminology—VistA M Server software
- Microsoft Windows environment
- M programming language

This manual provides an overall explanation of Kernel and the functionality contained in Kernel V. 8.0. However, no attempt is made to explain how the overall VistA programming system is integrated and maintained. Such methods and procedures are documented elsewhere. We suggest you look at the various VA home pages on the World Wide Web (WWW) and VA Intranet for a general orientation to VistA. For example, go to the Veterans Health Administration (VHA) Office of Information (OI) Health Systems Design & Development (HSD&D) Home Page at the following Intranet Web address:

<http://vista.med.va.gov/>

Reference Materials

Readers who wish to learn more about Kernel should consult the following:

- *Kernel Release Notes*
- *Kernel Installation Guide*
- *Kernel Systems Management Guide*
- *Kernel Developer's Guide*
- *Kernel Technical Manual*
- *Kernel Security Tools Manual* (this manual)
- Kernel Home Page at the following Web address:

<http://vista.med.va.gov/kernel/index.asp>

This site contains other information and provides links to additional documentation.

If the reader is not already familiar with VA FileMan or MailMan, the respective user, programmer, and technical manuals for each should be obtained and reviewed. Other source documents describing overall VistA policy are:

- *VA Programming Standards and Conventions (SAC)*
- *MIRMO/OIFO Operations Document*

VistA documentation is made available online in Microsoft Word format and in Adobe Acrobat Portable Document Format (PDF). The PDF documents *must* be read using the Adobe Acrobat Reader (i.e., ACROREAD.EXE), which is freely distributed by Adobe Systems Incorporated at the following Web address:

<http://www.adobe.com/>



REF: For more information on the use of the Adobe Acrobat Reader, please refer to the *Adobe Acrobat Quick Guide* at the following Web address:

<http://vista.med.va.gov/iss/acrobat/index.asp>

VistA documentation can be downloaded from the Health Systems Design and Development (HSD&D) VistA Documentation Library (VDL) Web site:

<http://www.va.gov/vdl/>

VistA documentation and software can also be downloaded from the Enterprise VistA Support (EVS) anonymous directories:

- Albany OIFO REDACTED
- Hines OIFO REDACTED
- Salt Lake City OIFO REDACTED
- Preferred Method REDACTED

This method transmits the files from the first available FTP server.



DISCLAIMER: The appearance of external hyperlink references in this manual does *not* constitute endorsement by the Department of Veterans Affairs (VA) of this Web site or the information, products, or services contained therein. The VA does *not* exercise any editorial control over the information you may find at these locations. Such links are provided and are consistent with the stated purpose of this VA Intranet Service.

1. Introduction

Overview

Kernel is a set of utility programs that provides an interface between operating systems, VistA software, and users. A key component of Kernel is Signon/Security. Kernel's security features are grouped into four chapters in this manual:

- User Security
- Menu Management
- Audit Features
- Software Integrity

The purpose of Kernel's security modules is to restrict access to the VistA computer system to only authorized users, to restrict authorized users to those tasks (menus/options) which they need to perform their jobs, to monitor user actions, to monitor selected changes to the database, and to monitor changes to programs. As such, Kernel offers the system-wide protection of all data on a VistA system.

All VistA applications make use of Kernel's security features to segregate functions among employees. For example, the CoreFLS (Integrated Funds Distribution, Control Point Activity, Accounting and Procurement) software uses security keys to distinguish options that only Control Point Officials may use. Many applications now use the Electronic Signature feature as a validation of user identity when sensitive or privileged actions are required (e.g., Order Entry/Results Reporting and Radiology use the Electronic Signature to approve/verify orders and results). All applications now employ the Program Integrity Checker to determine if the software programs have been altered. Kernel's security features are central to the operation of the VistA system.

It is important to note first that all VistA applications offer some software-specific security features. For example, the Pharmacy applications all record the identification of the user who enters a medication order. The CoreFLS software records all the users who affect an order for procurement of goods, including Control Point users, the Purchasing Agent, the Accounting Technician, and the Warehouse staff. Many software applications preserve the original data as entered by the user and record updates to the data as amendments or adjustments, thereby preserving the data for both legal and retrospective review. Therefore, the ISO should also work with the Application Coordinators at the facility so that the inherent security of the VistA system as a whole is protected.



REF: The reader is also encouraged to review other VistA manuals, especially the *VA FileMan Getting Started Manual*, *VA FileMan Advanced User Manual*, *Kernel Systems Management Guide*, and the *MailMan User Manual*. Review of the Technical Manuals for VA FileMan, Kernel, and MailMan is also suggested.

This manual addresses Kernel in terms of its security features. The material presented is intended for users whose primary interest is ADP (Automated Data Processing)-based information security. In this manual you will learn how to:

- Control user security
- Manage menus and options
- Establish and review audits
- Perform program integrity checks

This manual discusses each of these security tasks, devoting one chapter to each. Each chapter provides you examples of the use of the options within each section. You are encouraged to begin with the section Kernel Security During User Sessions in Chapter 1 to ensure that you first understand the general operation of Kernel.

Software management is not an issue for the Information Security Officer (ISO). Responsibility for management of this and other parts of Kernel rests with IRM (Information Resources Management) Service.



REF: Information about overall Kernel software management can be found in the *Kernel Systems Management Guide* and *Kernel Technical Manual*.

Software Management

Throughout this manual, advice and instruction are offered about the numerous tools Kernel provides for overall VistA management. Site parameters, for example, are discussed in various sections; information about managing computer security is discussed throughout.



REF: The *Kernel Installation Guide* also includes information about software management, such as recommended settings for site parameters and scheduling time frames for tasked options.

The *Kernel Systems Management Guide* also contains extensive information about managing and running Kernel.

To protect the security of VistA systems, distribution of this software for use on any other computer system by VistA sites is prohibited. All requests for copies of Kernel for non-VistA use should be referred to the VistA site's local Office of Information Field Office (OIFO).

Otherwise, there are no special legal requirements involved in the use of Kernel.

2. User Security

User security is the cornerstone of Kernel's system security features. This chapter discusses the ways in which the VistA computer system recognizes and screens users. The following topics are addressed:

- Kernel Security During User Sessions
- Kernel Security Codes
- Tips and General Advice for Users
- Reviewing Users

Kernel Security During User Sessions

Device Check

A user session begins when a user presses the return or enter key on their terminal. This simple step activates the Security features of Kernel. First, Kernel determines the device being activated and recognizes it as the user's terminal. Kernel then reviews the parameters for that device. Among other things, Kernel notes if the device is permitted to support users at that time. The first step of security is to assure that the device can be used.

User Identification

Kernel next queries the user for identification. This is done as a two-step procedure:

1. The user is first prompted to provide an Access code.
2. This is followed with a prompt for the user's Verify code.

The user must provide a valid pair of codes before the VistA system allows the user to perform any other actions. Both the Access and Verify codes that make up the pair must be valid. If the codes entered do not constitute a valid pair, the VistA system simply informs the user that the pair is invalid, but will not say which of the two codes, if either, is valid. Thus, the secrecy of both codes is maintained against inappropriate use.

If the user does not enter valid codes, the VistA system re-prompts the user for the codes. This is repeated until either:

- The user finally enters a valid pair of codes. Then Kernel proceeds to offer the user a set of authorized options.
- The user continues to attempt to enter valid codes, gives up, or enters a caret ("^") to exit from the signon process.
- The VistA system determines that the maximum allowable number of attempts has been exceeded and locks the device. Once locked, the device *cannot* be used by anyone until Kernel determines that the lockout time has passed. The number of attempts is established by the IRM Service. This may vary among devices. Typically, for a remote device such as a modem, the number of

attempts is set to a low value (e.g., 3), so that the potential for mischievous or malicious hackers to gain access to your system is minimized. After Kernel has locked a device, it can record the codes used for the repeated attempts. This is known as the Failed Access Audit. You can establish the detail to which your VistA system records for these events through the Establishing Kernel Audit Parameters option.

Menus and Options

Once the user is successfully recognized and signed on to the VistA system, a set of options is presented on the screen. These options are grouped into menus. Each user has a primary menu, which should reflect the duties that user needs to perform. For example, the primary menu for an Information Security Officer (ISO) will probably be System Security Options. This menu leads the user to other menus and options which cause the computer to perform specific operations. Sometimes, the options within a menu are locked with a security key. This key is essentially a second level of security. The user can only use a locked option if they have been given the appropriate security key. If the user does *not* have the security key, then even if the locked option is on the user's menu, they will *not* be able to use it. Options that provide specialized or supervisory access are usually locked with a security key.

Many options update data in the VistA system. Kernel and VA FileMan control these updates. VA FileMan is the database management system that defines data files and governs data input and reporting. Kernel and VA FileMan provide security controls to restrict data manipulation. Access authority can be defined for each user and can be temporarily granted through the use of options. The following types of access can be granted:

Access	Description
Read	Allows a user to read data from a file.
Write	Allows a user to update the data in a file.
Delete	Allows a user to permanently remove data from a file.
LAYGO	Allows a user to create a new entry when editing a file. (A special form of write.)
DD	Allows data dictionary access for changing or deleting the structure of the file itself (data dictionaries define what types of data are stored, where they are stored, and the relationships among the data fields).
Audit	Allows a user to define how a file is audited by VA FileMan.

Table 2-1. Security access types

Changes are made to data dictionaries by the IRM staff either to install authorized changes or to alter locally developed VistA applications.

The data on your VistA system can be very strictly protected. Each user is granted the access needed to perform a job. At the same time, users are prohibited from reviewing or altering data not essential to their tasks.

While users are working with the VistA system, it monitors some of their activity. Kernel is able to record the use of all options that the user activates. Thus, Kernel can track the use of sensitive options. This capability can also be used to determine which options are most frequently used.

Kernel registers if the user is inactive for a pre-determined length of time. If the user does not interact with the VistA system within the time-out period, Kernel returns them to the previous prompt, eventually terminating the session of the user. By automatically logging out inactive users, the VistA system limits the possibility that unattended terminals are not used by others.

Kernel facilitates the activities of all users while simultaneously providing essential security. The topics that follow describe how to use the security features of Kernel to monitor your VistA system.

Kernel Security Codes

The first line of security that Kernel employs is to ensure that only authorized users may access a VistA computer system. This is done with the security codes of your system. These codes are known as Access and Verify codes and Electronic Signature. Access and Verify codes are employed to recognize users. The Electronic Signature is a secret password that some users employ to sign documents via the computer. It is used by many VistA Applications as an additional security check for sensitive operations.

Access/Verify Codes

With each signon, the user must enter two codes to be recognized and allowed to proceed. These codes are the Access and Verify codes:

- **Access Code**—The Access code is assigned by IRM Service. This code is used by the computer to recognize the user. Each user has a unique Access code. The only way this code can be changed is for the IRM Service to edit it. When the code is established by IRM, it is encrypted (i.e., it is scrambled according to a cipher). The code is stored in the computer only in this encrypted form. Thus, even if the Access code is viewed, the viewer cannot determine what the user actually types to tell the computer this code.
- **Verify Code**—The Verify code is also generally assigned by the IRM Service. Like the Access code, the Verify code is also encrypted. This code is used by the computer to verify that the person entering the Access code can also enter a second code correctly. Thus, this code is used to determine if the user can verify who they are. After being given Access and Verify codes, the user can change the Verify code at any time. This is done through the Edit User Characteristics menu, which is found in the Toolbox menu (the Toolbox menu can be accessed by entering two question marks ("??") at the menu prompt.) It is important to note that the VistA system requires users to change their Verify code on a regular basis. The frequency of changing these codes is set in the KERNEL SYSTEM PARAMETERS file (#8989.3). Security policy states that codes should be changed at least every 90 days.

If IRM Service assigns only the Access code, then the first time the new user signs on to the VistA system, Kernel prompts them to establish a Verify code. The user *cannot* proceed with the session until this code is entered and validated by Kernel.

Access and Verify codes *must* adhere to the following criteria:

- Access and Verify codes *cannot* be identical.
- Verify codes (i.e., passwords) must be at least 8 characters in length.
- Strong passwords in general contain at least three of the following four character types:
 - Uppercase letters
 - Lowercase letters
 - Numbers
 - Special characters that are neither letters nor numbers (e.g., #, &, *, or @)

Because VistA is case-insensitive, VistA only has three sets of characters from which to build a Verify code (i.e., password):

- Letters (of any case)
- Numbers
- Special characters that are neither letters nor numbers (e.g., #, &, *, or @)
- Verify codes must be changed at least every 90 days. You must change your Verify code at periodic intervals as specified by IRM. Information systems shall not permit re-assignment of the last three passwords used. When required, you will be prompted during signon to pick a new Verify code.
- Accounts that have been inactive for 90 days shall be disabled.
- To preclude password guessing, an intruder lock out feature shall suspend accounts after five invalid attempts to log on. Where round-the-clock system administration service is available, system administrator intervention shall be required to clear a locked account. Where round-the-clock system administration service is not available, accounts shall remain locked out for at least ten minutes.



REF: These rules are taken from the *VA Account and Password Management Interim Policy* document.

NOTE: All of these restrictions are enforced whenever Access or Verify codes are created or changed.

REF: These changes were made to meet VHA DIRECTIVE 6210 available at the following Web address:

REDACTED

These restrictions are enforced whenever Access or Verify codes are created or changed. Users should avoid codes that offer no secrecy (e.g., a child's or spouse's name, a phone number or license plate). If desired, the system can automatically generate codes for users. However, these auto-generated codes can be cryptic (e.g., A\$BC402) and are often difficult for users to remember.

For both the Access and Verify codes, it is important to remember that there is no way for you to see the codes that were entered for any user. If users forget their Access or Verify codes, you must assign replacement codes.

Each time users sign on, they see a dialogue similar to the following (depending on the unique physical characteristics of your system):

```
Volume set: VAH      UCI: VAH      Device:  _LTA9130: (VAH604/LC-1-1)

ACCESS CODE:  <enter access code, it will not be displayed on screen>
VERIFY CODE:  <enter verify code, it will not be displayed on screen>
GOOD MORNING  Five      YOU LAST SIGNED ON TODAY AT 8:30
You have 3 new messages.
```

Figure 2-1. Entering a *valid* Access/Verify code pair—Sample user dialogue

When the system prompts for the Access and Verify codes, the text that the user types does *not* appear on the terminal screen. Thus, even if someone is watching a user sign on, the codes are not shown. If the user enters codes that the system recognizes as valid, it greets the user and proceeds according to the privileges that are assigned to that user.

