IBM

eserver

iSeries

Tips and Tools for Securing Your iSeries

Version 5

SC41-5300-06

IBM

@server

iSeries

Tips and Tools for Securing Your iSeries

Version 5

SC41-5300-06

Note

Before using this information and the product it supports, be sure to read the information in the Security Basic articles found on-line in the Information Center. The Internet URL address is http://www.ibm.com/eserver/iseries/infocenter.

Seventh Edition (August 2002)

| This edition replaces SC41-5300-05. This edition applies only to V4R5 and subsequent versions of OS/400®.

© Copyright International Business Machines Corporation 1996, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 1996, 2002

Figures

© Copyright IBM Corp. 1996, 2002

Tables

© Copyright IBM Corp. 1996, 2002

About Tips and Tools for Securing your iSeries (SC41-5300-06)

The role of computers in organizations is changing rapidly. IT managers, software providers, security administrators, and auditors need to take a new look at many areas that they have taken for granted in the past. iSeries security should be on that list.

Systems are providing many new functions that are vastly different from traditional accounting applications. Users are entering systems in new ways: LANs, switched lines (dial-up), wireless, networks of all types. Often, users never see a sign-on display. Many organizations are expanding to become an "extended enterprise", either with proprietary networks or with the Internet.

Suddenly, systems seem to have a whole new set of doors and windows. Systems managers and security administrators are justifiably concerned about how to protect information assets in this rapidly changing environment.

This information provides a set of practical suggestions for using the security features of iSeries and for establishing operating procedures that are security-conscious. The recommendations in this information apply to an installation with average security requirements and exposures. This information does not provide a complete description of the available iSeries security features. If you want to read about additional options or you need more complete background information, consult the publications that are described in Chapter 18, "Related information" on page 163.

This information also describes how to set up and use security tools that are part of OS/400. Chapter 5, "Configure the iSeries to use Security Tools" on page 33 and "Commands and menus for security commands" on page 34 provide reference information about the security tools. This information provides examples for using the tools.

Who should read this book

A **security officer** or **security administrator** is responsible for the security on a system. That responsibility usually includes the following tasks:

- Setting up and managing user profiles
- Setting system-wide values that affect security
- Administering the authority to objects
- Enforcing and monitoring the security policies

If you are responsible for security administration for one or more iSeries systems, this information is for you. The instructions in this information assume the following:

- You are familiar with basic iSeries operating procedures, such as signing on and using commands.
- You are familiar with the basic elements of iSeries security: security levels, security system values, user profiles, and object security.

© Copyright IBM Corp. 1996, 2002

Note: Chapter 2, "Basic elements of iSeries security" on page 7 provides a review of these elements. If these basic elements are new to you, then read the *Basic security and planning* topic in the iSeries Information Center. See "Prerequisite and related information" for more details.

- You have activated security on your system by setting the security level (QSECURITY) system value to at least 30.

IBM® continually enhances the security capabilities of iSeries. To take advantage of these enhancements, you should regularly evaluate the cumulative fix package that is currently available for your release. See if it contains fixes that are relevant to security.

How to use this information

If you have not set up your system to use the security tools or if you had the Security ToolKit for OS/400 installed for an earlier release, do the following:

1. Start with Chapter 3, "iSeries Security Wizard and Security Advisor" on page 15. It describes how to use these features to select which security tools are recommended and how to get started with them.
2. For more basic security information, you can review the Security Reference information, located on-line in the iSeries™ Information Center.

Note

This information has *many* tips for securing iSeries. Your system may only need protection in some areas. Use this information to educate yourself on possible security exposures and their remedies. Then focus your efforts on the areas that are most critical for your system.

Prerequisite and related information

Use the iSeries Information Center as your starting point for looking up iSeries technical information.

You can access the Information Center two ways:

- From the following Web site:

  `http://www.ibm.com/eserver/iseries/infocenter`
- From CD-ROMs that ship with your Operating System/400® order:

  iSeries Information Center, SK3T-4091-02. This package also includes the PDF versions of iSeries Information Center manuals, *iSeries Information Center: Supplemental Manuals*, SK3T-4092-01, which replaces the Softcopy Library CD-ROM.

The iSeries Information Center contains advisors and topics such as Java™, TCP/IP, Web serving, secured networks, logical partitions, clustering, CL commands, and system application programming interfaces (APIs). It also includes links to related IBM Redbooks™ and Internet links to other IBM Web sites such as the Technical Studio and the IBM home page.

With every new hardware order, you receive the *iSeries Setup and Operations CD-ROM*, SK3T-4098-01. This CD-ROM contains IBM iSeries Access for Windows® and the EZ-Setup wizard. IBM iSeries Access for Windows offers a powerful set of

client and server capabilities for connecting PCs to iSeries servers. The EZ-Setup wizard automates many of the iSeries setup tasks.

iSeries Navigator

IBM iSeries Navigator is a powerful graphical interface for managing your iSeries servers. iSeries Navigator functionality includes system navigation, configuration, planning capabilities, and online help to guide you through your tasks. iSeries Navigator makes operation and administration of the server easier and more productive and is the only user interface to the new, advanced features of the OS/400 operating system. It also includes Management Central function for managing multiple servers from a central system.

You can find more information on iSeries Navigator in the iSeries Information Center and at the following Web site:

`http://www.ibm.com/eserver/iseries/navigator/`

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other iSeries documentation, fill out the readers' comment form at the back of this book.

- If you prefer to send comments by mail, use the readers' comment form with the address that is printed on the back. If you are mailing a readers' comment form from a country other than the United States, you can give the form to the local IBM branch office or IBM representative for postage-paid mailing.
- If you prefer to send comments by FAX, use either of the following numbers:
  - United States, Canada, and Puerto Rico: 1-800-937-3430
  - Other countries: 1-507-253-5192
- If you prefer to send comments electronically, use one of these e-mail addresses:
  - Comments on books:

    RCHCLERK@us.ibm.com
  - Comments on the iSeries Information Center:

    RCHINFOC@us.ibm.com

Be sure to include the following:

- The name of the book or iSeries Information Center topic.
- The publication number of a book.
- The page number or topic of a book to which your comment applies.

Part 1. What's new for V5R2

This is the seventh edition of this information. This edition supports the versions V4R5 and subsequent versions of OS/400. New iSeries security features are highlighted in this section. Minor technical and wording changes have been made throughout this book. A vertical line (|) to the left of the text indicates a substantial change or addition.

© Copyright IBM Corp. 1996, 2002

Chapter 1. iSeries security enhancements

The following security enhancements are now available on the iSeries server:

1. **Single sign-on enablement:**To enable a single sign-on environment, IBM provides two technologies that work together to allow users to sign in with their Windows username and password and be authenticated to iSeries systems in the network:
   a. **Enterprise Identity Mapping (EIM):** EIM provides the mechanics for cross-platform single sign-on enablement. EIM provides a mechanism for associating Kerberos principals to a single EIM identifier that represents that user within the entire enterprise. Other user identities, such as an OS/400 username, can also be associated with this EIM identifier.
   b. **Network Authentication Service (NAS):** NAS allows an iSeries system that is being used in the capacity of centralized server, or a key distribution center, to authenticate principals (Kerberos users) to the network.
2. **Independent auxiliary storage pool (ASP):** Also referred to as an independent disk pool, an ASP is a collection of disk units that can be brought online or taken offline independent of the rest of the storage on a system, including the system disk pool, basic user disk pools, and other independent disk pools.
3. **CHGSYSVAL CL command:** This command restricts the changing of security-related system values, and the addition of digital certificates to a digital certificate store using the Add Verifier API and restricting digital certificate stores from having their passwords reset. See Chapter 3 of the IBM Security Reference for more information on using this command.
4. **Cryptographic hardware:** IBM now offers two cryptographic hardware solutions for customers that require a high level of security for data stored on their iSeries, and for SSL transactions. You can use the new IBM 2058 Cryptographic Accelerator to improve iSeries server performance by rerouting the processing of private keys away from the system processors. This hardware option is an excellent choice for iSeries implementations that handle high volumes of Secure Sockets Layer (SSL) transactions. Also, there are two new capabilities for the 4758 Cryptographic Coprocessor: unique key per transaction (UKPT) for financial pin processing and Common Cryptographic Architecture (CCA) 2.4.

© Copyright IBM Corp. 1996, 2002

Part 2. Basic iSeries security

© Copyright IBM Corp. 1996, 2002

Chapter 2. Basic elements of iSeries security

This topic provides a brief review of the basic elements that work together to provide iSeries security. In other parts of this book we go beyond the basics to provide tips for using these security elements to meet the needs of your organization.

Security levels

You can choose how much security you want the system to enforce by setting the security level (QSECURITY) system value. The system offers five levels of security:

Level 10:

The system does not enforce any security. No password is necessary. If the specified user profile does not exist on the system when someone signs on, the system creates one.

ATTENTION:

Beginning in V4R3 and future releases, you cannot set the QSECURITY system value to 10. If your system is currently at security level 10, it will remain at level 10 when you install Version 4 Release 3. If you change the security level to some other value, you cannot change it back to level 10. Because level 10 provides no security protection, security level 10 is not recommended by IBM. **IBM will not provide support for any problems that occur at security level 10 unless the problem can also be created at a higher security level.**

Level 20:

The system requires a user ID and password for signing on. Security level 20 is often referred to as **sign-on security**.By default, all users have access to all objects because all users have *ALLOBJ special authority.

Level 30:

The system requires a user ID and password for signing on. Users must have authority to use objects because users do not have any authority by default. This is called **resource security**.

Level 40:

The system requires a user ID and password for signing on. In addition to resource security, the system provides **integrity protection**functions. Integrity protection functions, such as the validation of parameters for interfaces to the operating system, are intended to protect both your system and the objects on your system from tampering by experienced system users. For most installations, level 40 is the recommended security level. When you receive a new iSeries system with V4R5 or later release, the security level is set to 40.

Level 50:

The system requires a user ID and password for signing on. The system enforces both resource security and the integrity protection of level 40,but adds **enhanced integrity protection**, such as the restriction of

© Copyright IBM Corp. 1996, 2002

message-handling between system state programs and user state programs. Security level 50 is intended for iSeries systems with high security requirements.

Note: Level 50 is the required level for C2 certification (and FIPS-140 certification).

Chapter 2 of the *iSeries Security Reference* book provides more information about the security levels and describes how to move from one security level to another.

Global settings

Your system has global settings that affect how work enters the system and how the system appears to other system users. These settings include the following:

Security system values:
Security system values are used to control security on your system. These values are broken into four groups:
- General security system values
- Other system values related to security
- System values that control passwords
- System values that control auditing

Several topics in this book discuss the security implications of specific system values. Chapter 3 in the *iSeries Security Reference* book describes all the security-relevant system values.

Network attributes:
Network attributes control how your system participates (or chooses not to participate) in a network with other systems. You can read more about network attributes in the *Work Management* book.