If the user fails to enter a set of codes that the system recognizes, the user sees a different dialogue, as shown below:

```
Volume set: VAH      UCI: VAH      Device:  _LTA9130: (VAH604/LC-1-1)

ACCESS CODE:  <enter access code>
VERIFY CODE:  <enter verify code>
Not a valid ACCESS CODE/VERIFY CODE pair

Volume set: VAH      UCI: VAH      Device:  _LTA9130: (VAH604/LC-1-1)

ACCESS CODE:  <enter access code>
VERIFY CODE:  <enter verify code>

GOOD EVENING  Five      YOU LAST SIGNED ON TODAY AT 17:24

There was 1 unsuccessful attempt since you last signed on:
```

Figure 2-2. Entering an *invalid* Access/Verify code pair—Sample user dialogue

Notice that the computer tells the user that the pair of codes was not valid. It does *not* specify that the Access code was bad or that the problem was with the Verify code. Thus, a potential hacker is not given any clues as to whether they are closing in on a valid account.

Take another look at the signon dialogue. Notice that the computer tells the user when they last signed on to the system. By asking each user to pay attention to this greeting, you can enlist them in the security process. If the computer reports to a user that they were signed on the day before, and the user knows that they were not at work, then they can report that information as a possible security violation.

If the user fails to gain access (i.e., multiple unsuccessful attempts are made in succession), then the device may lock. When the device locks, then no user can log onto the system through that device until either the lockout time expires or IRM Service intervenes. The device lockout time is defined in the KERNEL SYSTEM PARAMETERS file (#8989.3) as a system default but may be overridden by a lockout time which IRM Service associates with a specific device (e.g., a modem).

Electronic Signatures

A primary aspect of security in many VistA software applications involves the use of Electronic Signatures. Individuals in the system who have authority to approve actions, at whatever level, can enter and edit their own Electronic Signature code. This code is required before data may pass from one level of processing to another. An example of this is the release of a Request for Purchase (Form 2237) in CoreFLS. The Control Point Official must first enter their Electronic Signature before the request will be passed to Personnel Property Management for review and processing by Acquisition and Material Management Service.

Like the Access and Verify codes used when gaining access to the system, the Electronic Signature code is *not* visible on the terminal screen. These codes are also encrypted so that even when viewed in the NEW PERSON file (#200) by those with the highest levels of access, they are indecipherable.

Electronic Signatures can be established in two ways:

- The IRM Service can enter a code for the user.
- The code can be left blank and the user can establish it the first time the code is required. Kernel informs the user that a code is required and the user must enter one in order to proceed.

Like the Verify code, the user can change their Electronic Signature code at any time. Kernel provides an option for the user to do this. Those software applications that allow Electronic Signatures may also include this option for the appropriate class of users.

Tips and General Advice for Users

It is important that all users be educated as to how they can assist with Automated Data Processing (ADP) Security.

Users should be instructed that when performing basic tasks at a computer terminal, they will probably enter or review information that may be considered sensitive. As a result, protecting security codes is a crucial part of each user's job. Whether it is called an Access code, a Verify code, Password or a Logon code, security codes are the first line of defense against unauthorized users who may seek to defraud the VA or compromise its computer system. To protect security codes, teach your users to always comply with the following guidelines:

- Do *not* share Access and Verify codes with anyone.
- Do *not* tape Access and Verify codes under a desk, on a wall, on a terminal, or in other obvious hiding places.
- Do *not* leave the terminal unattended while you are signed on. Sign off or "lock" your computer each time you leave your terminal unattended.
- Do *not* use obvious codes such as your name and date of birth or your child's name and date of birth.
- Do *not* forget to protect all printouts, computer documents, and media containing sensitive data.
- Do *not* use the computer for personal business.
- Do change your security codes at regular intervals. Changing security codes regularly reduces the likelihood that an unauthorized user will detect a unique code.
- After signing on, you have a limited amount of time to enter data at any one prompt. If you fail to enter information within an allotted time period, you will be returned to the Select Option prompt. There, Kernel again waits for you to enter information. Should you fail once again to respond within the allotted time period, the computer will ask if you want to HALT. Failure to say No at this level causes the computer to log you off the system.

Reviewing Users

Your VistA computer system probably has hundreds, if not thousands, of authorized users. As was discussed in the Kernel Security During User Sessions topic, each user has a primary menu. This menu leads the user to the various options that they can invoke on the computer.

It is important to review the access each user has on your computer system. Among the items to check are:

- Does this user still require access to the computer?
- What files/data can this user access?
- Is the user's access level appropriate?
- If the user has changed departments, has their primary menu been updated to reflect new duties?

Using the Kernel Security options described in this chapter, you can gather the information to address the above questions. But remember, Kernel *cannot* tell you if the access is appropriate, it can only report the current access privileges for the users.

General Information About Users

The List Users option lists all users alphabetically. It shows each user's name, user number, the primary menu option, and the last date/time the user signed on to your system. The user number is used by the computer to uniquely distinguish users.

```
Select System Security Option:  REV <Enter> iew Users
Select Review Users Option:  LIST <Enter> users
START WITH NAME: FIRST//  A
GO TO NAME: LAST//  AZ
DEVICE: <Enter>
```

Figure 2-3. List users option—Sample user dialogue

USER LIST		MAR 11, 1992 10:48 PAGE 1	
NUMBER	NAME	PRIMARY MENU OPTION	LAST SIGN-ON DATE/TIME
69	KRUSERH,EIGHT	PRCSP CLERK	MAR 7, 1992 13:50
12	KRUSERI,NINE	XMUSER	MAR 10, 1992 10:17
42	KRUSERJ,TEN	PSG FILE	MAR 10, 1992 10:30
50	KRUSERK,ELEVEN	SDAPP	MAR 8, 1992 15:03

Figure 2-4. List users option—Sample report

The User Inquiry option displays various attributes for a specific user. If the user is currently signed on, it displays the job and device numbers, the signon time, the secondary menu, and the option and menu path currently being used by that individual. Otherwise, it shows the last signon time. It also displays the security keys held by the user. This option can be tailored by the IRM staff, so the following example may not exactly match that shown by your system:

```
Select Review User Option:  US <Enter> er Inquiry
Select NEW PERSON NAME:   KRUSERD,FOUR

KRUSERD,FOUR  (#23)
-----
Job: 576727625 (22602A49) on ISC,ISC:ISC6V3 from APR 26, 1992@16:41:18
Device: _LTA9130: (ISC604/LC-1-1)
Menu path:
      ISC MANAGER'S OPTION
        Systems Manager Menu
          System Security
            Review Users
              User Inquiry

ATTRIBUTES
-----
Creator    BROWN,BUSTER  Date entered  Aug 1, 1986
Primary menu    EVE      Fileman code(s)      @
Time-out      300      Type-ahead      Y
Title         Programmer  Office Phone    x1234
Auto-Menu     NO MENUS GENERATED  Last Sign-on   Apr 26, 1992

Secondary Menu Options
-----

TalkMan
Connect to IDCU Network
Project Management Menu

Keys Held
-----
XMMGR  XUMGR  XUPROG          XMNET  XUPACK
ZTMQ   XUSER  XMTALK  XUPROGMODE
XMPRIORITY  XUAUDITING  XUARCHIVE
```

Figure 2-5. User Inquiry option—Sample user dialogue and report

Users can generate a status report at any time by using the User Status Report option. This report shows the current users who are signed onto the system. For each user, the job number, device, time they signed on and the option they are currently using are displayed.



NOTE: The job number is an identification the computer employs for the duration of a user session.

```
Select Review Users Option: User S <Enter> tatus Report
Lookup pass .....
MAR 11, 1992 12:51      USER STATUS REPORT      VAH,VAH      PAGE 1
```

JOB NUMBER	USER NAME	TIME ON	DEVICE	CURRENT MENU OPTION
22602A49	KRNUSERA,ONE	10:50	_LTA9145:	User Inquiry
23897A44	KRNUSERD,FOUR	10:52	_LTA9130:	User Status Report
47582B99	KRNUSERL,TWELV	10:53	_LTA9270:	Print Bill

Figure 2-6. User Status Report option—Sample user dialogue and report

You can also locate an active user on the system by using the Find a User option. The user must be in the same account (or UCI) as you are. If the user is also on the same CPU, then you will be shown the user's menu path (i.e., all the options that user activated to get to the one they are currently using).

```
Select Review Users Option: FIND <Enter> a user
Find User: KRNUSERD,FOUR
```

User: KRNUSERD,FOUR is found on;
 Job: 576727625 (22602A49) on ISC,ISC:ISC6V3 from APR 26, 1992@16:41:18
 Device: _LTA9130: (ISC604/LC-1-1)
 Menu path:
 ISC MANAGER'S OPTION
 Systems Manager Menu
 System Security
 Review Users
 Find a user

DONE

Figure 2-7. Find a user option—Sample user dialogue

User's Access to VA FileMan Files

Ordinarily, access to data on the system is controlled through Menu Manager and the assignment of options. Users should only be assigned options that are appropriate to their duties. For example, most pharmacy users would have access to some subset of the pharmacy software options. Then, any data that can be accessed through the pharmacy options, can be accessed by pharmacy users. So access to data in VA FileMan files is ordinarily controlled by the assignment of options.

There is a special set of options, however, that let users manipulate data directly, not through any software; this set of options is part of the VA FileMan software itself. These VA FileMan options let

users directly view, modify, and print data from VA FileMan files, as well as modify the file structure itself. For users who have VA FileMan options in their menu tree, it is obviously important to control what access they have to files so that they do not have inappropriate access to view and modify data and files.

Access to VA FileMan files can be controlled in two ways; which way is used at your site depends on whether Kernel's optional File Access Security system has been enabled at your site:

- **File Manager Access Code**—If Kernel's optional File Access Security system has *not* been installed, access is controlled by a user's File Manager Access code.
- **ACCESSIBLE FILE multiple in their NEW PERSON file (#200)**—If Kernel's optional File Access Security system has been installed, access is controlled by entries in each user's ACCESSIBLE FILE multiple in their NEW PERSON file (#200) entry.

Has the File Access Security System been Enabled?

To determine if the File Access Security system has been installed, you can check with your IRM Service. You can also look at the Review Users menu. If the Access to VA FileMan Files option is present, and *not* marked out of order, that usually indicates that File Access Security system has been installed.

If the File Access Security System Has Not Been Enabled

If Kernel's File Access Security system has *not* been enabled on your system, a user's access to VA FileMan files through VA FileMan options is controlled by their File Manager Access code. You can determine a user's File Manager Access code from the User Inquiry report, in the Attributes section; it is the character string listed as "FileMan code(s)".

Each VA FileMan file can have an Access code associated with each of six types of access to the file (Read, Write, Delete, LAYGO, DD, and Audit, see Table 2-1). A user can only access a file in one of these ways if a character in the user's File Manager Access code matches the security code associated with that type of access for the file. The exception is if the user has a File Manager Access code of "@", programmer access, which enables access of all types to all files.



REF: For more information on file security based on File Manager Access code, please refer to the *VA FileMan Advanced User Manual*.

If the File Access Security System Has Been Enabled

If Kernel's File Access Security system has been enabled, a user's access to VA FileMan files through VA FileMan options is controlled by what files are listed in that user's ACCESSIBLE FILES multiple, in their NEW PERSON file (#200) entry.

To see which VA FileMan files a user may access when Kernel's File Access Security system has been enabled, use the Inquiry to a User's File Access option. This option displays the files a user can access, showing the user's level of access (e.g., READ, DELETE) for each file. The report also shows users who have programmer access to all files, which means that the user's File Manager Access code is set to "@". In the following examples (Figure 2-8 - Figure 2-11), you can see that ONE KRNUSERA has specific access to two files. On the other hand, the user TWO KRNUSERB has programmer mode access and so has access to all files. This option displays the files a user can access, showing the user's level of access (e.g., READ, DELETE) for each file.

```
Select Access to VA FileMan Files Option: INQ <Enter> uiry to a User's File Access

Select USER NAME: KRNUSERA, <Enter> ONE

DEVICE: PRINTER-1
```

Figure 2-8. Inquiry to a User's File Access option—Example of a user with access to two files

```
USER ACCESS TO FILES      MAR 16,1992   18:15   PAGE: 1
KRNUSERA,ONE  (11) DD    DELETE LAYGO  READ   WRITE  AUDIT
-----
3.1          TITLE                      YES    YES    YES    YES
3.2          TERMINAL TYPE YES          YES    YES    YES    YES
```

Figure 2-9. Inquiry to a User's File Access option—Sample report of a user with access to two files

```
Select Access to VA FileMan Files Option: INQ <Enter> uiry to a User's File Access

Select USER NAME: KRNUSERB, TWO

DEVICE: PRINTER-1
```

Figure 2-10. Inquiry to a User's File option—Example of a user with programmer access to all files

```
USER ACCESS TO FILES      MAR 16,1992   18:16   PAGE: 1
KRNUSERB,TWO  (00) DD    DELETE LAYGO  READ   WRITE  AUDIT
-----
Programmer Access to All Files
```

Figure 2-11. Inquiry to a User's File option—Sample report of a user with programmer access to all files

Sometimes, it is most useful to determine first if the focus should be on a specific set of files, then look at file access to those files. The List Access to Files by File number option lists, by file number, those users who have access to those files. For each type of access (e.g., READ, DELETE), the user's authority is listed. Thus, if a user has YES listed under the WRITE column of this report then that user can, via VA FileMan, WRITE data into the file.

```
Select Access to VA FileMan Files Option: L <Enter> list Access to Files by File
number
START WITH WHAT FILE: INSTITUTION// 3.2 <Enter>  TERMINAL TYPE      (112 entries)
GO TO WHAT FILE: TERMINAL TYPE// 3.5 <Enter>  DEVICE      (195 entries)
DEVICE: PRINTER-1
```

Figure 2-12. List Access to Files by File number option—Sample user dialogue

USER ACCESS TO FILES		MAR 11, 1992		10:57	PAGE: 1		
USER #	NAME	DD	DELETE	LAYGO	READ	WRITE	AUDIT

FILE:	3.2						
2	KRNUSERC, THREE		YES	YES	YES	YES	
7	KRNUSERD, FOUR	YES	YES	YES	YES	YES	
9	KRNUSERA, ONE		YES	YES	YES	YES	YES
FILE:	3.4						
7	KRNUSERD, FOUR		YES	YES	YES	YES	YES
9	KRNUSERE, FIVE		YES	YES	YES	YES	YES
FILE:	3.5						
7	KRNUSERD, FOUR	YES	YES	YES	YES	YES	YES

Figure 2-13. List Access to Files by File number option—Sample report

The Print Users Files option lists, by user, each file the user has access to and what that access is (e.g., READ, DELETE). Users who have no access are not listed.

```
Select Access to VA FileMan Files Option: Pr <Enter> int Users Files
START WITH NAME: FIRST// SIX
GO TO NAME: KRNUSERF
DEVICE: PRINTER-1
```

Figure 2-14. Print Users Files option—Listing file access for multiple users, sample user dialogue

USER ACCESS TO FILES			MAR 11, 1992	10:59	PAGE: 1		
FILE #	ACCESSIBLE FILE	DD	DELETE	LAYGO	READ	WRITE	AUDIT

KRNUSERF, SIX	(7)						
3.2	TERMINAL TYPE			YES	YES	YES	
3.4	COMMUNICATIONS	YES	YES	YES	YES	YES	
3.5	DEVICE		YES	YES	YES	YES	

Figure 2-15. Print Users Files option—Sample report



REF: For more information about Kernel's File Access Security system, please refer to the "File Access Security" chapter in the *Kernel Systems Management Guide*.

3. Menu Manager Security

This chapter discusses the methods of managing user security by use of the Menu Manager. The following topics are addressed:

- Examining Menus and Options
- Secure Menu Delegation

Menu Manager provides user-oriented menus that present the user with a selection of the activities they are authorized to perform on the VistA computer system.

In most cases, users have a set of options that are organized into a menu. Each menu can contain other menus or options that the user employs to perform tasks. Some menus or options have a security key. For these options, the user must hold the key or the option cannot be executed by that user. Moreover, the auto-menu and single-question-mark menu displays do *not* list locked options, so users ordinarily do *not* know that the locked option is available, even if it is grouped within a menu for that user. A double-question-mark menu display is needed to show the presence of the locked option. Thus, menus on your system are used to grant privileges to users.

Many of the security options available on your system either audit option use or display user options as part of their reports. Therefore, it is important that you become familiar with the tools and options you can use on your system and how they are assigned.

Examining Menus and Options

The Option Access By User option shows you which users can actually access and activate an option. You will be prompted for the name of an option. You will also be asked if you want to see menu paths. Menu paths are all the options that the user must select to actually get to the option you have selected to review. In the following example (Figure 3-1), only one menu path is shown. Those users must first pick the option EVE (Systems Manager Menu), then XUTIO (Device Handler) to get to XUDEV (Device Edit).

```
Select System Security Option: MENU <Enter> Management Review
Select Menu Management Review Option: Option <Enter> Access By User
Select OPTION NAME: DEVICE EDIT <Enter> XUDEV          Device Edit
Show menu paths? NO// Y <Enter> (YES)
DEVICE: HOME// PRINTER-1
```

Figure 3-1. Option Access By User option—Sample user dialogue

ACCESS TO 'Device Edit' [XUDEV]			
USER NAME	LAST ON	PRIMARY MENU	PATH(S)
-----	-----	-----	-----
KRNUSERA, ONE	03/10/92	EVE	1
KRNUSERD, FOUR	03/11/92	EVE	1

		MENU PATH(S)	-----
1. EVE ... XUTIO ... XUDEV			

Figure 3-2. Option Access By User option—Sample report

Obviously, it is difficult to make sense of an option just by seeing its name (e.g., XUDEV) and sometimes the associated text does not help significantly (e.g., Device Edit). The option to display information about a given option is called Inquire. By using the Inquire option, you can review an option. The display shows everything recorded about the characteristics of that option. If the option requires a security key, you will see a label LOCK and immediately to the right of that label will be the key that locks that option. There will also be many fields that may seem cryptic and confusing (e.g., DIC {DIC}). These fields are used by Kernel to affect the actions which the option requires.



REF: For a more comprehensive discussion of these fields, please refer to the *VA FileMan Programmer Manual*.

Select Menu Management Review Option: Inquire
Which OPTIONS item to display: DEVICE EDIT <Enter> XUDEV Device Edit

Figure 3-3. Inquire option—Sample user dialogue

OPTION LIST		OCT 5,1994 14:29	PAGE 1

NAME: XUDEV		MENU TEXT: Device Edit	
TYPE: edit		CREATOR: POSTMASTER	
HELP FRAME: XUDOC DEVICE LOOKUP		RESTRICT DEVICES?: NO	
DESCRIPTION:		This option will change the device characteristics for a given device.	
DIC {DIC}: 3.5		DIC(0): AEQMLZ	
DIE: 3.5		DR {DIE}: .01:9999	
TIMESTAMP OF PRIMARY MENU: 54262,51324			
UPPERCASE MENU TEXT: DEVICE EDIT			

Figure 3-4. Inquire option—Sample report

You can also use the Print Options File option to produce a formatted listing of the OPTION file (#19), showing each option and its associated parameters. You can limit the set of options printed by setting the bounds of your SORT. If you simply press <Enter> at the "START WITH NAME: FIRST//" prompt, and accept LAST at the "GO TO NAME: LAST" prompt, you will get all options.

CAUTION: On most Vista systems there are thousands of options; so, be warned that choosing all options generates a very large printout.

```
Select Menu Management Review Option: Print <Enter> Options File
SORT BY:  NAME// <Enter>
START WITH NAME: FIRST// XUDEV
GO TO NAME: LAST// XUDZ
          WITHIN NAME, SORT BY: <Enter>
DEVICE: PRINTER-1
```

Figure 3-5. Print Options File option—Sample user dialogue

```
Menu list by display terms [MENULIST]          MAR 12, 1992   11:27   PAGE 1
MENU TEXT      NAME      TYPE
DESCRIPTION
-----
Device Edit      XUDEV      edit
  This option will change the device characteristics for a given device.

  Edit file:   3.5
-----
Display Device Data  XUDISPLAY  print
  This option is used to print a list of all the devices in the DEVICE
  file.
  Print file:  %ZIS (1,
-----
```

Figure 3-6. Print Options File option—Sample report

As was mentioned at the beginning of this topic, the options on your system are organized into menus. One way to view these menus is as an outline that is used to guide the user through their actions. To see these menus in their outline form, use the Diagram Menus option. This option displays in outline form, all of those menus and options that are available to a given user, according to their security and primary menu.



NOTE: These diagrams are generally large, therefore, it is best to request the output to be printed.

Another way of seeing the structure of a menu can be achieved with the Abbreviated Menu Diagrams option. This form provides an abbreviated display of all the options available to a given user, including all menus and options, according to the user's security and primary menu (Option Names, Menu Text, and Synonyms).

Secure Menu Delegation

Your computer system probably has thousands of options. Managing these options and making sure that each user has the options they need to do their job is a time-consuming task. To assist the IRM Service in performing this task, Secure Menu Delegation was developed. Secure Menu Delegation allows the IRM Service to delegate the task of maintaining the options for each user to a trusted set of other users. From a security perspective, it is important to know who on your system has the privilege of managing menus and options. Secure Menu Delegation provides several options that support this goal.

If you need to review the options that a user can manage, use the Show a Delegate's Options option. This lists all the options delegated to a specified user. This user can assign which options are to be shown to other users. For each option, the date it was delegated and the user who delegated the option are shown.

```
Select Secure Menu Delegation Option:  SHOW <Enter> a Delegate's Options
Select NEW PERSON NAME:  KRNUSERA,ONE
```

Figure 3-7. Show a Delegate's Options option—Sample user dialogue

```
KRNUSERA,ONE a delegate of: KRNUSERD,FOUR on 8/28/91 at level 1

OPTION (INTERNAL #)          MENU TEXT                DELEGATED      (DUZ)
XUDEV   (93)                Device Edit          8/25/91        (22)
XUTERM  (105)               Terminal Type Edit   8/27/91        (22)
```

Figure 3-8. Show a Delegate's Options option—Sample report

You can also use the List Delegated Options and their Users option. This option produces a list of the options, in alphabetical order that have been delegated. For each option, the users who have been delegated these options are listed.

```
Select Secure Menu Delegation Option:  LIST <Enter> Delegated Options and their
Users

DEVICE:  PRINTER-1
```

Figure 3-9. List Delegated Options and their Users option—Sample user dialogue

```
DELEGATED OPTIONS BY USER          MAR 11,1992  11:12  PAGE 1
DELEGATED OPTIONS      MENU TEXT          NAME
-----
DG10                   Register Patient          KRNUSERG,SEVEN
DG10NOSTAT             10/10 Print, no registrat KRNUSERG,SEVEN
XUDEV                  Device Edit              KRNUSERD,FOUR
XUTERM                 Terminal Type Edit       KRNUSERD,FOUR
```

Figure 3-10. List Delegated Options and their Users option—Sample report

In addition, you can use the Print All Delegates and their Options option. This report displays, by user, all delegates and their delegated options. For each option, the date it was delegated and the user who delegated it are shown.

Select Secure Menu Delegation Option: **PRINT** <Enter> All Delegates and their Options

DEVICE: **PRINTER-1**

Figure 3-11. Print All Delegates and their Options option—Sample user dialogue

DELEGATED OPTIONS BY USER		OCT 5,1994 14:32	PAGE 1
NAME	CREATED BY	CREATION DATE	DELEGATION LEVEL
DELEGATED OPTIONS	DELEGATED BY	DATE DELEGATED	

KRNUSE ^{RG} ,SEVEN	KRNUSE ^{RA} ,ONE	MAY 20,1991	1
DG10	0	MAY 20,1991	
KRNUSE RD ,FOUR			
LRZ INQUIRE	KRNUSE ^{RA} ,ONE	JUL 21,1994	
LRZ MAIN			

Figure 3-12. Print All Delegates and their Options option—Sample report

4. Kernel Audit Features

Audit features of Kernel make it possible to monitor a wide range of computing activity. The following audits are discussed in this chapter:

- System Access Audits
- Option and Server Usage Audits

This chapter describes how to use options to:

- Set and display audit parameters.
- Initiate audits.
- Print reports.
- Purge audit logs for audits.

The security officer and site manager each have interests and responsibilities with system audits. Both are concerned with the following:

- Prevention of unauthorized access.
- Exercise of inappropriate levels of access authority.
- Potential corruption of the VistA database through inappropriate alteration of data or dictionaries.

Designing and carrying out intelligently planned audits can answer security needs of the facility and help ensure system integrity.

IRM's Responsibility

While maintaining a secure computing environment is the primary responsibility of the Information Security Officer, it is also one of many responsibilities of IRM Service. IRM Service must also attend to problems of system performance such as response time. An audit can degrade response in two ways. The first involves the system resources used to capture audit data during the interactive user session. If a commonly used option is being flagged every time it is invoked, considerable resources will be devoted to the auditing task. The second effect is indirect; as disk space is consumed with audit or other data, an overall slowdown may result as access to a crowded disk with fragmented files takes added time. IRM Service, then, while encouraging the ISO to undertake audits, would also urge that the minimal amount of audited data be collected and that it be purged when no longer needed as an audit trail.

Initiating Audits

One approach to take when embarking on a system review is to examine the existing audit logs that are automatically maintained by the system. These are the Sign-on Log and the PROGRAMMER MODE LOG file (#3.07). Observing daily activity over the course of several weeks provides a baseline for judging whether an occurrence is unusual. If an unexpected event or trend is seen, an audit can then be undertaken for a specified time period as an investigation. The results can be reviewed, printed if desired, and then purged. The goals of system security as well as system performance can thus be achieved.

When considering an event to audit, research should be done to determine whether a mechanism is already in place within a VistA software application. The Pharmacy software, for example, may include an option for recording the names of users who approve and verify patient prescriptions. The security measures of other software should be similarly catalogued. IRM staff and application coordinators will be able to provide this information.

System Access Audits

There are three topics associated with the auditing of system access:

- Maintaining a list of old Access and Verify codes.
- Recording information about successful signon attempts.
- Recording information about unsuccessful signon attempts.

Old Access and Verify Codes

As described earlier in this manual, Access and Verify codes periodically need to be changed. A user can choose a new Verify code at any time or may be required to do so if the Lifetime of Verify Code limit has been reached. When a new code is chosen, the old code is kept on file and *cannot* be reused until the file of old codes is purged.

Preventing reuse of codes is a security measure. A user could otherwise simply change the new code back to the previously used one, thus defeating the purpose of requiring the changing of codes. It is wise to retain the list of old codes for some amount of time, but it is also reasonable to purge on occasion. Disk space is one concern. Another is the ease with which users can select new codes, the ones on the list of old codes being unavailable.

Purge

The Purge Log of Old Access and Verify Codes option is used for purging the list of old codes and is located on the System Manager's Manage User File menu. Purging is a function that can be carried out either by IRM Service or the Information Security Officer. An example of the on-line dialogue is as follows:

```
Select User Management Option: Manage User File

Select Manage User File Option: PURGE <Enter> Log of Old Access and Verify Codes

This option will purge the log of old access and verify codes.
It will remove the record of all inactive access and verify codes older
than the date specified and allow for their reuse.

Do you wish to continue? NO// Y <Enter> (YES)

How far back do you wish to retain codes? (7-90 days) 7

54 old access codes have been purged.

445 old verify codes have been purged.
```

Figure 4-1. Purge Log of Old Access and Verify Codes option—Sample user dialogue

Sign-on Log

Each time a user successfully signs on, an entry is automatically made in the Sign-on Log.

Print Reports

The Print Sign-on Log option displays the name of the user, the time of sign-on, the device used, and the elapsed time for the session.



NOTE: The elapsed time cannot always be determined because an error may cause an abnormal exit or the user may be working in programmer mode.

Also, devices will be represented differently according to the system.

The following is an example of an on-line dialogue that prints today's (T) log:

```
Select System Audit Reports Option: PRINT <Enter> Sign-on Log
START WITH DATE/TIME: FIRST// T
GO TO DATE/TIME: LAST// <Enter>
DEVICE: PRINTER-1
```

Figure 4-2. Print Sign-on Log option—Sample user dialogue

USERS WHO HAVE SIGNED ONTO THE COMPUTER				NOV 23,1991 10:15 PAGE 1	
SIGNED ON LOCATION	ELAPSED TIME (MINUTES)	USER	DEVICE		
NOV 23,1991 06:20 IRM OFFICE	13	KRNUSERM,THIRTEEN	_LTA1903:-ROU		
NOV 23,1991 07:27 SYSTEMS OFC		KRNUSERN,FOURTEEN	_LTA1921:-ROU		
NOV 23,1991 08:33 MAS OFC #6	19	KRNUSERO,FIFTEEN	_LTA1712:-ROU		
NOV 23,1991 09:46 MAS OFC #6	20	KRNUSEMQ,SEVENTEEN	_LTA1872:-ROU		
NOV 23,1991 10:07 SYSTEMS OFC		KRNUSERP,SIXTEEN	_LTA1900:-ROU		

Figure 4-3. Print Sign-on Log option—Sample report

Purge

IRM purges the Sign-on Log. The Purge Sign-on Log option is scheduled by the IRM Service to run on a regular basis. For example, the purge may run each night and delete all entries over 30 days old, retaining an online log of the past 30 days.