Subsystem descriptions and other work management elements:
Work management elements determine how work enters the system and what environment the work runs in. Several topics in this information discuss the security implications of some work management values. The *Work Management* book provides complete information.

Communications configuration:
Your communications configuration also affects how work enters your system. Several topics in this information provide suggestions for protecting your system when it participates in a network.

User profiles

Every system user **must** have a user profile. You must create a user profile before a user can sign on.User profiles can also be used to control access to service tools such as DASD and main storage dumps. See “Manage service tools” on page 66 for more information.

The user profile is a powerful and flexible tool. It controls what the user can do and customizes the way the system appears to the user. The *iSeries Security Reference* book describes all the parameters in the user profile.

Group profiles

A group profile is a special type of user profile. You can use a group profile to define authority for a group of users, rather than giving authority to each user individually. You can also use a group profile as a pattern when you create individual user profiles by using the copy-profile function or if you use iSeries Navigator you can use the security policies menu to edit user authorities.

Chapter 5 and Chapter 7 in the *iSeries Security Reference* book provide more information about planning and using group profiles.

Resource security

Resource security on the system allows you to define who can use objects and how those objects can be used. The ability to access an object is called **authority**. When you set up object authority, you can need to be careful to give your users enough authority to do their work without giving them the authority to browse and change the system. Object authority gives permissions to the user for a specific object and can specify what the user is allowed to do with the object. An object resource can be limited through specific detailed user authorities, such as adding records or changing records. System resources can be used to give the user access to specific system-defined subsets of authorities: *ALL, *CHANGE, *USE, and *EXCLUDE.

Files, programs, libraries, and directories are the most common system objects that require resource security protection, but you can specify authority for any individual object on the system.

Chapter 6, “Protect information assets with object authority” discusses the importance of setting up object authority on your system. Chapter 5 of the *iSeries Security Reference* book describes the options for setting up resource security.

Limit access to program function

The limit access to program function allows you to provide security for the program when you do not have an iSeries object to secure for the program. Before the limit access to program function support was added in V4R3, you could accomplish this by creating an authorization list or other object, and checking the authority to the object to control access to the program function. Now you can use the limit access to program function to more easily control access to an application, parts of an application, or functions within a program.

There are two methods that you can use to manage user access to application functions through iSeries Navigator. The first uses Application Administration support:

1. Right-click the system that contains the function whose access setting you want to change.
2. Select **Application Administration**.
3. If you are on an administration system, select **Local Settings**. Otherwise, continue with the next step.
4. Select an administrable function.
5. Select **Default Access**, if applicable. By selecting this, you allow all users to access the function by default.

6. Select **All Object Access**, if applicable. By selecting this, you allow all users with all object system privelege to access this function.
7. Select **Customize**, if applicable. use the **Add** and **Remove** buttons on the **Customize Access** dialog to add or remove users or groups in the **Access allowed** and **Access denied** lists.
8. Select **Remove Customization**, if applicable. By selecting this, you delete any customized access for the selected function.
9. Click **OK** to close the **Application Administration** dialog.

The second method of managing user access involves iSeries Navigator's Users and Groups support:

1. In iSeries Navigator, expand **Users and Groups**.
2. Select **All Users**, **Groups**, or **Users Not in a Group** to display a list of users and groups.
3. Right-click a user or group, and select **Properties**.
4. Click **Capabilities**.
5. Click the **Applications** tab.
6. Use this page to change the access setting for a user or group.
7. Click **OK** twice to close the **Properties** dialog.

See "iSeries Navigator security" on page 152 for more information on iSeries Navigator security issues.

If you are an application writer, you can use limit access to program function APIs to do the following:

- Register a function
- Retrieve information about the function
- Define who can or cannot use the function
- Check to see if the user is allowed to use the function

Note: This support is **not** a replacement for resource security. Limit access to program function does not prevent a user from accessing a resource (such as a file or program) from another interface.

To use this support within an application, the application provider must register the functions when the application is installed. The registered function corresponds to a code block for specific functions in the application. When the application is run by the user, the application calls the API before the application calls the code block. The API calls the check usage API to see if the user is allowed to use the function. If the user is allowed to use the registered function, the code block is run. If the user is not allowed to use the function, the user is prevented from running the code block.

Note: API's involve registering a 30 character function ID in the registration data base (WRKREGINF). Although there are no exit points related to function IDs used by the limit access to function APIs, it is required to have exit points. To register anything in the registry, you **must** supply an exit point format name. To do this the Register Function API creates a dummy format name and uses this dummy format name for all functions that are registered. Because this is a dummy format name, no exit point program is ever called.

The system administrator specifies who is allowed or denied access to a function. The administrator can either use the API to manage the access to program function or use the iSeries Navigator Application Administration GUI. The *iSeries server API Reference* book provides information about the limit access to program function API's. For additional information about controlling access to functions, see "iSeries Navigator security" on page 152.

Security audits

People audit their system security for several reasons:

- To evaluate whether the security plan is complete.
- To make sure that the planned security controls are in place and working. This type of auditing is usually performed by the security officer as part of daily security administration. It is also performed, sometimes in greater detail, as part of a periodic security review by internal or external auditors.
- To make sure that system security is keeping pace with changes to the system environment. Some examples of changes that affect security are:
  - New objects created by system users
  - New users admitted to the system
  - Change of object ownership (authorization not adjusted)
  - Change of responsibilities (user group changed)
  - Temporary authority (not timely revoked)
  - New products installed
- To prepare for a future event, such as installing a new application, moving to a higher security level, or setting up a communications network.

The techniques described here are appropriate for all these situations. Which things you audit and how often depends on the size and security needs of your organization.

Security auditing involves using commands on your system and accessing log and journal information. You can create a special profile to be used by someone doing a security audit of your system. The auditor profile needs *AUDIT special authority to change the audit characteristics of the system. Some of the auditing tasks suggested in this chapter require a user profile with *ALLOBJ and *SECADM special authority. Set the password for the auditor profile to *NONE when the audit period has ended.

For more details on security auditing see Chapter 9, of the *Security Reference* book.

Example: System security attributes report

Figure 1 on page 12 shows an example of the output from the Print System Security Attributes (PRTSYSSECA) command. The report shows the settings for security-relevant system values and network attributes that are recommended for systems with normal security requirements. It also shows the current settings on your system.

Note: The *Current® Value* column on the report shows the current setting on your system. Compare this to the recommended value to see where you may have security exposures.

```
System Security Attributes

System Value
Name            Current value          Recommended value
QALWOBJRST      *NONE                  *NONE
QALWUSRDMN      *ALL                   QTEMP
QATNPGM         QEZMAIN   QSYS         *NONE
QAUDENDACN      *NOTIFY                *NOTIFY
QAUDFRCLVL      *SYS                   *SYS
QAUDCTL         *AUDLVL                *AUDLVL   *OBJAUD
QAUDLVL         *SECURITY              *AUTFAIL  *CREATE
                                       *DELETE   *SECURITY
                                       *SAVRST   *NOQTEMP
```

Figure 1. System Security Attributes Report-Sample (Part 1 of 4)

```
QAUTOCFG        0                      0
QAUTORMT        1                      0
QAUTOVRT        9999                   0
QCMNRCYLMT      0 0                    0 0
QCRTAUT         *CHANGE                Control at library level.
QCRTOBJAUD      *NONE                  Control at library level.
QDEVRCYACN      *DSCMSG                *DSCMSG
QDSCJOBITV      120                    120
QDSPSGNINF      1                      1
QINACTITV       60                     60
QINACTMSGQ      *ENDJOB                *ENDJOB
QLMTDEVSSN      0                      1
QLMTSECOFR      0                      1
QMAXSGNACN      2                      3
QMAXSIGN        3                      3
```

Figure 1. System Security Attributes Report-Sample (Part 2 of 4)

```
QPWDEXPITV      60                     60
QPWDLMTAJC      1                      1
QPWDLMTCHR      *NONE                  AEIOU@$#
QPWDLMTREP      1                      2
QPWDLVL         0
QPWDMAXLEN      8                      8
QPWDMINLEN      6                      6
QPWDPOSDIF      1                      1
QPWDRQDDGT      1                      1
QPWDRQDDIF      0                      1
QPWDVLDPGM      *NONE                  *NONE
QRETSVRSEC      0                      0
QRMTIPL         0                      0
QRMTSIGN        *FRCSIGNON             *FRCSIGNON
QSECURITY       50                     50
QSHRMEMCTL      1                      0
QSRVDMP         *DMPUSRJOB             *NONE
QUSEADPAUT      *NONE                  CRTAUTL AUTL(QUSEADPAUT) AUT(*EXCLUDE)
                                       CHGOBJOWN OBJ(QUSEADPAUT) OBJTYPE(*AUTL)
                                       CHGSYSVAL SYSVAL(QUSEADPAUT) VALUE(QUSEADPAUT)
QVFYOBJRST      1                      3
```

Figure 1. System Security Attributes Report-Sample (Part 3 of 4)

```
System Security Attributes

Network Attribute
Name            Current value          Recommended value
DDMACC          *OBJAUT                *REJECT
JOBACN          *FILE                  *REJECT
PCSACC          *OBJAUT                *REJECT
```

Figure 1. System Security Attributes Report-Sample (Part 4 of 4)

Chapter 3. iSeries Security Wizard and Security Advisor

iSeries server Security Wizard and Security Advisor tools can help you decide what security values to put into effect on your iSeries server. Using the iSeries server Security Wizard in iSeries Navigator you will produce reports that reflect your security needs based on your selected answers. You can then use this to configure your system security.

Use the iSeries Security Wizard or the iSeries Security Advisor to help you plan for and implement a basic security policy for your iSeries servers. The goal of both tools is to make it easier for you to implement and manage security on your systems. The wizard, which is available as part of OS/400, asks you several high-level questions about your server environment, and based on your answers, provides you with a set of recommendations that the wizard can apply to your system right away.

The Security Advisor is the on-line version of the Security Wizard. Located in our Technical Studio, it allows you to select your choices based on your security needs and then gives you a report suggesting what features are needed to secure your site.

The iSeries Security Advisor is a Web-based version of the wizard. It provides recommendations for implementing security on your system, just as the wizard does. However, the advisoer can not apply the recommendations. Rather, it outputs a list of system security values and other attributes that you should apply on your system, based on your answers to the advisor's questions. You can access the iSeries Security Advisor from the Technical Studio home page: http://www.iseries.ibm.com/tstudio/secure1/advisor/secwiz.htm

Security Wizard

Deciding which iSeries security system values you should use for your business can be perplexing. If you are new to security implementation on iSeries servers, or the environment in which you run your iSeries server has recently changed, the Security Wizard can help you with decisions.

What is a wizard?

- A wizard is a tool designed to be run by a novice user to install or configure something on a system.
- The wizard prompts the user for information by asking questions. The response to each question determines what question is asked next.
- When the wizard has asked all the questions, the user is presented with a finish dialog. The user then pushes the **Finish** button to install and configure the item.