Failed Access Attempts

The Failed Access Attempt Log can be used to record information about signons that were attempted but failed. To record information, an audit must be initiated by setting the relevant Kernel site parameters, as shown in Figure 4-4.



NOTE: If an audit has not been initiated, the Audit Display options described on the subsequent pages will show nothing.

A signon attempt fails if the number of permitted attempts is reached and a valid Access and Verify code pair has not been entered. The number of permitted attempts is a Kernel site parameter, usually set between 5 and 10. When the limit is reached, an entry is made in the log (if auditing has been initiated). Information is recorded depending upon whether a valid Access or Verify code was entered during an attempt. If a valid Access code is entered, a user name is associated with the attempt. If requested when establishing the audit parameters, the "text" of any invalid Access or Verify code attempt will be recorded.

The limit of attempts is usually determined by the DEFAULT # OF ATTEMPTS site parameter. If, however, a limit has been set for a particular device, that limit takes precedence for that device. Such preliminary device checking, though, can be bypassed altogether by using another site parameter, BYPASS DEVICE LOCKOUT, which circumvents the locking mechanism.

When an access attempt fails, the settings for lockout times are checked. If a lockout time is operative for the device, the user must wait for the lockout time to elapse before initiating another signon attempt.

Set Parameters

The parameters for the failed access attempt audit are displayed when entering a question mark as shown in Figure 4-4. In this case, all devices will be audited and the text entered for invalid Access or Verify codes will be recorded. Alternatively, audits can be set for specific devices (e.g., modems or Integrated Data Communications Utility [IDCU] ports that support remote access).



REF: The Option Audit is discussed subsequently; it is essentially set to NO AUDIT in (Figure 4-4).

```
Select System Security Option:  AUDIT <Enter> Features

Select Audit Features Option:  MAINTAIN <Enter> System Audit Options

Select Maintain System Audit Options Option:  ESTABLISH <Enter> System Audit
Parameters
Select KERNEL SYSTEM PARAMETERS DOMAIN NAME:  1 <Enter>      VAMC.VA.GOV
INITIATE AUDIT:  T <Enter> (NOV 23, 1991)
TERMINATE AUDIT:  T+7 <Enter> (NOV 30, 1991)
OPTION AUDIT:  // <Enter>
FAILED ACCESS ATTEMPT AUDIT:  ?
      CHOOSE FROM:
      A      ALL DEVICES/NO TEXT RECORDED
      D      SPECIFIED DEVICES/NO TEXT RECORDED
      AR     ALL DEVICES/TEXT RECORDED
      DR     SPECIFIED DEVICES/TEXT RECORDED
      N      NO AUDIT
FAILED ACCESS ATTEMPT AUDIT:  AR <Enter>  ALL DEVICES/TEXT RECORDED
```

Figure 4-4. Establish System Audit Parameters option—Sample user dialogue setting audit parameters

Initiate/Terminate Audits

The audit takes place if both an initiate and terminate date are specified. Auditing will not occur without specification of a date range. When auditing begins, it proceeds as specified by these parameters, including the Option Audit if it has been specified.

Display Parameters

Using the Display the Kernel Audit Parameters option, parameters for the failed access attempt audit can be displayed, as shown below:



NOTE: Option Audits, if any had been specified, would also be displayed.

```
Select System Security Option:  AUDIT <Enter> Features
Select Audit Features Option:  MAIN <Enter> tain System Audit Options
Select Maintain System Audit Options Option:  DISPLAY <Enter> the Kernel Audit
Parameters
SORT BY: NUMBER// <Enter>
        WITHIN NUMBER, SORT BY: <Enter>
DEVICE: <Enter>

List of current Kernel audit parameters
Initiate on: NOV 23,1991          Terminate on:  NOV 30,1991

Failed access attempt audit: ALL DEVICES/TEXT RECORDED
```

Figure 4-5. Display the Kernel Audit Parameters option—Sample user dialogue and report

Print Reports

The Failed Access Attempts Log option displays attempts by date and user, each on a separate page. If a valid Access code is entered, the associated user name will be printed. The page may then be taken to the user for verification of identity. If the named individual did not originate the signon attempt, it can be assumed that the valid Access code has been discovered and a new code should be issued.



NOTE: If the number of attempts shown exceeds the system limit (i.e., default or device-specific), the attempt may not have been initiated by an interactive user. Such an event should be further investigated.

```
Select System Audit Reports Option:  Failed Access Attempts Log
START WITH DATE/TIME OF ATTEMPT: FIRST// <Enter>
        START WITH USER: FIRST// <Enter>
DEVICE:  PRINTER-1
```

Figure 4-6. Failed Access Attempts Log option—Sample user dialogue

```

LOG OF USER FAILED ACCESS LIST  NOV 23,1991      10:07 AM    PAGE  1
-----
*** USER NAME:   KRNUSERN,FOURTEEN

DATE/TIME OF ATTEMPT:  AUG 29,1991  15:07
NUMBER OF ATTEMPTS:    4              TYPE OF FAILED ATTEMPT:  VERIFY
CPU: AAA              UCI:  TST        DEVICE: DEC SERVER (DSV2/LC-3-5)
TEXT ENTERED:
Verify: IDEALS
Verify: JUSTICE

```

Figure 4-7. Failed Access Attempts Log option—Sample report

Using the Device Failed Access Attempts option, failed attempt information can also be displayed for a particular device, as shown in Figure 4-9. Devices with a high number of failed attempts should be monitored. Of particular concern would be remote access devices (e.g., modems or IDCU ports).

```

Select Audit Display Option: Device Failed Access Attempts
DEVICE: PRINTER-1

```

Figure 4-8. Device Failed Access Attempts option—Sample user dialogue

DEVICE FAILED ACCESS ATTEMPTS		OCT 5,1994	PAGE 1
DEVICE	DATE/TIME	NUMBER OF ATTEMPTS	
DEC SERVER	AUG 31,1994 15:48	5	
DEC SERVER	SEP 6,1994 10:56	5	
DEC SERVER	SEP 6,1994 11:00	5	
DEC SERVER	SEP 6,1994 11:07	5	
DEC SERVER	SEP 12,1994 15:54	5	
DEC SERVER	SEP 16,1994 13:38	5	
DEC SERVER	SEP 19,1994 07:29	5	
DEC SERVER	SEP 19,1994 08:07	5	
DEC SERVER	SEP 19,1994 15:09	5	
DEC SERVER	SEP 19,1994 15:49	5	
DEC SERVER	SEP 19,1994 15:51	5	
DEC SERVER	SEP 20,1994 10:23	5	
DEC SERVER	SEP 21,1994 14:40	5	
DEC SERVER	SEP 28,1994 11:47	5	
DEC SERVER	SEP 29,1994 09:03	5	
DEC SERVER	OCT 1,1994 18:00	5	
SUBTOTAL		80	
TOTAL		80	

Figure 4-9. Device Failed Access Attempts option—Sample report

User Failed Access Attempts option, failed access attempts can also be displayed by the user name that is associated with the valid Access code entered during the failed signon attempt.

```
Select Audit Display Option:  User <Enter> Failed Access Attempts
START WITH USER: FIRST// <Enter>
  START WITH DATE/TIME OF ATTEMPT: FIRST// <Enter>
DEVICE:  PRINTER-1
```

Figure 4-10. User Failed Access Attempts option—Sample user dialogue

FAILED ACCESS ATTEMPTS	NOV 23, 1991	09:51	PAGE 1
USER	DATE/TIME	NUMBER	
-----	-----	-----	-----
KRNUSERM, THIRTEEN	AUG 29, 1991 15:25	5	

SUBTOTAL		5	
KRNUSEQ, SEVENTEEN	AUG 31, 1991 09:58	4	

SUBTOTAL		4	
KRNUSERP, SIXTEEN	AUG 29, 1991 15:07	2	

SUBTOTAL		2	

TOTAL		11	

Figure 4-11. User Failed Access Attempts option—Sample report

Purge

Use the Failed Access Attempt Log Purge option to purge the Failed Access Attempts log. This option is located on the System Security menu, as shown below:

```
Select Maintain System Audit Options Option:  FAILED <Enter> Access Attempt Log
Purge

PURGE BEGIN DATE: T-180 <Enter> (SEP 12, 1991)
PURGE END DATE: T-7 <Enter> (MAR 04, 1992)
Requested Start Time: NOW// <Enter> (MAR 11, 1992@10:42)
Request queued
```

Figure 4-12. Failed Access Attempt Log Purge option—Sample user dialogue

Option and Server Usage Audits

There are two files that hold information about the use of options:

- PROGRAMMER MODE LOG file (#3.07)
- AUDIT LOG FOR OPTIONS file (#19.081)

Both of these files are used to store data about when users accessed options.

Programmer Mode Log

All instances of the use of the Programmer Mode menu option are automatically logged by the system. The file is stored in the Manager's account so that UCI switching can be properly monitored. All instances are audited, since this option permits direct system access via the M programming language. Such access is necessary to manage and maintain VistA software, but the use of this option should not go without surveillance. Obviously, entry to programmer mode is reserved for your facility's most trusted users only.

Print Reports

The Display of Programmer Mode Entry List option can be used to show the use of programmer mode by user and by date/time. The list can be limited to individual users. The following example shows a list of all users for today (T):

```
Select System Security Option:  AUDIT <Enter> Features
Select Audit Features Option:   AUDIT <Enter> Display
Select Audit Display Option:    DISPLAY <Enter> of Programmer Mode Entry List
START WITH USER: FIRST// <Enter>
      START WITH DATE/TIME: FIRST// T
      GO TO DATE/TIME: LAST// <Enter>
DEVICE:  PRINTER-1
```

Figure 4-13. Display of Programmer Mode Entry List option—Sample user dialogue

Programmer Mode Entry List		MAR 11, 1992		PAGE 1
DUZ	USER NAME	UCI, VOL	DATE/TIME	

13	KRNUSERM, THIRTEEN	VAH, ROU	MAR 11, 1992	09:38
13	KRNUSERM, THIRTEEN	VAH, ROU	MAR 11, 1992	15:02
13	KRNUSERM, THIRTEEN	VAH, ROU	MAR 11, 1992	15:15
13	KRNUSERM, THIRTEEN	VAH, ROU	MAR 11, 1992	16:46

SUBCOUNT	4			

44	PITKIN, HANNAH	VAH, ROU	MAR 11, 1992	10:22
44	PITKIN, HANNAH	VAH, ROU	MAR 11, 1992	12:35

SUBCOUNT	2			

COUNT	6			

Figure 4-14. Display of Programmer Mode Entry List option—Sample user dialogue

Purge

Although the log of entry into programmer mode is automatically kept, it is not automatically purged. Purging must be done by using the Programmer Mode Entry Log Purge option, as shown below:

```
Select Maintain System Audit Options Option: Programmer <Enter> Mode Entry Log
Purge

PURGE BEGIN DATE: T-10 <Enter> (NOV 13, 1991)
PURGE END DATE: T <Enter> (NOV 23, 1991)
Requested Start Time: NOW// <Enter> (NOV 23, 1991@10:42)
Request queued
```

Figure 4-15. Programmer Mode Entry Log Purge option—Sample user dialogue

Option Audit

Programmer mode is just one of the many VistA options. Its importance in terms of access authority justifies automatic auditing of its use. Other options, however, do not need such constant audit.

Set Parameters

When establishing audit parameters, the range of auditing may extend from all options to just a few options. Auditing all options may be of little benefit, not to mention the massive amounts of data that would accumulate. Auditing all options within a namespace should similarly be used only when warranted (e.g., to further an investigation). If a problem arose with a particular VistA software application, however, an audit by namespace may be appropriate. For example, auditing the SD namespace would audit the use of all Scheduling options, XU would audit Kernel options, and DI would audit VA FileMan options. Or, if concern is with the activity of several VistA users, the audit can be set to record the option access of those users. Finally, a specific set of options may be identified for auditing.

Figure 4-16 sets parameters to audit use of VA FileMan's Modify File Attributes option [DIMODIFY] and Kernel's Device Edit option [XUDEV]. It further indicates that all option use by EIGHTEEN KRNUSERR will be audited, and that all options in the XQSMD namespace, Kernel's Secure Menu Delegation system, will be audited.



NOTE: The failed access attempt audit, previously set, will continue to be in effect for the same time period.



NOTE: There is a limit to the number of namespaces and users that you can audit at any one time. This limit is due to the fact that all namespaces and users are stored in one limited-length string during audits.

Initiate/Terminate Audits

Auditing takes place during the specified time period.



NOTE: Auditing includes the failed access attempt audit set in previously.