Security Wizard goals

The goal of the Security Wizard is to configure, based on a user's responses the following.

- Security related system values and network attributes
- Security related reporting for monitoring the system.
- To generate an Administrator Information Report and a User Information Report:

© Copyright IBM Corp. 1996, 2002

  - The Administrator Information Report contains recommended security settings and any procedures that should be followed prior to putting the recommendations into effect.
  - The User Information Report contains information that can be used for the business security policy. For example, password compositions rules are included in this report.
- To provide recommended settings for various security-related items on the system.

Security Wizard objectives

- The objectives of the Security Wizard are:
  - To determine what the system security settings should be, based on the users answers to the wizard's questions, then implement the settings when appropriate.
  - The wizard produces detailed information reports including the following.
    - Report explaining the Wizard's recommendations.
    - Report detailing the procedures that should be followed before implementation.
    - Report listing relevant information to be distributed to the users of the system.
- These items put basic security policy into effect on your system.
- The wizard recommends audit journal reports that you should schedule to run periodically. When scheduled, these reports help:
  - Ensure that security policies are followed.
  - Ensure that security policies are only changed with your approval.
  - Schedule reports to monitor security-related events on your system.
- The wizard allows you to save the recommendations or to apply some or all of the recommendations to your system.

Note: The Security Wizard can be used more than once on the same system to allow users who may have an older installation to review their current security. The Security Wizard can be used from a V3R7 system (when iSeries Navigator was introduced) upwards.

To use iSeries iSeries Navigator, you must have IBM iSeries Access for Windows installed on your Windows 95/NT PC and have an iSeries server connection from that PC. The user of the Wizard must be connected to an iSeries server. The user must have a user ID that has *ALLOBJ, *SECADM, *AUDIT and *IOSYSCFG special authority. For help in connecting your Windows 95/NT PC to your iSeries system, consult the IBM iSeries Access for Windows topic in the Information Center (see "Prerequisite and related information" on page xii for details).

To access the Security Wizard, do the following:

1. In iSeries Navigator, expand your server.
2. Right-click **Security**, and select **Configure**.
   - When a user starts the **Security** option of the iSeries Navigator a request is sent to the iSeries server to check the user's special authority.
   - Should the user not have all of the required special authority (*ALLOBJ, *AUDIT, *IOSYSCFG, *SECADM) then they will not see the **Configure** option and not be able to access the Security Wizard.
3. Assuming the user has the required authority:
   - Previous wizard responses are retrieved.
   - Current security settings are retrieved.

The Security Wizard will present you with one of three welcome screens. Which screen you see depends on which of the following conditions exists:

- The wizard has never been run for the target iSeries server.
- The wizard has been run before and the security changes were deferred.
- The wizard has been run before and the security changes were put into effect.

If you are not using iSeries Navigator, you can still get help planning for your security needs. The Security Advisor is an on-line version of the Security Wizard, with one difference. The advisor will not automatically configure your system. It will however, generate a report of recommended security options based on your answers. To access the Security Advisor point your Internet browser to the following URL:

`http://www.iseries.ibm.com/tstudio/secure1/advisor/secwiz.htm`

The Security Advisor is part of the iSeries Technical Studio.

Security Advisor

The Security Advisor is an on-line version of the Security Wizard. It asks the same questions as the Security Wizard and, based on your answers, generates the same recommendations. The main differences between the two tools are that:

- The Security Advisor **does not**—
  - Produce reports.
  - Compare current settings with the recommend settings.
  - Set any system value automatically.
- You cannot apply recommendations from the Security Advisor.

The Security Advisor generates a CL program that you can cut-and-paste and edit for your own use to automate security configuration. You can also link directly to the iSeries server documentation from the Security Advisor. This provides information about the system value or report that can help you determine if this setting is appropriate for your environment.

To access the Security Advisor, point your Internet browser to the following URL:

`http://www.iseries.ibm.com/tstudio/secure1/advisor/secwiz.htm`

The Security Advisor is part of the iSeries Technical Studio. This Web site answers many of your security questions and offers timely information on workshops, classes, and other resources. The URL for the Technical Studio address is:

`http://www.iseries.ibm.com/tstudio`

The URL for the iSeries server security section of the Technical Studio is:

`http://www.iseries.ibm.com/tstudio/secure1/secdex.htm`

Chapter 4. Control interactive sign-on

When you think about restricting entry to your system, start with the obvious, the Sign On display. The following are options that you can use to make it difficult for someone to sign on to your system by using the Sign On display.

Set password rules

To secure your system sign-on, do the following:
- Set a policy that states that passwords must not be trivial and must not be shared.
- Set system values to help you with enforcement. Table 1 shows recommended system value settings.

The combination of values in Table 1 is fairly restrictive and is intended to significantly reduce the likelihood of trivial passwords. However, your users may find it difficult and frustrating to select a password that meets these restrictions.

Consider providing users with the following:
1. A list of the criteria for passwords.
2. Examples of passwords that are and are not valid.
3. Suggestions for how to think of a good password.

Run the Configure System Security (CFGSYSSEC) command to set these values. Use the Print System Security Attributes (PRTSYSSECA) command to print your current settings for these system values.

Chapter 3 of the *iSeries Security Reference* book. "Values set by the Configure System Security command" on page 42 provides more information about the CFGSYSSEC command.

Table 1. System Values for Passwords

System value name	Description	Recommended value
QPWDEXPITV	How often the system users must change their passwords. You can specify a different value for individual users in the user profile.	60 (days)
QPWDLMTAJC	Whether the system prevents adjacent characters that are the same.	1 (yes)
QPWDLMTCHR	What characters may not be used in passwords.[2]	AEIOU#$@
QPWDLMTREP	Whether the system prevents the same character from appearing more than once in the password.	2 (not allowed consecutively)
QPWDLVL	Whether user profile passwords are limited to 10 characters or a maximum of 128.	0[3]
QPWDMAXLEN	The maximum number of characters in a password.	8
QPWDMINLEN	The minimum number of characters in a password.	6
QPWDPOSDIF	Whether each character in a password must be different from the character in the same position on the previous password.	1 (yes)
QPWDRQDDGT	Whether the password must have at least one numeric character.	1 (yes)
QPWDRQDDIF	How long a user must wait before using the same password again.[2]	5 or less (expiration intervals)[1]

© Copyright IBM Corp. 1996, 2002

Table 1. System Values for Passwords (continued)

System value name	Description	Recommended value
QPWDVLDPGM	What exit program is called to validate a newly assigned password.	*NONE

Notes:

1. The QPWDEXPITV system value specifies how often you must change your password, such as every 60 days. This is the **expiration interval**. The QPWDRQDDIF system value specifies how many expiration intervals must pass before you can use the same password again. Chapter 3 of the *iSeries Security Reference* book provides more information about how these system values work together.
2. QPWDLMTCHR is not enforced at password levels 2 or 3. See "Password levels" for details.
3. Refer to "Plan password level changes" to determine the password level that is right for your needs.

Password levels

Starting with V5R1 of the operating system, the QPWDLVL system value offers increased password security. In previous releases, users were limited to passwords that were no more than 10 characters long, from a limited range of characters. Now, users can select a password (or passphrase) with as many as 128 characters, depending on the password level at which their system is set. Password levels are:

- **Level 0:** Systems are shipped at this level. At level 0, passwords are no more than 10 characters in length, containing only A-Z, 0–9, #, @, $, and _ characters. Passwords at level 0 are less secure than those at higher password levels.
- **Level 1:** Same rules as password level 0, but passwords for iSeries Support for Windows Network Neighborhood (hereafter referred to as iSeries NetServer) are not saved.
- **Level 2:** Passwords are secure at this level. This level can be used for testing purposes. Passwords are saved for users on level 0 or 1 if they are 10 characters or less, and use the character set for level 0 or 1 passwords. Passwords (or passphrases) at this level have the following characteristics:
  - as many as 128 characters in length.
  - comprised of any available keyboard characters.
  - may not be comprised entirely of blanks; blanks are removed from the end of the password.
  - case sensitive.
- **Level 3:** Passwords at this level are the most secure, and utilize the most advanced encryption algorithms available. Passwords at this level have the same characteristics as at level 2. Passwords for iSeries NetServer are not saved at this level.

You should only use password levels 2 and 3 if every system in your network meets this criteria:

- Operating system is V5R1 or later
- Password level is set to 2 or 3

Similarly, users must all log in using the same password level. Password levels are global; users cannot choose the level at which they want their password secured.

Plan password level changes

Changing password levels should be planned carefully. Operations with other systems may fail or users may not be able to sign on to the system if you haven't

planned for the password level change adequately. Prior to changing the QPWDLVL system value, make sure you have saved your security data using the SAVSECDTA or SAVSYS command. If you have a current backup, you will be able to reset the passwords for all users' profiles if you need to return to a lower password level.

Products that you use on the system, and on clients with which the system interfaces, may have problems when the password level (QPWDLVL) system value is set to 2 or 3. Any product or client that sends passwords to the system in an encrypted form, rather than in the clear text a user enters on a sign-on screen, must be upgraded to work with the new password encryption rules for QPWDLVL 2 or 3. Sending the encrypted password is known as **password substitution**.

Password substitution is used to prevent a password from being captured during transmission over a network. Password substitutes generated by older clients that do not support the new algorithm for QPWDLVL 2 or 3, even if the specific characters are correct, will not be accepted. This also applies to any iSeries to iSeries peer access which utilizes the encrypted values to authenticate from one system to another.

The problem is compounded by the fact that some affected products (such as Java Toolbox) are provided as middleware. A third party product that incorporates a prior version of one of these products will not work correctly until rebuilt using an updated version of the middleware.

Given this and other scenarios, it is easy to see why careful planning is necessary before changing the QPWDLVL system value.

Considerations for changing QPWDLVL from 0 to 1

Password level 1 allows a system, which does not have a need to communicate with the Windows 95/98/ME AS/400® Client Support for Windows Network Neighborhood (iSeries NetServer) product, to have the iSeries NetServer passwords eliminated from the system. Eliminating unnecessary encrypted passwords from the system increases the overall security of the system.

At QPWDLVL 1, all current, pre-V5R1 password substitution and password authentication mechanisms will continue to work. There is very little potential for breakage except for functions and services that require the iSeries NetServer password.

Considerations for changing QPWDLVL from 0 or 1 to 2

Password level 2 introduces the use of case sensitive passwords up to 128 characters in length (also called passphrases) and provides the maximum ability to revert back to QPWDLVL 0 or 1.

Regardless of the password level of the system, password level 2 and 3 passwords are created whenever a password is changed or a user signs on to the system. Having a level 2 and 3 password created while the system is still at password level 0 or 1 helps prepare for the change to password level 2 or 3.