When initiating an audit, review all parameters that have been specified. Any that were specified in the past will again be audited unless they are deleted. For example, if a previous audit had specified other options, users, or namespaces, those would need to be deleted if the current audit was not intended to include them. The Establish System Audit Parameters option can be used to delete prior settings by using the at-sign ("@") when prompted with the item to delete.

```

Select System Security Option:  AUDIT <Enter> Features

Select Audit Features Option:  MAINTAIN <Enter> System Audit Options

Select Maintain System Audit Options Option:  ESTABLISH <Enter> System Audit
Parameters
Select KERNEL SYSTEM PARAMETERS DOMAIN NAME:  1 <Enter>    VAMC.VA.GOV
INITIATE AUDIT:  T <Enter> (NOV 23, 1991)
TERMINATE AUDIT:  T+7 <Enter> (NOV 30, 1991)
OPTION AUDIT:  // ?
      CHOOSE FROM:
      n          NO AUDIT
      a          ALL OPTIONS AUDITED
      s          SPECIFIC OPTIONS AUDITED
      u          USERS AUDITED
OPTION AUDIT:  // SPECIFIC OPTIONS AUDITED
Select OPTION TO AUDIT:  DIMODIFY
      ARE YOU ADDING 'DIMODIFY' AS A NEW OPTION TO AUDIT
      (THE 1ST FOR THIS KERNEL SYSTEM PARAMETERS)?  Y <Enter>    (YES)
Select OPTION TO AUDIT:  DEVICE EDIT <Enter>    XUDEV          Device Edit
      ARE YOU ADDING 'XUDEV' AS A NEW OPTION TO AUDIT
      (THE 2ND FOR THIS KERNEL SYSTEM PARAMETERS)?  Y <Enter>    (YES)
Select OPTION TO AUDIT:  <Enter>
Select NAMESPACE TO AUDIT:  ^OPT
      1    OPTION AUDIT
      2    OPTION TO AUDIT
CHOOSE 1-2:  1
OPTION AUDIT:  SPECIFIC OPTIONS AUDITED//  USERS AUDITED
Select USER TO AUDIT:  KRNUSEERR,EIGHTEEN
      ARE YOU ADDING 'KRNUSEERR,EIGHTEEN' AS A NEW USER TO AUDIT
      (THE 1ST FOR THIS KERNEL SYSTEM PARAMETERS)?  Y <Enter>    (YES)
Select USER TO AUDIT:  <Enter>
Select NAMESPACE TO AUDIT:  XQSMD <Enter>
      ARE YOU ADDING 'XQSMD' AS A NEW NAMESPACE TO AUDIT
      (THE 1ST FOR THIS KERNEL SYSTEM PARAMETERS)?  Y <Enter>    (YES)
Select NAMESPACE TO AUDIT:  <Enter>
FAILED ACCESS ATTEMPT AUDIT:  AR// ?
FAILED ACCESS ATTEMPT AUDIT:  AR <Enter>    ALL DEVICES/TEXT RECORDED

```

Figure 4-16. Set Parameters for Audit

Display Parameters

Use the Display the Kernel Audit Parameters option to display the parameters for the Option Audit. Parameters for auditing options as well as parameters for auditing failed attempts are shown in Figure 4-18:

```
Select System Security Option:  AUDIT <Enter> Features
Select Audit Features Option:  MAINTAIN <Enter> System Audit Options
Select Maintain System Audit Options Option: DISPLAY <Enter> the Kernel Audit
Parameters
SORT BY: NUMBER// <Enter>
        WITHIN NUMBER, SORT BY: <Enter>
DEVICE: PRINTER-1
```

Figure 4-17. Display the Kernel Audit Parameters option—Sample user dialogue

```
List of current Kernel audit parameters
Initiate on: NOV 23,1991          Terminate on:  NOV 30, 1991
Option(s) to Audit:  DIMODIFY
Option(s) to Audit:  XUDEV
Namespace(s) to Audit: XQSMD
User to Audit:      KRNUSERR,EIGHTEEN

Failed access attempt audit: ALL DEVICES/TEXT RECORDED
```

Figure 4-18. Display the Kernel Audit Parameters option—Sample report

Print Reports

The Audited Options Log option displays information about the audited options. The option name is shown along with the time of use, user, CPU, device, and job number.

```
Select System Audit Reports Option: AUDITED <Enter> Options Log
START WITH DATE/TIME: FIRST// <Enter>
START WITH OPTION: FIRST// <Enter>
DEVICE: PRINTER-1
```

Figure 4-19. Audited Options Log option—Sample user dialogue

MENU	OPTION	AUDIT LOG	NOV 23,1991	10:12 AM	PAGE 1

***	OPTION:	XUDEV			
	USER:	KRNUSE	R,NINETEEN		
DATE/TIME	(ENTRY):	OCT 28, 1991	19:21	(EXIT):	OCT 28, 1991 19:23
CPU:	BBB	DEVICE:	89	JOB:	7
***	OPTION:	XUDEV			
	USER:	KRNUSE	R,TWENTY		
DATE/TIME	(ENTRY):	DEC 1, 1991	14:29	(EXIT):	DEC 1, 1991 14:35
CPU:	AAA	DEVICE:	89	JOB:	10

Figure 4-20. Audited Options Log option—Sample report

Using the Option Audit Display option, audited options can be displayed sorting by option name and then by date/time. A specific set of options for a particular time period can be shown, as in Figure 4-22.

```
Select Audit Display Option: OPTION <Enter> Audit Display
START WITH OPTION: FIRST// <Enter>
START WITH DATE/TIME: FIRST// <Enter>
DEVICE: PRINTER-1
```

Figure 4-21. Option Audit Display option—Sample user dialogue

MENU	OPTION	AUDIT LIST	NOV 23,1991	PAGE 1
OPTION:	XUDEV		USER:	KRNUSE
ENTRY:	OCT 28, 1991 19:21	EXIT:	OCT 28, 1991 19:23	R,FOURTEEN
OPTION:	XUDEV		USER:	KRNUSE
ENTRY:	NOV 1, 1991 14:29	EXIT:	NOV 1, 1991 14:35	R,THIRTEEN

Figure 4-22. Option Audit Display option—Sample report

Using the User Audit Display option, the audited option log can also be sorted according to users and then by options. The use of options by a particular user can be displayed, as in Figure 4-24.

```
Select Audit Display Option: USER <Enter> Audit Display
START WITH USER: FIRST// KRNUSE
GO TO USER:LAST// KRNUSE
START WITH OPTION: FIRST// <Enter>
DEVICE: PRINTER-1
```

Figure 4-23. User Audit Display option—Sample user dialogue

USER MENU OPTION AUDIT LIST		NOV 23,1991	PAGE 1

USER:	KRNUSEERR,EIGHTEEN	OPTION:	XUDEV
ENTRY:	OCT 28,1991 19:21	EXIT:	OCT 28,1991 19:23
USER:	KRNUSEERR,EIGHTEEN	OPTION:	XUDEV
ENTRY:	NOV 1,1991 14:29	EXIT:	NOV 1,1991 14:35

Figure 4-24. User Audit Display option—Sample report

Purge

Use the Audited Options Purge option to purge the Audited Options Log, as shown below. This log may need regular purging, since it can quickly grow in size with data that is of little significance with respect to auditing.

```
Select Maintain System Audit Options Option: AUDITED <Enter> Options Purge
PURGE BEGIN DATE: T-30 <Enter> (OCT 23, 1991)
PURGE END DATE: T <Enter> (NOV 11, 1991)
Request queued
```

Figure 4-25. Audited Options Purge option—Sample user dialogue

Server Audit

Servers are automated mail readers designed to process incoming mail messages, and (possibly) execute routines and options in response to incoming mail messages. To guard against inappropriate server activity, server-type options and server requests should be reviewed and monitored.

The design of server-type options can be reviewed by IRM Service using Menu Management options. Some of the OPTION file (#19) attributes to note are the routine and entry/exit actions since they will determine how the server will function. Attributes governing when a server can run are the prohibited times restrictions and the Server Action. The Server Action can be set to honor server requests, ignore server requests, or hold the request and simply notify the mail group associated with the server that a request has been received. Once server activity has begun, a bulletin can be sent to alert the mail group associated with the bulletin. Finally, the server audit flag in the OPTION file (#19) can be set for particular server-type options so that an audit trail can be maintained in the AUDIT LOG FOR OPTIONS file (#19.081).

To receive bulletins concerning server activity, the Security Officer will need to contact IRM Service to be added to the mail groups associated with the server bulletins. The default server bulletin is XQSERVER. It will be used unless another bulletin is named in the Server Bulletin field of the OPTION file (#19).

The Information Security Officer may carry out overall audits of server activity by using the option audit techniques described in the previous topic of this chapter. The example provided in this topic illustrates how to include the auditing of servers in the set of Kernel audit parameters.

VA FileMan audits, described later in this manual, can be used to monitor changes to data in the OPTION file (#19) for servers and other options. For example, a data audit could be set for the entry and exit actions to detect any changes.

Set Parameters

All server requests are issued via the Postmaster. The Postmaster exists on each system to manage mail. The Postmaster has a user number of .5 and mail baskets that function as queues for network transmissions. The Postmaster is the "user" of all server requests, so auditing the Postmaster captures information about server activity.

Auditing the XQSRV namespace will similarly monitor server activity. This is a special case of auditing by namespace. It has been designed to facilitate the auditing of server activity. While specifying a namespace ordinarily sets an audit flag for all options that are named in a particular way (beginning with the characters of the namespace), auditing of the XQSRV namespace additionally flags any server-type options, regardless of namespace. It is thus unnecessary to itemize individual server-type options within the multiple of specific options to audit.

The following example illustrates how the Postmaster and the XQSRV namespace can be added to the list of audit parameters to monitor servers:

```
Select System Security Option: AUDIT <Enter> Features

Select Audit Features Option: MAINTAIN <Enter> System Audit Options

Select Maintain System Audit Options Option: ESTABLISH <Enter> System Audit Parameters

Select KERNEL SYSTEM PARAMETERS DOMAIN NAME: 1 <Enter> VAMC.VA.GOV
INITIATE AUDIT: T <Enter> (NOV 23, 1991)
TERMINATE AUDIT: 1/1/92 <Enter> (JAN 1, 1992)
OPTION AUDIT: SPECIFIC OPTIONS AUDITED// USERS AUDITED
Select USER TO AUDIT: KRNUSERR,EIGHTEEN// .5 <Enter> POSTMASTER
ARE YOU ADDING 'POSTMASTER' AS A NEW USER TO AUDIT
(THE 2ND FOR THIS KERNEL SYSTEM PARAMETERS)? Y <Enter> (YES)
Select USER TO AUDIT: <Enter>
Select NAMESPACE TO AUDIT: XQSM// XQSRV
ARE YOU ADDING 'XQSRV' AS A NEW NAMESPACE TO AUDIT
(THE 2ND FOR THIS KERNEL SYSTEM PARAMETERS)? Y <Enter> (YES)
Select NAMESPACE TO AUDIT: <Enter>
FAILED ACCESS ATTEMPT AUDIT: AR// <Enter>
```

Figure 4-26. Adding the Postmaster and the XQSRV namespace to the list of audit parameters to monitor servers

Initiate/Terminate Audits

As stated before, all identified users, options, and namespaces will be audited within the designated time frame. The option to display the Kernel audit parameters can always be used to show what has been specified for auditing. To turn off auditing, it is not enough to simply set the OPTION AUDIT FLAG to No Audit. *All* items within the user, option, and namespace multiples *must* be explicitly deleted one by one.

Display Parameters

Use the Display the Kernel Audit Parameters option to display the parameters. This display of Kernel audit parameters indicates that server activity will be audited since the Postmaster is audited and the XQSRV namespace is audited. All other operative parameters are also shown:

```
Select System Security Option: AUDIT <Enter> Features
Select Audit Features Option: MAINTAIN <Enter> System Audit Options
Select Maintain System Audit Options Option: DISPLAY <Enter> the Kernel Audit
Parameters
SORT BY: NUMBER// <Enter>
        WITHIN NUMBER, SORT BY: <Enter>
DEVICE: PRINTER-1
```

Figure 4-27. Display the Kernel Audit Parameters option—Sample user dialogue

```
List of current Kernel audit parameters
Initiate on: NOV 23,1991          Terminate on:  JAN 1, 1992
Option(s) to Audit:  DIMODIFY
Option(s) to Audit:  XUDEV
Namespace(s) to Audit: XQSM
Namespace(s) to Audit: XQSRV
User to Audit:      KRNUSER, EIGHTEEN
User to Audit:      POSTMASTER

Failed access attempt audit: ALL DEVICES/TEXT RECORDED
```

Figure 4-28. Display the Kernel Audit Parameters option—Sample report

Print Reports

The Server Audit Display option lists the audit trail for servers. It shows the option name, the user and sender names, the entry and exit times, and the number and subject of the mail message. In addition, the action taken when the server request was received is shown as a comment. The comment indicates whether the server ran normally or if any errors occurred during the process.

```
Select System Security Option: AUDIT <Enter> Features
Select Audit Features Option: SYSTEM <Enter> Audit Reports
Select Audit Display Option:  SERVER <Enter> Audit Display
DEVICE: PRINTER-1
```

Figure 4-29. Server Audit Display option—Sample user dialogue

```

                                SERVER OPTION REQUESTS    FEB 26, 1992    17:40

SERVER OPTION:      ZZSRVTST                                USER:  POSTMASTER
ENTRY:      JAN 22, 1992    14:03                        EXIT:   JAN 22, 1992    14:03
MESSAGE #:      2922                                      SENDER: PROGRAMMER, CHIEF
SUBJECT:  TEST OF SERVERS

COMMENTS:

Routine ^ZZSRVTST ended normally
```

Figure 4-30. Server Audit Display option—Sample report

Purge

Using the Audited Options Purge option, the audit trail for all options including servers will be purged within the specified time range:

```
Select Maintain System Audit Options Option: AUDITED <Enter> Options Purge
PURGE BEGIN DATE: 1/1/92 <Enter> (JAN 1, 1992)
PURGE END DATE: T <Enter> (FEB 26, 1992)
Requested Start Time: NOW// <Enter> (FEB 26, 1992@18:42)
Request queued
```

Figure 4-31. Audited Options Purge option—Sample user dialogue

VA FileMan Audits

VA FileMan has methods of auditing data values as well as the structural changes to the data dictionaries themselves. A data audit records changes made to the data on file (e.g., a change in a patient's social security number), see the example in the *VA FileMan Advanced User Manual*. A data dictionary (DD) audit, on the other hand, monitors alterations in the data attributes (the definitions of the fields of the file).



REF: For more information and instructions on VA FileMan data audits and data dictionary audits, please refer to the Auditing chapter of the *VA FileMan Advanced User Manual*.

5. Software Integrity

Several mechanisms exist to ensure the integrity of the programs of the VistA system. This chapter discusses four tools that can be used to check the integrity of programs:

- Program Integrity Checker Option
- Verify Program Integrity Option
- Checking Programs Received via Network Mail/PackMan
- Checking Secured Programs Received via Network Mail/PackMan

The developers who create the programs that make up the VistA system have given special consideration to the security needed by the individual software (e.g., Pharmacy, Laboratory) as well as those of the system at large. The programs written for these software applications strictly control the data entered and reviewed by users. In many cases, the programs even mark data with a user's system identification (e.g., DUZ) so users can determine who entered or changed the data.

But what safeguards are there to be sure no one changes the programs themselves? The first safeguard is obvious. No user should be allowed to enter programmer mode on your system who is not a trusted employee, skilled in the M programming language. On a secure system, the only way such users can enter programmer mode is from the Programmer Mode menu option. Also, security keys *must* be assigned to such a user to use the Programmer Mode option. And, as stated earlier, every time a user does enter programmer mode through the Programmer Mode option, the system updates the log for that event.