Prior to changing QPWDLVL to 2, you should use the DSPAUTUSR or PRTUSRPRF TYPE(*PWDINFO) commands to locate all user profiles which do not have a password that is usable at password level 2. Depending on which profiles these commands locate, you may want to use one of the following mechanisms to have a password level 2 and 3 password added to the profiles.

- Change the password for the user profile using the CHGUSRPRF or CHGPWD CL command or the QSYCHGPW API. This will cause the system to change the password that is usable at password levels 0 and 1; and the system also creates two equivalent case sensitive passwords that are usable at password levels 2 and 3. An all uppercase and all lowercase version of the password is created for use at password level 2 or 3.

  For example, changing the password to C4D2RB4Y results in the system generating C4D2RB4Y and c4d2rb4y password level 2 passwords.
- Sign on to the system through a mechanism that presents the password in clear text (does not use password substitution). If the password is valid and the user profile does not have a password that is usable at password levels 2 and 3, the system creates two equivalent case sensitive passwords that are usable at password levels 2 and 3. An all uppercase and all lowercase version of the password is created for use at password level 2 or 3.

The absence of a password that is usable at password level 2 or 3 can be a problem whenever the user profile also does not have a password that is usable at password levels 0 and 1 or when the user tries to sign on through a product that uses password substitution. In these cases, the user will not be able to sign on when the password level is changed to 2.

If a user profile does not have a password that is usable at password levels 2 and 3, the user profile does have a password that is usable at password levels 0 and 1, and the user signs on through a product that sends clear text passwords, then the system validates the user against the password level 0 password and creates two password level 2 passwords (as described above) for the user profile. Subsequent sign ons will be validated against the password level 2 passwords.

Any client/service which uses password substitution will not work correctly at QPWDLVL 2 if the client/service hasn't been updated to use the new password (passphrase) substitution scheme. The administrator should check whether a client/service which hasn't been updated to the new password substitution scheme is required.

The clients/services that use password substitution include:

- TELNET
- iSeries Access
- iSeries Host Servers
- QFileSrv.400
- iSeries NetServer print support
- DDM
- DRDA®
- SNA LU6.2

It is highly recommended that the security data be saved prior to changing to QPWDLVL 2. This can help make the transition back to QPWDLVL 0 or 1 easier if that becomes necessary.

It is recommended that the other password system values, such as QPWDMINLEN and QPWDMAXLEN not be changed until after some testing at QPWDLVL 2 has occurred. This will make it easier to transition back to QPWDLVL 1 or 0 if necessary. However, the QPWDVLDPGM system value must specify either *REGFAC or *NONE before the system will allow QPWDLVL to be changed to 2.

Therefore, if you use a password validation program, you may wish to write a new one that can be registered for the QIBM_QSY_VLD_PASSWRD exit point by using the ADDEXITPGM command.

iSeries NetServer passwords are still supported at QPWDLVL 2, so any function/service that requires an iSeries NetServer password should still function correctly.

Once the administrator is comfortable with running the system at QPWDLVL 2, they can begin to change the password system values to exploit longer passwords. However, the administrator needs to be aware that longer passwords will have these effects:

- If passwords greater than 10 characters are specified, the password level 0 and 1 password is cleared. This user profile would not be able to signon if the system is returned to password level 0 or 1.
- If passwords contain special characters or do not follow the composition rules for simple object names (excluding case sensitivity), the password level 0 and 1 password is cleared.
- If passwords greater than 14 characters are specified, the iSeries NetServer password for the user profile is cleared.
- The password system values only apply to the new password level 2 value and do not apply to the system generated password level 0 and 1 password or iSeries NetServer password values (if generated).

Considerations for changing QPWDLVL from 2 to 3

After running the system at QPWDLVL 2 for some period of time, the administrator can consider moving to QPWDLVL 3 to maximize his password security protection.

At QPWDLVL 3, all iSeries NetServer passwords are cleared so a system should not be moved to QPWDLVL 3 until there is no need to use iSeries NetServer passwords.

At QPWDLVL 3, all password level 0 and 1 passwords are cleared. The administrator can use the DSPAUTUSR or PRTUSRPRF commands to locate user profiles which don't have password level 2 or 3 passwords associated with them.

Change to a lower password level

Returning to a lower QPWDLVL value, while possible, is not expected to be a completely painless operation. In general, the mind set should be that this is a one-way trip from lower QPWDLVL values to higher QPWDLVL values. However, there may be cases where a lower QPWDLVL value must be reinstated.

The following sections each discuss the work required to move back to a lower password level.

Considerations for changing from QPWDLVL 3 to 2: This change is relatively easy. Once the QPWDLVL is set to 2, the administrator needs to determine if any user profile is required to contain iSeries NetServer passwords or password level 0 or 1 passwords and, if so, change the password of the user profile to an allowable value.

Additionally, the password system values may have to be changed back to values compatible with iSeries NetServer and password level 0 or 1 passwords, if those passwords are needed.

Considerations for changing from QPWDLVL 3 to 1 or 0: Because of the very high potential for causing problems for the system (like no one can sign on because all of the password level 0 and 1 passwords have been cleared), this change is not supported directly. To change from QPWDLVL 3 to QPWDLVL 1 or 0, the system must first make the intermediary change to QPWDLVL 2.

Considerations for changing from QPWDLVL 2 to 1: Prior to changing QPWDLVL to 1, the administrator should use the DSPAUTUSR or PRTUSRPRF TYPE(*PWDINFO) commands to locate any user profiles that do not have a password level 0 or 1 password. If the user profile will require a password after the QPWDLVL is changed, the administrator should ensure that a password level 0 and 1 password is created for the profile using one of the following mechanisms:

- Change the password for the user profile using the CHGUSRPRF or CHGPWD CL command or the QSYCHGPW API. This will cause the system to change the password that is usable at password levels 2 and 3; and the system also creates an equivalent uppercase password that is usable at password levels 0 and 1. The system is only able to create the password level 0 and 1 password if the following conditions are met:
  - The password is 10 characters or less in length.
  - The password can be converted to uppercase EBCDIC characters A-Z, 0-9, @, #, $, and underscore.
  - The password does not begin with a numeric or underscore character.

  For example, changing the password to a value of RainyDay would result in the system generating a password level 0 and 1 password of RAINYDAY. But changing the the password value to Rainy Days In April would cause the system to clear the password level 0 and 1 password (because the password is too long and it contains blanks).

  No message or indication is produced if the password level 0 or 1 password could not be created.
- Sign on to the system through a mechanism that presents the password in clear text (does not use password substitution). If the password is valid and the user profile does not have a password that is usable at password levels 0 and 1, the system creates an equivalent uppercase password that is usable at password levels 0 and 1. The system is only able to create the password level 0 and 1 password if the conditions listed above are met.

The administrator can then change QPWDLVL to 1. All iSeries NetServer passwords are cleared when the change to QPWDLVL 1 takes effect (next IPL).

Considerations for changing from QPWDLVL 2 to 0: The considerations are the same as for changing from QPWDLVL 2 to 1 except that all iSeries NetServer passwords are retained when the change takes effect.

Considerations for changing from QPWDLVL 1 to 0: After changing QPWDLVL to 0, the administrator should use the DSPAUTUSR or PRTUSRPRF commands to locate any user profiles that do not have an iSeries NetServer password. If the user profile requires an iSeries NetServer password, it can be created by changing the user's password or signing on through a mechanism that presents the password in clear text.

The administrator can then change QPWDLVL to 0.

Change known passwords

Do the following to close some well-known entrances into the iSeries server that may exist on your system.

__ Step 1. Make sure that no user profiles still have default passwords (equal to the user profile name). You can use the Analyze Default Passwords (ANZDFTPWD) command. (See "Avoid default passwords" on page 30.)

__ Step 2. Try to sign on to your system with the combinations of user profiles and passwords that are shown in Table 2. These passwords are published, and they are the first choice of anyone who is trying to break into your system. If you can sign on, use the Change User Profile (CHGUSRPRF) command to change the password to the recommended value.

__ Step 3. Start the Dedicated Service Tools (DST) and try to sign on with the passwords that are shown in Table 2. Refer to the iSeries Information Center—>Security—>Service Tools. See "Prerequisite and related information" on page xii for information on accessing the iSeries Information Center.

__ Step 4. If you can sign on to DST with any of these passwords, you should change the passwords. The iSeries Information Center—>Security—>Service Tools provide detailed instructions on how to change the service tools user IDs and passwords. See "Prerequisite and related information" on page xii for information on accessing the iSeries Information Center.

__ Step 5. Finally, make sure that you cannot sign on just by pressing the Enter key at the Sign On display without entering a user ID and password. Try several different displays. If you can sign on without entering information on the Sign On display, do one of the following:

- Change to security level 40 or 50 (QSECURITY system value).

  Note: Your applications might run differently when you increase your security level to 40 or 50.

- Change all of the workstation entries for interactive subsystems to point to job descriptions that specify USER(*RQD).

Table 2. Passwords for IBM-supplied profiles

User ID	Password	Recommended value
QSECOFR	QSECOFR[1]	A nontrivial value known only to the security administrator. **Write down the password that you have selected and store it in a safe place.**
QSYSOPR	QSYSOPR	*NONE[2]
QPGMR	QPGMR	*NONE[2]
QUSER	QUSER	*NONE[2, 3]
QSRV	QSRV	*NONE[2]
QSRVBAS	QSRVBAS	*NONE[2]

Table 2. Passwords for IBM-supplied profiles (continued)

User ID	Password	Recommended value

Notes:

1. The system arrives with the *Set password to expired* value for the QSECOFR set to *YES. The first time that you sign on to a new system, you must change the QSECOFR password.
2. The system needs these user profiles for system functions, but you should not allow users to sign on with these profiles. For new systems installed with V3R1 or later releases, this password is shipped as *NONE.

   When you run the CFGSYSSEC command, the system sets these passwords to *NONE.
3. To run iSeries Access for Windows using TCP/IP, the QUSER user profile must be enabled.

Table 3. Passwords for dedicated service tools

DST Level	User ID[1]	Password	Recommended value
Basic capability	11111111	11111111	A nontrivial value known only to the security administrator.[2]
Full capability	22222222	22222222[3]	A nontrivial value known only to the security administrator.[2]
Security capability	QSECOFR	QSECOFR[3]	A nontrivial value known only to the security administrator.[2]
Service capability	QSRV	QSRV[3]	A nontrivial value known only to the security administrator.[2]

Notes:

1. A user ID is only required for PowerPC® AS (RISC) releases of the operating system.
2. If your hardware service representative needs to sign on with this user ID and password, change the password to a new value after the hardware service representative leaves.
3. The service tools user profile will expire as soon as it is used for the first time.

Note: DST passwords can only be changed by an authenticated device. This is also true for all passwords and corresponding user IDs that are identical. For more information on authenticated devices, see the Operations Console setup information in the iSeries Information Center.

Set sign-on values

Table 4 shows several values that you can set to make it more difficult for an unauthorized person to sign on to your system. If you run the CFGSYSSEC command, it sets these system values to the recommended settings. You can read more about these system values in Chapter 3 of the *iSeries Security Reference* book.

Table 4. Sign-on system values

System value name	Description	Recommended setting
QAUTOCFG	Whether the system automatically configures new devices.	0 (No)
QAUTOVRT	The number of virtual device descriptions that the system will automatically create if no device is available for use.	0

Table 4. Sign-on system values (continued)

System value name	Description	Recommended setting
QDEVRCYACN	What the system does when a device reconnects after an error.[1]	*DSCMSG
QDSCJOBITV	How long the system waits before ending a disconnected job.	120
QDSPSGNINF	Whether the system displays information about previous sign-on activity when a user signs on.	1 (Yes)
QINACTITV	How long the system waits before taking action when an interactive job is inactive.	60
QINACTMSGQ	What the system does when the QINACTITV time period is reached.	*ENDJOB
QLMTDEVSSN	Whether the system prevents a user from signing on at more than one work station at the same time.	1 (Yes)
QLMTSECOFR	Whether users with *ALLOBJ or *SERVICE special authority can sign on only at specific work stations.	1 (Yes)[2]
QMAXSIGN	Maximum consecutive, incorrect sign-on attempts (user profile or password is incorrect).	3
QMAXSGNACN	What the system does when the QMAXSIGN limit is reached.	3 (Disable both user profile and device)

Notes:

1. The system can disconnect and reconnect TELNET sessions when the device description for the session is explicitly assigned.
2. If you set the system value to 1 (Yes), you will need to explicitly authorize users with *ALLOBJ or *SERVICE special authority to devices. The simplest way to do this is to give the QSECOFR user profile *CHANGE authority to specific devices.

Change sign-on error messages

Hackers like to know when they are making progress toward breaking into a system. When an error message on the Sign On display says `Password not correct`, the hacker can assume that the user ID is correct. You can frustrate the hacker by using the Change Message Description (CHGMSGD) command to change the text for two sign-on error messages. Table 5 shows the recommended text.

Table 5. Sign-on error messages

Message ID	Shipped text	Recommended text
CPF1107	CPF1107 – Password not correct for user profile.	Sign-on information is not correct **Note:** Do not include the message ID in the message text.
CPF1120	CPF1120 – User `XXXXX` does not exist.	Sign-on information is not correct. **Note:** Do not include the message ID in the message text.

Schedule availability of user profiles

You may want some user profiles to be available for sign-on only at certain times of the day or certain days of the week. For example, if you have a profile set up for a security auditor, you may want to enable that user profile only during the hours that the auditor is scheduled to work. You might also want to disable user profiles with *ALLOBJ special authority (including the QSECOFR user profile) during off-hours.

You can use the Change Activation Schedule Entry (CHGACTSCDE) command to set up user profiles to be enabled and disabled automatically. For each user profile that you want to schedule, you create an entry that defines the user profile's schedule.

For example, if you want the QSECOFR profile to be available only between 7 in the morning and 10 in the evening, you would type the following on the CHGACTSCDE display:

```
              Change Activation Scd Entry (CHGACTSCDE)

Type choices, press Enter.

User profile . . . . . > QSECOFR       Name
Enable time  . . . . . > '7:00'        Time, *NONE
Disable time . . . . . > '22:00'       Time, *NONE
Days . . . . . > *MON          *ALL, *MON, *TUE, *WED...
                               > *TUE
                               > *WED
                               > *THU
               + for more values > *FRI
```

Figure 2. Schedule Profile Activation Display–Sample

In fact, you might want to have the QSECOFR profile available only for a very limited number of hours each day. You can use another user profile with the *SECOFR class to perform most system functions. Thus, you avoid exposing a well-known user profile to hacking attempts.

You can use the Display Audit Journal Entries (DSPAUDJRNE) command periodically to print the CP (Change Profile) audit journal entries. Use these entries to verify that the system is enabling and disabling user profiles according to your planned schedule.

Another method for checking to ensure that user profiles are being disabled on your planned schedule is to use the Print User Profile (PRTUSRPRF) command. When you specify *PWDINFO for the report type, the report includes the status of each selected user profile. If, for example, you regularly disable all user profiles with *ALLOBJ special authority, you can schedule the following command to run immediately after the profiles are disabled:

```
PRTUSRPRF TYPE(*PWDINFO) SELECT(*SPCAUT) SPCAUT(*ALLOBJ)
```

Remove inactive user profiles

Your system should contain only user profiles that are necessary. If you no longer need a user profile because the user either has left or has taken a different job within the organization, remove the user profile. If someone is gone from the organization for an extended period, disable (deactivate) that user's profile. An unnecessary user profile may provide unauthorized entry to your system.

Disable user profiles automatically

You can use the Analyze Profile Activity (ANZPRFACT) command to regularly disable user profiles that have been inactive for a specified number of days. When you use the ANZPRFACT command, you specify the number of inactive days that the system looks for. The system looks at the last used date, the restore date, and the creation date for the user profile.

Once you have specified a value for the ANZPRFACT command, the system schedules a job to run weekly at 1 a.m. (starting with the day after you first specified a value). The job examines all profiles and disables inactive profiles. You do not need to use the ANZPRFACT command again unless you want to change the number of inactive days.

You can use the Change Active Profile List (CHGACTPRFL) command to make some profiles exempt from ANZPRFACT processing. The CHGACTPRFL command creates a list of user profiles that the ANZPRFACT command will not disable, no matter how long those profiles have been inactive.

When the system runs the ANZPRFACT command, it writes a CP entry in the audit journal for each user profile that is disabled. You can use the DSPAUDJRNE command to list the user profiles that are newly disabled.

Note: The system writes audit entries only if the QAUDCTL value specifies *AUDLVL and the QAUDLVL system value specifies *SECURITY.

Another method for checking to ensure that user profiles are being disabled on your planned schedule is to use the Print User Profile (PRTUSRPRF) command. When you specify *PWDINFO for the report type, the report includes the status of each selected user profile.

Remove user profiles automatically

You can use the Change Expiration Schedule Entry (CHGEXPSCDE) command to manage the removing or disabling of user profiles. If you know that a user is leaving for an extended period, you can schedule the user profile to be removed or disabled.

The first time that you use the CHGEXPSCDE command, it creates a job schedule entry that runs at 1 minute after midnight every day. The job looks at the QASECEXP file to determine whether any user profiles are scheduled for removal on that day.

With the CHGEXPSCDE command, you either disable or delete a user profile. If you choose to delete a user profile, you must specify what the system will do with the objects that the user owns. Before you schedule a user profile for deletion, you need to research the objects that the user owns. For example, if the user owns programs that adopt authority, do you want those programs to adopt the ownership of the new owner? Or does the new owner have more authority than

necessary (such as special authority)? Perhaps, you need to create a new user profile with specific authorities to own the programs that need to adopt authority.

You also need to research whether any application problems will occur if you delete the user profile. For example, do any job descriptions specify the user profile as the default user?

You can use the Display Expiration Schedule (DSPEXPSCD) command to display the list of profiles that are scheduled to be disabled or removed.

You can use the Display Authorized Users (DSPAUTUSR) command to list all of the user profiles on your system. Use the Delete User Profile (DLTUSRPRF) command to delete outdated profiles.

Security note:: You disable a user profile by setting its status to *DISABLED. When you disable a user profile, you make it unavailable for interactive use. You cannot sign on with or change your job to a disabled user profile. Batch jobs can run under a user profile that is disabled.

Avoid default passwords

When you create a new user profile, the default is to make the password the same as the user profile name. This provides an opportunity for someone to enter your system, if someone knows your policy for assigning profile names and knows that a new person is joining your organization.

When you create new user profiles, consider assigning a unique, non-trivial password instead of using the default password. Tell the new user the password confidentially, such as in a "Welcome to the System" letter that outlines your security policies. Require the user to change the password the first time that the user signs on by setting the user profile to PWDEXP(*YES).

You can use the Analyze Default Passwords (ANZDFTPWD) command to check all the user profiles on your system for default passwords. When you print the report, you have the option of specifying that the system should take action (such as disabling the user profile) if the password is the same as the user profile name. The ANZDFTPWD command prints a list of the profiles that it found and any action that it took.

Note: Passwords are stored on your system in one-way encrypted form. They cannot be decrypted. The system encrypts the specified password and compares it to the stored password just as it would check a password when you sign on the system. If you are auditing authority failures (*AUTFAIL), the system will write a PW audit journal entry for each user profile that does *not* have a default password (for systems running V4R1 or earlier releases). Beginning with V4R2, the system does not write PW audit journal entries when you run the ANZDFTPWD command.

Monitor sign-on and password activity

If you are concerned about unauthorized attempts to enter your system, you can use the PRTUSRPRF command to help you monitor sign-on and password activity.

Following are several suggestions for using this report:

- Determine whether the password expiration interval for some user profiles is longer than the system value and whether the longer expiration interval is justified. For example, in the report, USERY has a password expiration interval of 120 days.
- Run this report regularly to monitor unsuccessful sign-on attempts. Someone who is trying to break into your system may be aware that your system takes action after a certain number of unsuccessful attempts. Each night, the would-be intruder might try fewer times than your QMAXSIGN value to avoid alerting you to the attempts. However, if you run this report early each morning and notice that certain profiles often have unsuccessful sign-on attempts, you might suspect that you have a problem.
- Identify user profiles that have not been used for a long time or whose passwords have not been changed for a long time.

Store password information

To support some network functions and communications requirements, iSeries servers provide a secure method for storing passwords that can be decrypted. Your system uses these passwords, for example, to establish a SLIP connection with another system. (“Security and dial-out sessions” on page 128 describes this use of stored passwords.)

iSeries servers stores these special passwords in a secure area that is not accessible to any user programs or interfaces. Only explicitly authorized system functions can set these passwords and retrieve them.

For example, when you use a stored password for dial-out SLIP connections, you set the password with the system command that creates the configuration profile (WRKTCPPTP). You must have *IOSYSCFG to use the command. A specially coded connection script retrieves the password and decrypts it during the dial-out procedure. The decrypted password is not visible to the user or in any job log.

As a security administrator, you need to decide whether you will allow passwords that can be decrypted to be stored on your system. You use the Retain Server Security Data (QRETSVRSEC) system value to specify this. The default is 0 (No). Therefore, your system will not store passwords that can be decrypted unless you explicitly set this system value.

If you have network or communications requirements for stored passwords, you should set appropriate policies and understand the policies and practices of your communications partners. For example, when you use SLIP to communicate with another iSeries server, both systems should consider setting up special user profiles for establishing the sessions. The special profiles should have limited authority on the system. This limits the impact to your system if a stored password is compromised on a partner system.