To monitor the integrity of VistA software applications, all VistA software is complemented with a Program Integrity Checker. A program integrity checker is basically a table of values with a numeric entry for each program that belongs to a software application. The numeric value for a program is the sum of the ASCII values of the characters in that routine. This is an example of a checksum. After the integrity table is built for a software application, then those sums can be included with the software. You can compare the sums for components on the current system with the software's original checksums at any time.

For software created before the advent of the Kernel Installation and Distribution System (KIDS), use the Program Integrity Checker option to verify software integrity. For software distributed with the KIDS process, use the Verify Package Integrity option to verify software integrity.

You should not be overly concerned if the integrity checker for a software application reports a discrepancy (the sums do not match). This does show that a program has been altered. However, usually, the most likely source for the change is that IRM has installed a patch, or correction to the software. IRM updates to a software application since the original installation are one cause of mismatches in the checksums.

Program Integrity Checker Option

The Program Integrity Checker option can be used to run a software-specific integrity checker. It can check software integrity for software distributed before the advent of the Kernel Installation and Distribution System (KIDS).



REF: For information on how to check software integrity for software distributed with KIDS, please refer to the "Verify Program Integrity Option" topic that follows.

With the Program Integrity Checker option, you are prompted for a namespace (e.g., XU for Kernel). Kernel then appends the letters NTEG (e.g., XUNTEG). If a program is found that matches the constructed name, that integrity checker is run. The output for a software-specific integrity checker lists the routines included in the software and indicates if the checksum on record matches that of the routine in its current state. Variances between the record and the current routine are flagged on the display.

The following example shows a subset of the integrity checker for Kernel during its development (hence the checker had not been updated to account for the changes made to routine XQ6):

```
Select System Security Option: Program Integrity Checker

Select PACKAGE PREFIX: XU

DEVICE: HOME// <Enter>

Running ^XUNTEG...

Checksum routine created on MAR 03, 1992@11:40:57

XQ          ok
XQ1         ok
XQ11        ok
XQ12        ok
XQ1V5       ok
XQ6         Calculated 9993246, off by -5328572
XQ9         ok
XQ91        ok
```

Figure 5-1. Program Integrity Checker option—Sample user dialogue and report

The IRM Service ordinarily runs the Program Integrity Checker option each time they load new programs onto the VistA computer system. This is done to ensure that the newly received programs have not been damaged. This is a wise precaution.

Verify Program Integrity Option

You can use the Verify Package Integrity option to compare checksums of software components against the checksums of the components when they were originally transported. It works only for software that was distributed using KIDS. For software distributed before the advent of KIDS, use the Program Integrity Checker option to check software integrity.

Any discrepancies are reported. Currently, routines are the only components that are checked, but checksums will be extended to other software components in the future.

The checksums of components for the currently installed software are verified against checksums stored in the BUILD file (#9.6) entry for the software. If the most recent version of the BUILD file (#9.6) entry for a software application has been purged, the Verify Package Integrity option will no longer be able to verify checksums for the loaded software. Because of this, the most recent build entry for a software application should not be purged in most cases.



REF: For more information on KIDS, please refer to the "KIDS" section of the *Kernel Systems Management Guide*.

```
Select Utilities Option: Verify Package Integrity
Select BUILD NAME: KERNEL 8.0
DEVICE: HOME// <Enter>

PACKAGE: KERNEL 8.0           Feb 05, 1995 10:02 am           Page 1
-----

    758 Routine checked, 0 failed.

Select Utilities Option:
```

Figure 5-2. Verify Package Integrity option—Sample user dialogue and report

Checking Programs Received via Network Mail/PackMan

IRM receives new programs via MailMan. Usually, these MailMan messages originate from another computer, most commonly one of the computers at an Office of Information field Office (OIFO). These MailMan messages are received and read just like the conversational messages you routinely exchange. But, instead of regular language, they contain M programs. These are commonly referred to as PackMan messages.

IRM Service uses PackMan messages to update VistA computer system programs. It is a fast and efficient method to install new programs without the risk of typographical errors. PackMan provides options to:

- Construct messages.
- Install messages.
- Compare messages with programs already installed on your VistA system.

This manual does not discuss the options to construct or install messages in this manual. Instead, it concentrates on those options to compare PackMan messages with resident programs.

Why would you want to compare a PackMan message with a resident program? Simply, PackMan offers users another tool to determine if a program on the computer has been altered. If you save the PackMan message, you can compare its contents with your resident programs at any time. So, users always have a ready check against the altering of sensitive programs.

The following figures work through an example with the user FOUR KRNUSERD. He will begin by reading his mail and noting that he has three messages in his IN basket. One of these messages, "DEMO PACKAGE FOR MANUAL" contains a PackMan message (Figure 5-3). After he selects that message, MailMan begins to display its contents (Figure 5-4). As you can see, this is not a regular conversational message. Instead, its contents seem somewhat cryptic. The message begins with a program, called A6SDIOO1.

```
Select System Security Option: MAILMAN <Enter> Menu
MailMan 7.1 service for KRNUSERD,FOUR at VAMC.VA.GOV
You last used MailMan: 18 Apr 92 17:51

Select MailMan Menu Option: READ <Enter> a message
Read MAIL BASKET: IN// <Enter>
LAST Message Number: 3   Messages in BASKET: 3

IN Basket Message: 1// ?

*=NEW          ##### Subject #####          ### From ###
  1. A6SDINIT - Security Demo                KRNUSERD,FOUR
  2. INSTALLATION OF DHCP PACKAGE            KRNUSERD,FOUR
  3. DEMO PACKAGE FOR MANUAL                  KRNUSERA,ONE A.
Enter '?HELP' or '???' to see all the other exciting things you can do !

IN Basket Message: 1// 3
```

Figure 5-3. Sample mailbox entries, including a sample PackMan message


```

Subj: PACKAGE FOR SECURITY DEMO ON 4/14/92  11 Apr 92 10:13  337 Lines
From: KRUSERA, ONE A.          in 'IN' basket.
-----
$ROU A6SDI001
A6SDI001 ;
;;Version 1
F I=1:2 S X=$T(Q+I) Q:X="" S Y=$E($T(Q+I+1),4,999),X=$E(X,4,999) S:$A(Y)=126
I=I+1,Y=$E(Y,2,999)_$E($T(Q+I+1),5,99) S:$A(Y)=61 Y=$E(Y,2,999) X NO E S @X=Y
Q Q
;;^DIC(16014,0,"GL")
;;^DIZ(16014,
;;^DIC("B","SECURITY OFFICER LIST",16014)
;;=
;;^DD(16014,0)
;;=FIELD^^5^6
;;^DD(16014,0,"ID",2)
;;S %I=Y,Y=$S('$D(^ (0)):',"',$D(^DIC(4,+SP(^ (0),U,3),0))#2:SP(^ (0),U,1),1:""),
C=$P(^DD(4,.01,0),U,2) D Y^DIQ:Y]"" W " " ,Y,@("$E("_DIC_"%I,0),0)") S Y=%I
K %I

Press RETURN to continue or '^' to exit: ^<Enter>

```

Figure 5-4. Sample PackMan message contents

At this point, the user enters a caret ("^") to tell MailMan to stop displaying the message (Figure 5-4). MailMan prompts the user for an action. In this example, the user enters "X," which activates PackMan (Figure 5-5).



NOTE: PackMan is reserved for use by very privileged users only. The user has a File Manager Access code of "@"; therefore, he can use PackMan.

First, the message is summarized (i.e., the contents are itemized).

```

Select MESSAGE Action: IGNORE (in IN basket)// X
Select PackMan function: SUM <Enter> MARIZE MESSAGE
Line 1      Routine ROU A6SDI001
Line 57     Routine ROU A6SDI002
Line 109    Routine ROU A6SDI003
Line 135    Routine ROU A6SDI004
Line 181    Routine ROU A6SDINI1
Line 220    Routine ROU A6SDINI2
Line 235    Routine ROU A6SDINI3
Line 279    Routine ROU A6SDINIT
Line 320    Routine ROU A6SDM
Line 330    Routine ROU A6SDP

```

Figure 5-5. Activating PackMan

Now that the user is certain about the content of the PackMan message, he compares the message contents to the programs that reside on his VistA computer system. Using PackMan, the Compare option is activated. PackMan now compares the message, line by line, with programs of the same name. Any discrepancies are displayed.

In the following example (Figure 5-7), the program A6SDM does *not* match. The actual detail shown by PackMan may be difficult for a non-programmer to decipher, but you should discuss such reports with your IRM Service. In all likelihood, the change to the program is appropriate and does not constitute a security violation.

```
Select PackMan function: COMPARE MESSAGE
DEVICE: HOME// PRINTER-1
```

Figure 5-6. Comparing PackMan message contents with programs on a VistA system

```
Line 1    Comparing Routine  ROU A6SDI001
-----
Line 57   Comparing Routine  ROU A6SDI002
-----
Line 109  Comparing Routine  ROU A6SDI003
-----
Line 135  Comparing Routine  ROU A6SDI004
-----
Line 181  Comparing Routine  ROU A6SDINI1
-----
Line 220  Comparing Routine  ROU A6SDINI2
-----
Line 235  Comparing Routine  ROU A6SDINI3
-----
Line 279  Comparing Routine  ROU A6SDINIT
-----
Line 320  Comparing Routine  ROU A6SDM
1{ ;A6SDM ;ISC - SEND MSG BACK TO }      1{ ;A6SDM ;ISC - SEND MSG BACK TO }
{SOURCE TO SIGNIFY INSTALL DONE ;}      {SOURCE TO SIGNIFY INSTALL DONE ;}
{4/18/95  17:56}                        {4/7/95 13:25}
  ^                                     ^
3{ S A6SD(1,0)="Security Demonstra}      3{ S A6SD(1,0)="Security Demo Pack}
      ^                                     ^
      {te on Package Installed"}          {age Installed"}
-----
Line 330  Comparing Routine  ROU A6SDP
-----
Select PackMan function: <Enter>

Select MESSAGE Action: IGNORE (in IN basket)// <Enter> Ignored.
```

Figure 5-7. PackMan and program comparison report

Checking Secured Programs Received via Network Mail/PackMan

VA FileMan allows a user to create a PackMan message containing the programs that make up a software application. Under this option, instead of creating programs that are saved on the computer disk, the routines are written directly into a MailMan message (i.e., no routines are written to the disk). The programmer provides a Scramble Hint by which the message is encrypted. When the message is received, the recipient can use the Scramble Hint to decode the software. If the message is intact (i.e., it has not been tampered with), the software can then be installed with the PackMan utilities. The message can also be saved to compare against the routines on the disk at future dates much the same way in which an integrity checker or standard PackMan message is used.

The only major difference with secured messages is that you must first provide the correct scramble password before you can do anything with the message.

In the following example, the user looks at a message titled "A6SDINIT - Security Demo". He uses MailMan as before, enters a password, and proceeds as if this were a standard PackMan message.

```
Select Systems Security Menu Option: MAILMAN <Enter> Menu
MailMan 7.0 service for KRUSERA,ONE at KERNEL.REDACTED
You last used MailMan: 14 Apr 92 17:39

Select MailMan Menu Option: READ <Enter> a message
Read MAIL BASKET: IN// <Enter>
LAST Message Number: 5    Messages in BASKET: 3

IN Basket Message: 1// ?
*=NEW          ##### Subject #####          ### From ###
  1. ADP SECURITY MEETING                    KRUSERZ,TWENTY-SIX
  2. INSTALLATION OF DHCP PACKAGE            KRUSERA,ONE
  5. A6SDINIT - Security Demo                 KRUSERD,FOUR
Enter '?HELP' or '???' to see all the other exciting things you can do !

IN Basket Message: 1// 5
```

Figure 5-8. Sample mailbox entries, including a sample encrypted PackMan message

```

This text was scrambled with the scramble hint: 'DEMO '
Enter scramble password: <TYPE IN THE PASSWORD HERE>

Subj: A6SDINIT - Security Demo 13 Apr 89 11:21 513 Lines
From: KRNUSERD,FOUR in 'IN' basket.
-----
$TXT
$ROU ^A6SDM
A6SDM ;SFISC - SEND MSG BACK TO SOURCE TO SIGNIFY INSTALL DONE
;4/7/89 13:25
;V1
S A6SD(1,0)="Security Demo Package Installed"
S XMY(DUZ)="",XMY("G.A6SD INSTALL")=""
S XMSUB="INSTALLATION OF DHCP PACKAGE: SECURITY OFFICER LIST"
S XMTEXT="A6SD(" N DIFROM D ENX^XMD
K XMSUB,XMTEXT,XMY
Q
$END ROU A6SDM
$ROU ^A6SDP
A6SDP ;SFISC/CNP - PRINT LIST OF SECURITY OFFICERS ;4/5/89 18:55
;V1
Press RETURN to continue or '^' to exit:

```

Figure 5-9. Sample encrypted PackMan message contents

At this point, The user enters a caret ("^") to tell MailMan to stop displaying the message. MailMan prompts the user for an action. As in the earlier example, he enters "X" (Figure 5-10). This activates PackMan



NOTE: PackMan is reserved for use by very privileged users only. The user has a File Manager Access code of "@"; therefore, he can use PackMan.