Chapter 5. Configure the iSeries to use Security Tools

This information describes how to set up your system to use the security tools that are part of OS/400. When you install OS/400, the security tools tools are ready to use. The topics that follow provide suggestions for operating procedures with the security tools.

Operate Security Tools securely

When you install OS/400, the objects that are associated with the security tools are secure. To operate the security tools securely, avoid making authority changes to any security tool objects.

Following are the security settings and requirements for security tool objects:

- The security tool programs and commands are in the QSYS product library. The commands and the programs ship with the public authority of *EXCLUDE. Many of the security tool commands create files in the QUSRSYS library. When the system creates these files, the public authority for the files is *EXCLUDE.

  Files that contain information for producing changed reports have names that begin with QSEC. Files that contain information for managing user profiles have names that begin with QASEC. These files contain confidential information about your system. Therefore, you should not change the public authority to the files.
- The security tools use your normal system setup for directing printed output. These reports contain confidential information about your system. To direct the output to a protect output queue, make appropriate changes to the user profile or job description for users who will be running the security tools.
- Because of their security functions and because they access many objects on the system, the security tool commands require *ALLOBJ special authority. Some of the commands also require *SECADM, *AUDIT, or *IOSYSCFG special authority. To ensure that the commands run successfully, you should sign on as a security officer when you use the security tools. Therefore, you should not need to grant private authority to any security tool commands.

Avoid file conflicts

Many of the security tool report commands create a database file that you can use to print a changed version of the report. Commands and menus for security commands tells the file name for each command. You can only run a command from one job at a time. Most of the commands now have checks that enforce this. If you run a command when another job has not yet finished running it, you will receive an error message.

Many print jobs are long-running jobs. You need to be careful to avoid file conflicts when you submit reports to batch or add them to the job scheduler. For example, you might want to print two versions of the PRTUSRPRF report with different selection criteria. If you are submitting reports to batch, you should use a job queue that runs only one job at a time to ensure that the report jobs run sequentially.

If you are using the job scheduler, you need to schedule the two jobs far enough apart that the first version completes before the second job starts.

© Copyright IBM Corp. 1996, 2002

Save Security Tools

You save the security tool programs whenever you run either the Save System (SAVSYS) command or an option from the Save menu that runs the SAVSYS command.

The security tool files are in the QUSRSYS library. You should already be saving this library as part of your normal operating procedures. The QUSRSYS library contains data for many licensed programs on your system. See the Information Center for more information about what commands and options save the QUSRSYS library.

Commands and menus for security commands

This section describes the commands and menus for security tools. Examples of how to use the commands are included throughout this information.

Two menus are available for security tools:

- The SECTOOLS (Security Tools) menu to run commands interactively.
- The SECBATCH (Submit or Schedule Security Reports to Batch) menu to run the report commands in batch. The SECBATCH menu has two parts. The first part of the menu uses the Submit Job (SBMJOB) command to submit reports for immediate processing in batch.

  The second part of the menu uses the Add Job Schedule Entry (ADDJOBSCDE) command. You use it to schedule security reports to be run regularly at a specified day and time.

Security Tools menu options

Table 6 describes these menu options and the associated commands:

Table 6. Tool commands for user profiles

Menu[1] option	Command name	Description	Database file used
1	ANZDFTPWD	Use the Analyze Default Passwords command to report on and take action on user profiles that have a password equal to the user profile name.	QASECPWD[2]
2	DSPACTPRFL	Use the Display Active Profile List command to display or print the list of user profiles that are exempt from ANZPRFACT processing.	QASECIDL[2]
3	CHGACTPRFL	Use the Change Active Profile List command to add and remove user profiles from the exemption list for the ANZPRFACT command. A user profile that is on the active profile list is permanently active (until you remove the profile from the list). The ANZPRFACT command does not disable a profile that is on the active profile list, no matter how long the profile has been inactive.	QASECIDL[2]

Table 6. Tool commands for user profiles (continued)

Menu[1] option	Command name	Description	Database file used
4	ANZPRFACT	Use the Analyze Profile Activity command to disable user profiles that have not been used for a specified number of days. After you use the ANZPRFACT command to specify the number of days, the system runs the ANZPRFACT job nightly. You can use the CHGACTPRFL command to exempt user profiles from being disabled.	QASECIDL[2]
5	DSPACTSCD	Use the Display Profile Activation Schedule command to display or print information about the schedule for enabling and disabling specific user profiles. You create the schedule with the CHGACTSCDE command.	QASECACT[2]
6	CHGACTSCDE	Use the Change Activation Schedule Entry command to make a user profile available for sign on only at certain times of the day or week. For each user profile that you schedule, the system creates job schedule entries for the enable and disable times.	QASECACT[2]
7	DSPEXPSCD	Use the Display Expiration Schedule command to display or print the list of user profiles that are scheduled to be disabled or removed from the system in the future. You use the CHGEXPSCDE command to set up user profiles to expire.	QASECEXP[2]
8	CHGEXPSCDE	Use the Change Expiration Schedule Entry command to schedule a user profile for removal. You can remove it temporarily (by disabling it) or you can delete it from the system. This command uses a job schedule entry that runs every day at 00:01 (1 minute after midnight). The job looks at the QASECEXP file to determine whether any user profiles are set up to expire on that day. Use the DSPEXPSCD command to display the user profiles that are scheduled to expire.	QASECEXP[2]
9	PRTPRFINT	Use the Print Profile Internals command to print a report containing information on the number of entries contained in a user profile. The number of entries determines the size of the user profile.	

Notes:

1. Options are from the SECTOOLS menu.
2. This file is in the QUSRSYS library.

You can page down on the menu to see additional options. Table 7 on page 36 describes the menu options and associated commands for security auditing:

Table 7. Tool commands for security auditing

Menu[1] option	Command name	Description	Database file used
10	CHGSECAUD	Use the Change Security Auditing command to set up security auditing and to change the system values that control security auditing. When you run the CHGSECAUD command, the system creates the security audit (QAUDJRN) journal if it does not exist. The CHGSECAUD command provides options that make it simpler to set the QAUDLVL (audit level) system value. You can specify *ALL to activate all of the possible audit level settings. Or, you can specify *DFTSET to activate the most commonly used settings (*AUTFAIL, *CREATE, *DELETE, *SECURITY, and *SAVRST). **Note:** If you use the security tools to set up auditing, be sure to plan for management of your audit journal receivers. Otherwise, you might quickly encounter problems with disk utilization.	
11	DSPSECAUD	Use the Display Security Auditing command to display information about the security audit journal and the system values that control security auditing.	

Notes:

1. Options are from the SECTOOLS menu.

Use the Security Batch menu

Following is the first part of the SECBATCH menu:

```
SECBATCH         Submit or Schedule Security Reports To Batch
                                                                    System:
Select one of the following:

  Submit Reports to Batch
     1. Adopting objects
     2. Audit journal entries
     3. Authorization list authorities
     4. Command authority
     5. Command private authorities
     6. Communications security
     7. Directory authority
     8. Directory private authority
     9. Document authority
    10. Document private authority
    11. File authority
    12. File private authority
    13. Folder authority
```

When you select an option from this menu, you see the Submit Job (SBMJOB) display. If you want to change the default options for the command, you can press F4 (Prompt) on the *Command to run* line.

To see the Schedule Batch Reports, page down on the SECBATCH menu. By using the options on this part of the menu, you can, for example, set up your system to run changed versions of reports regularly. You can page down for additional menu

options. When you select an option from this part of the menu, you see the Add Job Schedule Entry (ADDJOBSCDE) display.

You can position your cursor on the *Command to run* line and press F4 (Prompt) to choose different settings for the report. You should assign a meaningful job name so that you can recognize the entry when you display the job schedule entries.

Security Batch menu options

Table 8 describes the menu options and associated commands for security reports.

When you run security reports, the system prints only information that meets both the selection criteria that you specify and the selection criteria for the tool. For example, job descriptions that specify a user profile name are security-relevant. Therefore, the job description (PRTJOBDAUT) report prints job descriptions in the specified library only if the public authority for the job description is not *EXCLUDE *and* if the job description specifies a user profile name in the USER parameter.

Similarly, when you print subsystem information (PRTSBSDAUT command), the system prints information about a subsystem only when the subsystem description has a communications entry that specifies a user profile.

If a particular report prints less information than you expect, consult the online help information to find out the selection criteria for the report.

Table 8. Commands for security reports

Menu[1] option	**Command name**	**Description**	**Database file used**
1, 40	PRTADPOBJ	Use the Print Adopting Objects command to print a list of objects that adopt the authority of the specified user profile. You can specify a single profile, a generic profile name (such as all profiles that begin with Q), or all user profiles on the system. This report has two versions. The full report lists all adopted objects that meet the selection criteria. The changed report lists differences between adopted objects that are currently on the system and adopted objects that were on the system the last time that you ran the report.	QSECADPOLD[2]
2, 41	DSPAUDJRNE	Use the Display Audit Journal Entries command to display or print information about entries in the security audit journal. You can select specific entry types, specific users, and a time period.	QASYxxJ4[3]

Table 8. Commands for security reports (continued)

Menu[1] option	Command name	Description	Database file used
3, 42	PRTPVTAUT *AUTL	When you use the Print Private Authorities command for *AUTL objects, you receive a list of all the authorization lists on the system. The report includes the users who are authorized to each list and what authority the users have to the list. Use this information to help you analyze sources of object authority on your system. This report has three versions. The full report lists all authorization lists on the system. The changed report lists additions and changes to authorization since you last ran the report. The deleted report lists users whose authority to the authorization list has been deleted since you last ran the report. When you print the full report, you have the option to print a list of objects that each authorization list secures. The system will create a separate report for each authorization list.	QSECATLOLD[2]
6, 45	PRTCMNSEC	Use the Print Communications Security command to print the security-relevant settings for objects that affect communications on your system. These settings affect how users and jobs can enter your system. This command produces two reports: a report that displays the settings for configuration lists on the system and a report that lists security-relevant parameters for line descriptions, controllers, and device descriptions. Each of these reports has a full version and a changed version.	QSECCMNOLD[2]
15, 54	PRTJOBDAUT	Use the Print Job Description Authority command to print a list of job descriptions that specify a user profile and have public authority that is not *EXCLUDE. The report shows the special authorities for the user profile that is specified in the job description. This report has two versions. The full report lists all job description objects that meet the selection criteria. The changed report lists differences between job description objects that are currently on the system and job description objects that were on the system the last time that you ran the report.	QSECJBDOLD[2]

Table 8. Commands for security reports (continued)

Menu[1] option	Command name	Description	Database file used
See note 4	PRTPUBAUT	Use the Print Publicly Authorized Objects command to print a list of objects whose public authority is not *EXCLUDE. When you run the command, you specify the type of object and the library or libraries for the report. Use the PRTPUBAUT command to print information about objects that every user on the system can access. This report has two versions. The full report lists all objects that meet the selection criteria. The changed report lists differences between the specified objects that are currently on the system and objects (of the same type in the same library) that were on the system the last time that you ran the report.	QPBxxxxxx[5]
See note 5.	PRTPVTAUT	Use the Print Private Authorities command to print a list of the private authorities to objects of the specified type in the specified library. Use this report to help you determine the sources of authority to objects. This report has three versions. The full report lists all objects that meet the selection criteria. The changed report lists differences between the specified objects that are currently on the system and objects (of the same type in the same library) that were on the system the last time that you ran the report. The deleted report lists users whose authority to an object has been deleted since you last printed the report.	QPVxxxxxx[5]
24, 63	PRTQAUT	Use the Print Queue Report to print the security settings for output queues and job queues on your system. These settings control who can view and change entries in the output queue or job queue. This report has two versions. The full report lists all output queue and job queue objects that meet the selection criteria. The changed report lists differences between output queue and job queue objects that are currently on the system and output queue and job queue objects that were on the system the last time that you ran the report.	QSECQOLD[2]

Table 8. Commands for security reports (continued)

Menu[1] option	Command name	Description	Database file used
25, 64	PRTSBSDAUT	Use the Print Subsystem Description command to print the security-relevant communications entries for subsystem descriptions on your system. These settings control how work can enter your system and how jobs run. The report prints a subsystem description only if it has communications entries that specify a user profile name. This report has two versions. The full report lists all subsystem description objects that meet the selection criteria. The changed report lists differences between subsystem description objects that are currently on the system and subsystem description objects that were on the system the last time that you ran the report.	QSECSBDOLD[2]
26, 65	PRTSYSSECA	Use the Print System Security Attributes command to print a list of security-relevant system values and network attributes. The report shows the current value and the recommended value.	
27, 66	PRTTRGPGM	Use the Print Trigger Programs command to print a list of trigger programs that are associated with database files on your system. This report has two versions. The full report lists every trigger program that is assigned and meets your selection criteria. The changed report lists trigger programs that have been assigned since the last time that you ran the report.	QSECTRGOLD[2]
28, 67	PRTUSROBJ	Use the Print User Objects command to print a list of the user objects (objects not supplied by IBM) that are in a library. You might use this report to print a list of user objects that are in a library (such as QSYS) that is in the system portion of the library list. This report has two versions. The full report lists all user objects that meet the selection criteria. The changed report lists differences between user objects that are currently on the system and user objects that were on the system the last time that you ran the report.	QSECPUOLD[2]
29, 68	PRTUSRPRF	Use the Print User Profile command to analyze user profiles that meet specified criteria. You can select user profiles based on special authorities, user class, or a mismatch between special authorities and user class. You can print authority information, environment information, password information, or password level information.	
30, 69	PRTPRFINT	Use the Print Profile Internals command to print a report of internal information on the number of entries.	

Table 8. Commands for security reports (continued)

Menu[1] option	Command name	Description	Database file used
31, 70	CHKOBJITG	Use the Check Object Integrity command to determine whether operable objects (such as programs) have been changed without using a compiler. This command can help you to detect attempts to introduce a virus program on your system or to change a program to perform unauthorized instructions. The *iSeries Security Reference* book provides more information about the CHKOBJITG command.	

Notes:

1. Options are from the SECBATCH menu.
2. This file is in the QUSRSYS library.
3. xx is the two-character journal entry type. For example, the model output file for AE journal entries is QSYS/QASYAEJ4. The model output files are described in Appendix F of the *iSeries Security Reference* book.
4. The SECBATCH menu contains options for the object types that are typically of concern to security administrators. For example, use options 11 or 50 to run the PRTPUBAUT command against *FILE objects. Use the general options (18 and 57) to specify the object type.
5. The SECBATCH menu contains options for the object types that are typically of concern to security administrators. For example, options 12 or 51 run the PRTPVTAUT command against *FILE objects. Use the general options (19 and 58) to specify the object type.
6. The xxxxxx in the name of the file is the object type. For example, the file for program objects is called QPBPGM for public authorities and QPVPGM for private authorities. The files are in the QUSRSYS library.

   The file contains a member for each library for which you have printed the report. The member name is the same as the library name.

Commands for customizing security

Table 9 describes the commands that you can use to customize the security on your system. These commands are on the SECTOOLS menu.

Table 9. Commands for customizing your system

Menu[1] option	Command name	Description	Database file used
60	CFGSYSSEC	Use the Configure System Security command to set security-relevant system values to their recommended settings. The command also sets up security auditing on your system. "Values set by the Configure System Security command" on page 42 describes what the command does. **Note:** To obtain security recommendations customized for your situation, run the iSeries Security Wizard or the iSeries Security Advisor instead of running this command. See Chapter 3, "iSeries Security Wizard and Security Advisor" on page 15 for information on these tools.	
61	RVKPUBAUT	Use the Revoke Public Authority command to set the public authority to *EXCLUDE for a set of security-sensitive commands on your system. "Functions of the Revoke Public Authority command" on page 44 lists the actions that the RVKPUBAUT command performs.	

Table 9. Commands for customizing your system (continued)

Menu[1] option	Command name	Description	Database file used
Notes: 1. Options are from the SECTOOLS menu.			

Values set by the Configure System Security command

Table 10 lists the system values that are set when you run the CFGSYSSEC command. The CFGSYSSEC command runs a program that is called QSYS/QSECCFGS.

Table 10. Values set by the CFGSYSSEC command

System value name	Setting	System value description
QALWOBJRST	*NONE	Whether system state programs and programs that adopt authority can be restored
QAUTOCFG	0 (No)	Automatic configuration of new devices
QAUTOVRT	0	The number of virtual device descriptions that the system will automatically create if no device is available for use.
QDEVRCYACN	*DSCMSG (Disconnect with message)	System action when communications is re-established
QDSCJOBITV	120	Time period before the system takes action on a disconnected job
QDSPSGNINF	1 (Yes)	Whether users see the sign-on information display
QINACTITV	60	Time period before the system takes action on an inactive interactive job
QINACTMSGQ	*ENDJOB	Action that the system takes for an inactive job
QLMTDEVSSN	1 (Yes)	Whether users are limited to signing on at one device at a time
QLMTSECOFR	1 (Yes)	Whether *ALLOBJ and *SERVICE users are limited to specific devices
QMAXSIGN	3	How many consecutive, unsuccessful sign-on attempts are allowed
QMAXSGNACN	3 (Both)	Whether the system disables the workstation or the user profile when the QMAXSIGN limit is reached.
QRMTSIGN	*FRCSIGNON	How the system handles a remote (pass-through or TELNET) sign-on attempt.
QRMTSVRATR	0 (Off)	Allows the system to be analyzed remotely.
QSECURITY [1 on page 43]	50	The level of security that is enforced
QVFYOBJRST	3 (Verify signatures on restore)	Verify object on restore
QPWDEXPITV	60	How often users must change their passwords
QPWDMINLEN	6	Minimum length for passwords
QPWDMAXLEN	8	Maximum length for passwords
QPWDPOSDIF	1 (Yes)	Whether every position in a new password must differ from the same position in the last password
QPWDLMTCHR	See note 2 on page 43	Characters that are not allowed in passwords
QPWDLMTAJC	1 (Yes)	Whether adjacent numbers are prohibited in passwords
QPWDLMTREP	2 (Cannot be repeated consecutively)	Whether repeating characters in are prohibited in passwords

Table 10. Values set by the CFGSYSSEC command (continued)

System value name	Setting	System value description
QPWDRQDDGT	1 (Yes)	Whether passwords must have at least one number
QPWDRQDDIF	1 (32 unique passwords)	How many unique passwords are required before a password can be repeated
QPWDVLDPGM	*NONE	The user exit program that the system calls to validate passwords

Notes:

1. If you are currently running with a QSECURITY value of 40 or lower, be sure to review the information in Chapter 2 of the *iSeries Security Reference* book before you change to a higher security level.
2. The restricted characters are stored in message ID CPXB302 in the message file QSYS/QCPFMSG. They are shipped as AEIOU@$#. You can use the Change Message Description (CHGMSGD) command to change the restricted characters. The QPWDLMTCHR system value is not enforced at password levels 2 or 3.

The CFGSYSSEC command also sets the password to *NONE for the following IBM-supplied user profiles:

QSYSOPR

QPGMR

QUSER

QSRV

QSRVBAS

Finally, the CFGSYSSEC command sets up security auditing using the Change Security Auditing (CHGSECAUD) command. The CFGSYSSEC command turns on action and object auditing and also, specifies the default set of actions to audit on the CHGSECAUD command.

Customize the program

If some of these settings are not appropriate for your installation, you can create your own version of the program that processes the command. Do the following:

__ Step 1. Use the Retrieve CL Source (RTVCLSRC) command to copy the source for the program that runs when you use the CFGSYSSEC command. The program to retrieve is QSYS/QSECCFGS. When you retrieve it, give it a *different name*.

__ Step 2. Edit the program to make your changes. Then compile it. When you compile it, make sure that you *do not* replace the IBM-supplied QSYS/QSECCFGS program. Your program should have a different name.

__ Step 3. Use the Change Command (CHGCMD) command to change the program to process command (PGM) parameter for the CFGSYSSEC command. Set the PGM value to the name of your program. For example, if you create a program in the QGPL library that is called MYSECCFG, you would type the following:

```
CHGCMD CMD(QSYS/CFGSYSSEC) PGM(QGPL/MYSECCFG)
```

Note: If you change the QSYS/QSECCFGS program, IBM cannot guarantee or imply reliability, serviceability, performance or function of the program. The implied warranties of merchantability and fitness for a particular purpose are expressly disclaimed.

Functions of the Revoke Public Authority command

You can use the Revoke Public Authority (RVKPUBAUT) command to set the public authority to *EXCLUDE for a set of commands and programs. The RVKPUBAUT command runs a program that is called QSYS/QSECRVKP. As it is shipped, the QSECRVKP revokes public authority (by setting public authority to *EXCLUDE) for the commands that are listed in Table 11 and the application programming interfaces (APIs) that are listed in Table 12. When your system arrives, these commands and APIs have their public authority set to *USE.

The commands that are listed in Table 11 and the APIs that are listed in Table 12 all perform functions on your system that may provide an opportunity for mischief. As security administrator, you should explicitly authorize users to run these commands and programs rather than make them available to all system users.

When you run the RVKPUBAUT command, you specify the library that contains the commands. The default is the QSYS library. If you have more than one national language on your system, you need to run the command for each QSYSxxx library.