The user simply compares the message contents to the programs that reside on his VistA computer system. Using PackMan, the Compare option is activated. As before, PackMan compares the message, line by line, with programs of the same name. This time, the programs saved in the PackMan message match those installed on our VistA computer system.

```

Select MESSAGE Action: IGNORE (in IN basket)// X

Select PackMan function: ?
ANSWER WITH PackMan function NUMBER, OR NAME
CHOOSE FROM:
1          ROUTINE LOAD
2
GLOBAL LOAD
3          PACKAGE LOAD
4          SUMMARIZE MESSAGE
5          PRINT MESSAGE
6          INSTALL MESSAGE
7          COMPARE MESSAGE

Select PackMan function: COMPARE MESSAGE <Enter>
DEVICE: HOME// <Enter> DECSEVER

```

**Typing an X
activates PackMan**

Figure 5-10. Activating PackMan

```

Line 2    Comparing Routine  ROU ^A6SDM
-----
Line 12   Comparing Routine  ROU ^A6SDP
-----
Line 20   Comparing Data Dictionary  DDD ^SECURITY OFFICER LIST
Line 78   Comparing Data Dictionary  DDD ^JOB SERIES
Line 120  Comparing FileMan Data  DTA ^JOB SERIES
Line 142  Comparing Bulletins  BUL ^
Line 156  Comparing Input Templates  DIE ^
Line 162  Comparing Print Templates  DIP ^
Line 184  Comparing Security keys  KEY ^
Line 192  Comparing Options  OPT ^
Line 270  Comparing Package File  PKG ^
Line 374  Comparing Routine  ROU ^A6SDINI1
-----
Line 407  Comparing Routine  ROU ^A6SDINI2
-----
Line 419  Comparing Routine  ROU ^A6SDINI3
-----
Line 460  Comparing Routine  ROU ^A6SDINIT
-----
Line 499  Comparing Routine  ROU ^A6SDNTEG
-----

```

Figure 5-11. PackMan and program comparison report

Glossary

AUDIT ACCESS	A user's authorization to mark the information stored in a computer file to be audited.
AUDITING	Monitoring computer usage such as changes to the database and other user activity. Audit data can be logged in a number of VA FileMan and Kernel files.
AUTO MENU	An indication to Menu Manager that the current user's menu items should be displayed automatically. When AUTO MENU is not in effect, the user must enter a question mark at the menu's select prompt to see the list of menu items.
CAPACITY MANAGEMENT	The process of assessing a system's capacity and evaluating its efficiency relative to workload in an attempt to optimize system performance. Kernel provides several utilities.
CARET	A symbol expressed as ^ (caret). In many M systems, a caret is used as an exiting tool from an option. Also, this symbol is sometimes referred to as the up-arrow symbol.
CHECKSUM	A numeric value that is the result of a mathematical computation involving the characters of a routine or file.
CIPHER	A system that arbitrarily represents each character as one or more other characters. (See also: ENCRYPTION.)
COMMON MENU	Options that are available to all users. Entering two question marks ("??") at the menu's select prompt will display any SECONDARY MENU OPTIONS available to the signed-on user along with the common options available to all users.
COMPILED MENU SYSTEM (^XUTL GLOBAL)	Job-specific information that is kept on each CPU so that it is readily available during the user's session. It is stored in the ^XUTL global, which is maintained by the menu system to hold commonly referenced information. The user's place within the menu trees is stored, for example, to enable navigation via menu jumping.
COMPUTED FIELD	This field takes data from other fields and performs a predetermined mathematical function (e.g., adding two columns together). You will not, however, see the results of the mathematical function on the screen. Only when you are printing or displaying information on the screen will you see the results for this type of field.

DEVICE HANDLER	The Kernel module that provides a mechanism for accessing peripherals and using them in controlled ways (e.g., user access to printers or other output devices).
DIFROM	VA FileMan utility that gathers all software components and changes them into routines (namespaceI* routines) so that they can be exported and installed in another VA FileMan environment.
DOUBLE QUOTE (")	A symbol used in front of a Common option's menu text or synonym to select it from the Common menu. For example, the five character string "TBOX selects the User's Toolbox Common option.
DR STRING	The set of characters used to define the variable DR when calling VA FileMan. Since a series of parameters may be included within quotes as a literal string, the variable's definition is often called the DR string. To define the fields within an edit sequence, for example, the programmer may specify the fields using a DR string rather than an INPUT template.
DUZ(0)	A local variable that holds the FILE MANAGER ACCESS CODE of the signed-on user.
ENCRYPTION	Scrambling data or messages with a cipher or code so that they are unreadable without a secret key. In some cases encryption algorithms are one directional, that is, they only encode and the resulting data cannot be unscrambled (e.g., Access and Verify codes).
FILE ACCESS SECURITY SYSTEM	Formerly known as Part 3 of the Kernel Inits. If the File Access Security conversion has been run, file-level security for VA FileMan files is controlled by Kernel's File Access Security system, not by File Manager Access codes (i.e., FILE MANAGER ACCESS CODE field).
FORCED QUEUING	A device attribute indicating that the device can only accept queued tasks. If a job is sent for foreground processing, the device will reject it and prompt the user to queue the task instead.
GO-HOME JUMP	A menu jump that returns the user to the primary menu presented at signon. It is specified by entering two carets ("^^") at the menu's select prompt. It resembles the Rubber-band Jump but without an option specification after the carets.
HELP PROCESSOR	A Kernel module that provides a system for creating and displaying online documentation. It is integrated within the menu system so that help frames associated with options can be displayed with a standard query at the menu's select prompt.
HOST FILE SERVER (HFS)	A procedure available on layered systems whereby a file on the host system can be identified to receive output. It is implemented by the Device Handler's HFS device type.

HUNT GROUP	An attribute of an entry in the DEVICE file (#3.5) that allows several devices to be used interchangeably; useful for sending network mail or printing reports. If the first hunt group member is busy, another member can stand in as a substitute.
INDEX (%INDEX)	A Kernel utility used to verify routines and other M code associated with a software application. Checking is done according to current ANSI MUMPS standards and VistA programming standards. This tool can be invoked through an option or from direct mode (>D ^%INDEX).
INIT	Initialization of a software application. INIT* routines are built by VA FileMan's DIFROM and, when run, recreate a set of files and other software components.
JUMP	In VistA applications, the Jump command allows you to go from a particular field within an option to another field within that same option. You can also Jump from one menu option to another menu option without having to respond to all the prompts in between. To jump, type a caret ("^", uppercase-6 key on most keyboards) and then type the name of the field or option you wish to jump to. (See also GO-HOME JUMP, PHANTOM JUMP, RUBBER-BAND JUMP, or UP-ARROW JUMP.)
JUMP START	A logon procedure whereby the user enters the "Access code;Verify code;option" to go immediately to the target option, indicated by its menu text or synonym. The jump syntax can be used to reach an option within the menu trees by entering "Access;Verify;^option".
KERMIT	A standard file transfer protocol. It is supported by Kernel and can be set up as an alternate editor.
MANAGER ACCOUNT	A UCI that can be referenced by non-manager accounts (e.g., production accounts). Like a library, the MGR UCI holds percent routines and globals (e.g., ^%ZOSF) for shared use by other UCIs.
MENU CYCLE	The process of first visiting a menu option by picking it from a menu's list of choices and then returning to the menu's select prompt. Menu Manager keeps track of information (e.g., the user's place in the menu trees) according to the completion of a cycle through the menu system.
MENU MANAGER	The Kernel module that controls the presentation of user activities (e.g., menu choices or options). Information about each user's menu choices is stored in the Compiled Menu System, the ^XUTL global, for easy and efficient access.
MENU SYSTEM	The overall Menu Manager logic as it functions within the Kernel framework.

MENU TEMPLATE	An association of options as pathway specifications to reach one or more final destination options. The final options must be executable activities and not merely menus for the template to function. Any user can define user-specific MENU templates via the corresponding Common option.
MENU TREES	The menu system's hierarchical tree-like structures that can be traversed or navigated, like pathways, to give users easy access to various options.
PAC	Programmer Access Code. An optional user attribute that can function as a second level password into programmer mode.
PART 3 OF THE KERNEL INIT	See FILE ACCESS SECURITY SYSTEM.
PATTERN MATCH	A preset formula used to test strings of data. Refer to your system's M Language Manuals for information on Pattern Match operations.
PHANTOM JUMP	Menu jumping in the background. Used by the menu system to check menu pathway restrictions.
PRIMARY MENUS	The list of options presented at signon. Each user must have a PRIMARY MENU OPTION in order to sign on and reach Menu Manager. Users are given primary menus by IRM. This menu should include most of the computing activities the user will need.
PROGRAMMER ACCESS	Privilege to become a programmer on the system and work outside many of the security controls of Kernel. Accessing programmer mode from Kernel's menus requires having the programmer's at-sign security code, which sets the variable DUZ(0)=@.
PROTOCOL	An entry in the PROTOCOL file (#101). Used by the Order Entry/Results Reporting (OE/RR) software to support the ordering of medical tests and other activities. Kernel includes several protocol-type options for enhanced menu displays within the OE/RR software.
QUEUEING	Requesting that a job be processed in the background rather than in the foreground within the current session. Kernel's TaskMan module handles the queueing of tasks.
QUEUEING REQUIRED	An option attribute that specifies that the option must be processed by TaskMan (the option can only be queued). The option can be invoked and the job prepared for processing, but the output can only be generated during the specified time periods.
RESOURCE	A method that enables sequential processing of tasks. The processing is accomplished with a RES device type designed by the application programmer and implemented by IRM. The process is controlled via the RESOURCE file (#3.54).

RUBBER-BAND JUMP	<p>A menu jump used to go out to an option and then return, in a bouncing motion. The syntax of the jump is two carets ("^^", uppercase-6 on most keyboards) followed by an option's menu text or synonym (e.g., ^^Print Option File). If the two carets are not followed by an option specification, the user is returned to the primary menu.</p> <p>(See also: GO-HOME JUMP.)</p>
SCHEDULING OPTIONS	<p>A way of ordering TaskMan to run an option at a designated time with a specified rescheduling frequency (e.g., once per week).</p>
SCROLL/NO SCROLL	<p>The Scroll/No Scroll button (also called Hold Screen) allows the user to "stop" (No Scroll) the terminal screen when large amounts of data are displayed too fast to read and "restart" (Scroll) when the user wishes to continue.</p>
SECONDARY MENU OPTIONS	<p>Options assigned to individual users to tailor their menu choices. If a user needs a few options in addition to those available on the primary menu, the options can be assigned as secondary options. To facilitate menu jumping, secondary menus should be specific activities, not elaborate and deep menu trees.</p>
SECURE MENU DELEGATION (SMD)	<p>A controlled system whereby menus and keys can be allocated by people other than IRM staff (e.g., application coordinators) who have been so authorized. SMD is a part of Menu Manager.</p>
SERVER	<p>In VistA, an entry in the OPTION file (#19). An automated mail protocol that is activated by sending a message to the server with the "S.server" syntax. A server's activity is specified in the OPTION file (#19) and can be the running of a routine or the placement of data into a file.</p>
SIGNON/SECURITY	<p>The Kernel module that regulates access to the menu system. It performs a number of checks to determine whether access can be permitted at a particular time. A log of signons is maintained.</p>
SPECIAL QUEUING	<p>An option attribute indicating that TaskMan should automatically run the option whenever the system reboots.</p>
SPOOLER	<p>An entry in the DEVICE file (#3.5). It uses the associated operating system's spool facility, whether it's a global, device, or host file. Kernel manages spooling so that the underlying OS mechanism is transparent. In any environment, the same method can be used to send output to the spooler. Kernel will subsequently transfer the text to a global for subsequent despooling (printing).</p>
SYNONYM	<p>In VistA, a field in the OPTION file (#19). Options can be selected by their menu text or synonym.</p> <p>(See also: MENU TEXT.)</p>

TASKMAN	The Kernel module that schedules and processes background tasks (also called Task Manager).
TIMED READ	The amount of time Kernel will wait for a user response to an interactive READ command before starting to halt the process.
UP-ARROW JUMP	In the menu system, entering a carets ("^", uppercase-6 on most keyboards) followed by an option name accomplishes a jump to the target option without needing to take the usual steps through the menu pathway.
Z EDITOR (^%Z)	A Kernel tool used to edit routines or globals. It can be invoked with an option, or from direct mode after loading a routine with >X ^%Z.
ZOSF GLOBAL (^%ZOSF)	The Operating System File—a manager account global distributed with Kernel to provide an interface between VistA software and the underlying operating system. This global is built during Kernel installation when running the manager setup routine (ZTMGRSET). The nodes of the global are filled-in with operating system-specific code to enable interaction with the operating system. Nodes in the ^%ZOSF global can be referenced by application programmers so that separate versions of the software need not be written for each operating system.



REF: For a comprehensive list of commonly used infrastructure- and security-related terms and definitions, please visit the ISS Glossary Web page at the following Web address:

REDACTED

For a comprehensive list of acronyms, please visit the ISS Acronyms Web site at the following Web address:

REDACTED

Appendix A—VistA Security Forms

When a user is granted an account on a VistA system, two forms are prepared by Kernel to meet system security policies and procedures:

- User Account Notification
- Computer Account Access Policy

Upon creation of a new user account, the Kernel will automatically print these forms for the user(s).

The Medical Information Security Service in the Medical Information Resources Management Office (MIRMO) has reviewed these documents and their suggested wording is shown below. Your facility can customize these documents to meet your local needs. The documents are stored as Help Frames and can be edited via the Help Frame menu in Kernel.

USER ACCOUNT NOTIFICATION

Department of Veterans Affairs

Your VA Facility
123 Any Address
Anytown, State, Zip

A user account has been created in your name to enable you to access on-line clinical and/or administrative data required to perform your duties as an employee of the Department of Veterans Affairs. Please read the enclosed NEW USER INFORMATION before you attempt your first log-on to the system. Questions about access should be referred to the AIS Application Coordinator in your service, your facility Information Security Officer (ISO), or your IRM Service.

Your Computer Access Coordinator is:

Your Facility Information Security Officer:

Your Alternate Information Security Officer:

Figure A-1. User Account Notification message

COMPUTER ACCOUNT ACCESS POLICY

Department of Veterans Affairs

Your VA Facility

As an authorized user of VHA automated information systems (AISs) and having access to data stored in them, I will be given sufficient access to perform my assigned duties. I will use this access ONLY for its intended purpose and understand the following policies that apply to VA data and computer systems:

I agree to safeguard all passwords (e.g., Access/Verify codes, electronic signature codes) assigned to me and am strictly prohibited from disclosing these codes to anyone including family, friends, fellow workers, supervisor(s), and subordinates for ANY reason.

I understand that I may be held accountable for all entries/changes made to any government AIS using my passwords.

I am aware of the regulations and facility AIS security policies designed to ensure the confidentiality of all sensitive information. I am aware that information about patients or employees is confidential and protected from unauthorized disclosure by law. I understand that my obligation to protect VA information does not end with either the termination of my access to this facility's systems or with the termination of my government employment.

I will exercise common sense and good judgment in the use of electronic mail. I understand that electronic mail is not inherently confidential and I have no expectation of privacy in using it. I understand that technical or administrative problems may create situations which requires viewing of my messages. I also understand that facility management officials may authorize access to my electronic mail messages whenever there is a legitimate purpose for such access.

I understand that a violation of this notice constitutes disregard of a local and/or VHA policy and will result in appropriate disciplinary action as defined in VA employee conduct Regulations (VAR 820(b)) as well as suspension/termination of access privileges.

I affirm with my signature that I have read, understand, and agree to fulfill the provisions of this User Access notice.

Signature: _____

Figure A-2. Computer Account Access Policy message

Index

A

- Abbreviated Menu Diagrams Option, 3-3
- Access Codes, 2-1, 2-4, 2-5
 - Format, 2-4
 - Number of Attempts, 2-1
 - User Advice, 2-8
- Access to VA FileMan Files Option, 2-12
- ACCESSIBLE FILE Multiple Field, 2-12
- Acronyms (ISS)
 - Home Page Web Address, Glossary, 6
- Adobe Acrobat Quick Guide Web Address, xiv
- Appendix A—VistA Security Forms, 1
- Assumptions About the Reader, xiii
- AUDIT LOG FOR OPTIONS File (#19.081), 4-9, 4-15
- Audited Options Log Option, 4-13
- Audited Options Purge Option, 4-14, 4-18
- Audits
 - Audited Options Log Option, 4-13
 - Display of Programmer Mode Entry List Option, 4-9
 - Display the Kernel Audit Parameters Option, 4-6, 4-12
 - Establish System Audit Parameters Option, 4-5, 4-10
 - Failed Access Attempt Log Purge Option, 4-8
 - Failed Access Attempts Log, 4-4, 4-8
 - Failed Access Attempts Log Option, 4-6
 - Failed Access Audit, 2-2
 - Features, 4-1
 - Initiate/Terminate, 4-5, 4-11, 4-16
 - Initiating, 4-2
 - IRM's Responsibility, 4-1
 - Namespaced Options, 4-10
 - Old Access and Verify Codes, 4-2
 - Print Sign-on Log Option, 4-3
 - Programmer Mode Entry Log Purge Option, 4-10
 - Purge Log of Old Access and Verify Codes Option, 4-3
 - Purge Sign-on Log Option, 4-4
 - Servers, 4-15
 - Sign-on Log, 4-3
 - System Access Audits, 4-2

- User Failed Access Attempts Option, 4-8
- VA FileMan, 4-19

B

- BUILD file (#9.6), 5-3
- Bulletins
 - XQSERVER, 4-15
- BYPASS DEVICE LOCKOUT Field, 4-4

C

- Callout Boxes, xi
- Checking
 - Programs Received via Network
 - Mail/PackMan, 5-4
 - Secured Programs Received via Network
 - Mail/PackMan, 5-7
- Checksums, 5-1, 5-2, 5-3
- Codes
 - Access, 2-1, 2-4, 2-5
 - Security, 2-4
 - Verify, 2-1, 2-4, 2-5
- Contents, v

D

- Data Dictionary
 - Data Dictionary Utilities Menu, xii
 - Listings, xii
- DEFAULT # OF ATTEMPTS Field, 4-4
- Device Check, 2-1
- Devices
 - locked, 2-1
 - Locked, 2-6
- Diagram Menus Option, 3-3
- Display of Programmer Mode Entry List Option, 4-9
- Display Parameters, 4-16
- Display the Kernel Audit Parameters Option, 4-6, 4-12, 4-16
- Documentation
 - History, iii
 - Symbols, xi

E

- Edit User Characteristics Menu, 2-4
- Electronic Signatures, 2-6
 - Code, 2-6
- Establish System Audit Parameters Option, 4-11
- Establishing Kernel Audit Parameters Option, 2-2
- EVS Anonymous Directories, xiv
- Examining Menus and Options, 3-1

F

- Failed Access Attempt Log Purge Option, 4-8
- Failed Access Attempts Log, 4-4, 4-8
- Failed Access Attempts Log Option, 4-6
- Failed Access Audit, 2-2
- Fields
 - ACCESSIBLE FILE Multiple, 2-12
 - BYPASS DEVICE LOCKOUT, 4-4
 - DEFAULT # OF ATTEMPTS, 4-4
 - OPTION AUDIT FLAG, 4-16
- Figures and Tables, vii
- Files
 - AUDIT LOG FOR OPTIONS (#19.081), 4-9, 4-15
 - BUILD (#9.6), 5-3
 - KERNEL SYSTEM PARAMETERS (#8989.3), 2-4, 2-6
 - NEW PERSON (#200), 2-7, 2-12
 - OPTION (#19), 3-3, 4-15
 - PROGRAMMER MODE LOG (#3.07), 4-2, 4-9
- Find a User Option, 2-11
- Forms
 - VistA Security Clearance, 1

G

- General Information About Users, 2-9
- Glossary, 1
 - ISS Home Page Web Address, Glossary, 6

H

- Has the File Access Security System been Enabled?, 2-12
- Help
 - At Prompts, xii
 - Online, xii
 - Question Marks, xii

- History, Revisions to Documentation and Patches, iii
- Home Pages
 - Adobe Acrobat Quick Guide Web Address, xiv
 - Adobe Web Address, xiv
 - Health Systems Design and Development Web Address, xiii
- ISS
 - Acronyms Home Page Web Address, Glossary, 6
 - Glossary Home Page Web Address, Glossary, 6
 - Kernel Home Page Web Address, xiii
 - VistA Documentation Library (VDL) Home Page Web Address, xiv
- How to
 - Obtain Technical Information Online, xii
 - Use this Manual, x

I

- If the File Access Security System Has Been Enabled, 2-12
- If the File Access Security System Has Not Been Enabled, 2-12
- Initiating Audits, 4-2, 4-5, 4-11, 4-16
- Inquire Option, 3-2
- Inquiry to a User's File Access Option, 2-13
- Introduction, 1-1
- IRM's Responsibility, 4-1
- ISS
 - Acronyms Home Page Web Address, Glossary, 6
 - Glossary Home Page Web Address, Glossary, 6

K

- Kernel
 - Audit Features, 4-1
 - Home Page Web Address, xiii
 - Security Codes, 2-4
 - Security During User Sessions, 2-1
- KERNEL SYSTEM PARAMETERS File (#8989.3), 2-4, 2-6
- Keys
 - Security, 2-2, 3-1

L

- List Access to Files by File number Option, 2-14
- List Delegated Options and their Users Option, 3-4
- List File Attributes Option, xii
- List Users Option, 2-9
- Locked Devices, 2-1, 2-6
- Logs
 - Failed Access Attempts Log, 4-4, 4-8
 - Old Access and Verify Codes, 4-2
 - Programmer Mode Log, 4-9
 - PROGRAMMER MODE LOG File (#3.07), 4-2
 - Sign-on Log, 4-2, 4-3

M

- Manage User File Menu, 4-3
- Management
 - Software, 1-2
- Menu Management
 - Diagram Menus Option, 3-3
 - Inquire Option, 3-2
 - Option Access by User Option, 3-1
 - Print Options File Option, 3-3
- Menu Manager Security, 3-1
- Menus, 2-2, 3-1
 - Data Dictionary Utilities, xii
 - Edit User Characteristics, 2-4
 - Manage User File, 4-3
 - Primary, 2-2, 2-8, 2-9, 3-3
 - Programmer Mode, 4-9, 5-1
 - Review Users, 2-12
 - Security, 3-1
 - System Security, 4-8
 - System Security Options, 2-2
 - Toolbox, 2-4
- Messages
 - PackMan, 5-4

N

- NEW PERSON File (#200), 2-7, 2-12

O

- Obtaining
 - Data Dictionary Listings, xii
- Old Access and Verify Codes, 4-2
- Online
 - Documentation, xii

- Technical Information, How to Obtain, xii
- Option Access By User Option, 3-1
- Option Audit Display Option, 4-13
- OPTION AUDIT FLAG Field, 4-16
- OPTION File (#19), 3-3, 4-15
- Options, 2-2, 3-1
 - Abbreviated Menu Diagrams, 3-3
 - Access to VA FileMan Files, 2-12
 - Audited Options Log, 4-13
 - Audited Options Purge, 4-14, 4-18
 - Data Dictionary Utilities, xii
 - Diagram Menus, 3-3
 - Display of Programmer Mode Entry List, 4-9
 - Display the Kernel Audit Parameters, 4-6, 4-12, 4-16
 - Edit User Characteristics, 2-4
 - Establish System Audit Parameters, 4-11
 - Establishing Kernel Audit Parameters, 2-2
 - Failed Access Attempt Log Purge, 4-8
 - Failed Access Attempts Log, 4-6
 - Find a User, 2-11
 - Inquire, 3-2
 - Inquiry to a User's File Access, 2-13
 - List Access to Files by File number, 2-14
 - List Delegated Options and their Users Option, 3-4
 - List File Attributes, xii
 - List Users, 2-9
 - Manage User File, 4-3
 - Option Access By User, 3-1
 - Option Audit, 4-10
 - Option Audit Display, 4-13
 - Print All Delegates and their Options Option, 3-5
 - Print Options File Option, 3-3
 - Print Sign-on Log, 4-3
 - Print Users Files, 2-14
 - Program Integrity Checker, 1-1, 5-1, 5-2, 5-3
 - Programmer Mode, 4-9, 5-1
 - Programmer Mode Entry Log Purge, 4-10
 - Purge Log of Old Access and Verify Codes, 4-3
 - Purge Sign-on Log, 4-4
 - Review Users, 2-12
 - Server Audit Displayn, 4-17
 - Show a Delegate's Options, 3-4
 - System Security, 4-8
 - System Security Options, 2-2
 - Toolbox, 2-4
 - Usage Audits, 4-9
 - User Audit Display, 4-14

- User Failed Access Attempts, 4-8
- User Inquiry, 2-10
- User Status Report, 2-11
- Verify Package Integrity, 5-1
- Verify Program Integrity, 5-3
- Orientation For Kernel Developer's Guide, x

P

- PackMan
 - Compare Option, 5-5
 - Summarize Option, 5-5
 - VA FileMan, 5-7
- PackMan Messages, 5-4
- Parameters
 - Setting, 4-15
- Parameters
 - BYPASS DEVICE LOCKOUT Field, 4-4
 - DEFAULT # OF ATTEMPTS Field, 4-4
 - Setting, 4-5, 4-10
- Parameters
 - Display, 4-16
- Patches
 - History, iv
- Primary Menu, 2-2, 2-8, 2-9, 3-3, 2, 5
- Print All Delegates and their Options Option, 3-5
- Print Options File Option, 3-3
- Print Reports, 4-3, 4-17
- Print Sign-on Log Option, 4-3
- Print Users Files Option, 2-14
- Program Integrity Checker Option, 1-1, 5-1, 5-2, 5-3
- Programmer Mode Entry Log Purge Option, 4-10
- Programmer Mode Log, 4-9
- PROGRAMMER MODE LOG File (#3.07), 4-2, 4-9
- Programmer Mode Menu, 4-9, 5-1
- Purge
 - Audited Options Purge Option, 4-14, 4-18
 - Purge Log of Old Access and Verify Codes, 4-3
 - Purge Log of Old Access and Verify Codes Option, 4-3
- Purge Log of Old Access and Verify Codes Option, 4-3
- Purge Sign-on Log Option, 4-4

Q

- Question Mark Help, xii

R

- Reader, Assumptions About the, xiii
- Reference Materials, xiii
- Review Users Menu, 2-12
- Reviewing Users, 2-8
- Revision History, iii
 - Documentation, iii
 - Patches, iv

S

- SAC, xiv
- Secure Menu Delegation, 3-4
 - List Delegated Options and their Users Option, 3-4
 - Print All Delegates and their Options Option, 3-5
 - Show a Delegate's Options Option, 3-4
- Security
 - Codes, 2-4
 - During User Sessions, 2-1
 - Finding Users Online, 2-11
 - Forms, 1
 - Keys, 2-2, 3-1
 - List Access to Files by File number Option, 2-14
 - Status Report, 2-11
 - User, 2-1
 - User Inquiry Option, 2-10
- Server Audit Display Option, 4-17
- Servers
 - Audit, 4-15
 - Usage Audits, 4-9
- Set Parameters, 4-5, 4-10, 4-15
- Show a Delegate's Options Option, 3-4
- Sign-on Log, 4-2, 4-3
- Software
 - Integrity, 5-1
 - Checksums, 5-1, 5-2, 5-3
 - MailMan, 5-4
 - PackMan
 - Compare, 5-5
 - Patches, 5-1
 - Program Integrity Checker Option, 5-1, 5-2
 - Programs, 5-1
 - VA FileMan, 5-7
 - Verify Program Integrity Option, 5-3

- Management, 1-2
- Symbols
 - Found in the Documentation, xi
- System Access Audits, 4-2
- System Security Menu, 4-8
- System Security Options Menu, 2-2

T

- Terminating Audits, 4-5, 4-11, 4-16
- Tips and General Advice for Users, 2-8
- Toolbox Menu, 2-4
- Trusted Facility Manual, x

U

- URLs
 - Adobe Acrobat Quick Guide Web Address, xiv
 - Adobe Home Page Web Address, xiv
 - Health Systems Design and Development Home Page Web Address, xiii
- ISS
 - Acronyms Home Page Web Address, Glossary, 6
 - Glossary Home Page Web Address, Glossary, 6
 - Kernel Home Page Web Address, xiii
 - VistA Documentation Library (VDL) Home Page Web Address, xiv
- Use this Manual, How to, x
- User Audit Display Option, 4-14
- User Failed Access Attempts Option, 4-8
- User Identification, 2-1
- User Inquiry Option, 2-10
- User Security, 2-1
 - Finding Users Online, 2-11
 - List Access to Files by File number Option, 2-14
 - Listing, 2-9

- Status Report, 2-11
- User Inquiry Option, 2-10
- User Status Report Option, 2-11
- User's Access to VA FileMan Files, 2-11

V

- VA FileMan
 - Audits, 4-19
 - File Security, 2-2
 - Secured Messages, 5-7
- Verify Codes, 2-1, 2-4, 2-5
 - Format, 2-4
 - User Advice, 2-8
- Verify Package Integrity Option, 5-1
- Verify Program Integrity Option, 5-3
- VistA Documentation Library (VDL) Home Page Web Address, xiv

W

- Web Pages
 - Adobe Acrobat Quick Guide Web Address, xiv
 - Adobe Home Page Web Address, xiv
 - Health Systems Design and Development Home Page Web Address, xiii
- ISS
 - Acronyms Home Page Web Address, Glossary, 6
 - Glossary Home Page Web Address, Glossary, 6
 - Kernel Home Page Web Address, xiii
 - VistA Documentation Library (VDL) Home Page Web Address, xiv

X

- XQSERVER Bulletin, 4-15