Table 11. Commands whose public authority is set by the RVKPUBAUT command

ADDAJE	CHGJOBQE	RMVCMNE
ADDCFGLE	CHGPJE	RMVJOBQE
ADDCMNE	CHGRTGE	RMVPJE
ADDJOBQE	CHGSBSD	RMVRTGE
ADDPJE	CHGWSE	RMVWSE
ADDRTGE	CPYCFGL	RSTLIB
ADDWSE	CRTCFGL	RSTOBJ
CHGAJE	CRTCTLAPPC	RSTS36F
CHGCFGL	CRTDEVAPPC	RSTS36FLR
CHGCFGLE	CRTSBSD	RSTS36LIBM
CHGCMNE	ENDRMTSPT	STRRMTSPT
CHGCTLAPPC	RMVAJE	STRSBS
CHGDEVAPPC	RMVCFGLE	WRKCFGL

The APIs in Table 12 are all in the QSYS library:

Table 12. Programs whose public authority is set by the RVKPUBAUT command

QTIENDSUP
QTISTRSUP
QWTCTLTR
QWTSETTR
QY2FTML

When you run the RVKPUBAUT command, the system sets the public authority for the root directory to *USE (unless it is already *USE or less).

Customize the program

If some of these settings are not appropriate for your installation, you can create your own version of the program that processes the command. Do the following:

__ Step 1. Use the Retrieve CL Source (RTVCLSRC) command to copy the source for the program that runs when you use the RVKPUBAUT command. The program to retrieve is QSYS/QSECRVKP. When you retrieve it, give it a *different name*.

__ Step 2. Edit the program to make your changes. Then compile it. When you compile it, make sure that you *do not* replace the IBM-supplied QSYS/QSECRVKP program. Your program should have a different name.

__ Step 3. Use the Change Command (CHGCMD) command to change the program to process command (PGM) parameter for the RVKPUBAUT command. Set the PGM value to the name of your program. For example, if you create a program in the QGPL library that is called MYRVKPGM, you would type the following:

```
CHGCMD CMD(QSYS/RVKPUBAUT) PGM(QGPL/MYRVKPGM)
```

Note: If you change the QSYS/QSECRVKP program, IBM cannot guarantee or imply reliability, serviceability, performance or function of the program. The implied warranties of merchantability and fitness for a particular purpose are expressly disclaimed.

Part 3. Advanced iSeries security

© Copyright IBM Corp. 1996, 2002

Chapter 6. Protect information assets with object authority

Your challenge as security administrator is to protect your organization's information assets without frustrating the users on your system. You need to make sure that users have enough authority to do their jobs without giving them the authority to browse throughout the system and to make unauthorized changes.

Security tip
Authority that is too tight can backfire. Users sometimes react to authority restrictions that are too tight by sharing passwords with each other.

The OS/400 operating system provides integrated object security. Users must use the interfaces that the system provides to access objects. For example, if you want to access a database file, you must use commands or programs that are intended for accessing database files. You cannot use a command that is intended for accessing a message queue or a job log.

Whenever you use a system interface to access an object, the system verifies that you have the authority to the object that is required by that interface. Object authority is a powerful and flexible tool for protecting the assets on your system. Your challenge as a security administrator is to set up an effective object security scheme that you can manage and maintain.

Object authority enforcement

Whenever you try to access an object, the operating system checks your authority to that object. However, if the security level on your system (QSECURITY system value) is set to 10 or 20, every user automatically has authority to access every object because every user profile has *ALLOBJ special authority.

Object authority tip: If you are not sure whether you are using object security, check the QSECURITY (security level) system value. If QSECURITY is 10 or 20, you are not using object security.

You must plan and prepare before you change to security level 30 or higher. Otherwise, your users may not be able to access the information that they need.

The **Basic system security and planning** topic in the Information Center provides a method for analyzing your applications and deciding how you should set up object security. If you are not yet using object security or if your object security scheme is outdated and convoluted, read this topic to help you get started.

Menu security

The iSeries server was originally designed as a follow-on product for S/36 and S/38. Many iSeries server installations were, at one time, S/36 installations or S/38 installations. To control what users could do, security administrators on those earlier systems often used a technique that is referred to as **menu security** or **menu access control**.

© Copyright IBM Corp. 1996, 2002

Menu access control means that when a user signs on, the user sees a menu. The user can perform only the functions that are on the menu. The user cannot get to a command line on the system to perform any functions that are not on the menu. In theory, the security administrator does not have to worry about authority to objects because menus and programs control what users can do.

The iSeries server provides several user profile options to assist with menu access control, you can use the:

- **Initial menu** (INLMNU) parameter to control what menu the user first sees after the user signs on.
- **Initial program** (INLPGM) parameter to run a setup program before the user sees a menu. Or, you can use the INLPGM parameter to restrict a user to running a single program.
- **Limit capabilities** (LMTCPB) parameter to restrict a user to a limited set of commands. It also prevents the user from specifying a different initial program or menu on the Sign On display. (The LMTCPB parameter only limits commands that are entered from the command line.)

Limitations of menu access control

Computers and computer users have changed a great deal in the past few years. Many tools, such as query programs and spreadsheets, are available so that users can do some of their own programming to off-load IS departments. Some tools, such as SQL or ODBC, provide the capability to view information and to change information. To enable these tools within a menu structure is very difficult.

Fixed-function ("green-screen") workstations are rapidly being replaced by personal computers and computer-to-computer networks. If your system participates in a network, users may enter your system without ever seeing a sign-on display or a menu.

As a security administrator who is trying to enforce menu access control, you have two basic problems:

- If you are successful in limiting users to menus, your users will probably be unhappy because their ability to use modern tools is limited.
- If you are not successful, you could jeopardize critical, confidential information that menu access control is supposed to protect. When your system participates in a network, your ability to enforce menu access control decreases. For example, the LMTCPB parameter applies only to commands that are entered from a command line in an interactive session. The LMTCPB parameter has no affect on requests from communications sessions, such as PC file transfer, FTP, or remote commands.

Enhance menu access control with object security

With the many new options that are available to connect to systems, a viable iSeries server security scheme for the future cannot rely solely on menu access control. This topic provides suggestions for moving toward an object security environment to complement your menu access control.

The *Basic system security and planning* topic in the Information Center describes a technique for analyzing the authority that users must have to objects to run your current applications. You then assign users to groups and give the groups appropriate authority. This approach is reasonable and logical. However, if your

system has been operational for many years and has many applications, the task of analyzing applications and setting up object authority probably seems overwhelming.

Object authority tip: Your current menus combined with programs that adopt the authority of the program owners may provide a transition beyond menu access control. Be sure to protect both the programs that adopt authority and the user profiles that own them.

You may be able to use your current menus to help you set up a transition environment while you gradually analyze your applications and objects. Following is an example that uses the Order Entry (OEMENU) menu and the associated files and programs.

Example: Set up a transition environment

This example starts with the following assumptions and requirements:

- All of the files are in the library ORDERLIB.
- You do not know the names of all the files. You also do not know what authority the menu options require to different files.
- The menu and all the programs that it calls are in a library called ORDERPGM.
- You want everyone who can sign on to your system to be able to view information in all the order files, customer files, and item files (with queries or spreadsheets, for example).
- Only users whose current sign-on menu is the OEMENU should be able to change the files. And, they must use the programs on the menu to do this.
- System users other than the security administrators do not have *ALLOBJ or *SECADM special authority.

Perform the following steps to change this menu-access-control environment to accommodate the need for queries:

__ Step 1. Make a list of the users whose initial menu is the OEMENU.

You can use the Print User Profile (PRTUSRPRF *ENVINFO) command to list the environment for every user profile on your system. The report includes the initial menu, initial program, and current library. Figure 7 on page 66 shows an example of the report.

__ Step 2. Make sure that the OEMENU object (it may be a *PGM object or a *MENU object) is owned by a user profile that is not used for sign on. The user profile should be disabled or have a password of *NONE. For this example, assume that OEOWNER owns the OEMENU program object.

__ Step 3. Make sure that the user profile that owns the OEMENU program object is not a group profile. You can use the following command:

```
DSPUSRPRF USRPRF(OEOWNER) TYPE(*GRPMBR)
```

__ Step 4. Change the OEMENU program to adopt the authority of the OEOWNER user profile. (Use the CHGPGM command to change the USRPRF parameter to *OWNER.)

Note: *MENU objects cannot adopt authority. IF OEMENU is a *MENU object, you can adapt this example by doing one of the following:

- Create a program to display the menu.

- Use adopted authority for the programs that run when the user selects options from the OEMENU menu.

__ Step 5. Set the public authority to all of the files in ORDERLIB to *USE by typing the following two commands:

```
RVKOBJAUT OBJ(ORDERLIB/*ALL) OBJTYPE(*FILE) USER(*PUBLIC)
     AUT(*ALL)
GRTOBJAUT OBJ(ORDERLIB/*ALL) OBJTYPE(*FILE) USER(*PUBLIC)
     AUT(*USE)
```

Remember that if you select *USE authority, users can copy the file by using PC file transfer or FTP.

__ Step 6. Give the profile that owns the menu program *ALL authority to the files by typing the following:

```
GRTOBJAUT OBJ(ORDERLIB/*ALL) OBJTYPE(*FILE) USER(OEOWNER)
     AUT(*ALL)
```

For most applications, *CHANGE authority to files is sufficient. However, your applications may perform functions, such as clearing physical file members, that require more authority than *CHANGE. Eventually, you should analyze your applications and provide only the minimum authority that is necessary for the application. However, during the transition period, by adopting *ALL authority, you avoid applications failures that may be caused by insufficient authority.

__ Step 7. Restrict authority to the programs in the order library by typing the following:

```
GRTOBJAUT OBJ(ORDERPGM/*ALL) OBJTYPE(*PGM) USER(*PUBLIC)
     AUT(*EXCLUDE)
```

__ Step 8. Give the OEOWNER profile authority to the programs in the library by typing the following:

```
GRTOBJAUT OBJ(ORDERPGM/*ALL) OBJTYPE(*PGM) USER(OEOWNER)
     AUT(*USE)
```

__ Step 9. Give the users that you identified in step 1 authority to the menu program by typing the following for each user:

```
GRTOBJAUT OBJ(ORDERPGM/OEMENU) OBJTYPE(*PGM)
     USER(user-profile-name) AUT(*USE)
```

When you have completed these steps, all system users who are not explicitly excluded will be able to access (but not change) the files in the ORDERLIB library. Users who have authority to the OEMENU program will be able to use the programs that are on the menu to update files in the ORDERLIB library. Only users who have authority to the OEMENU program will now be able to change the files in this library. A combination of object security and menu access control protects the files.

When you complete similar steps for all the libraries that contain user data, you have created a simple scheme for controlling database updates. This method prevents system users from updating database files except when they use the approved menus and programs. At the same time, you have made database files available for viewing, analyzing, and copying by users with decision-support tools or with links from another system or from a PC.

Object authority tip: When your system participates in a network, *USE authority may provide more authority than you expect. For example, with FTP, you can make a copy of a file to another system (including a PC) if you have *USE authority to the file.

Use library security to complement menu security

To access an object in a library, you must have authority both to the object and to the library. Most operations require either *EXECUTE authority or *USE authority to the library.

Depending on your situation, you may be able to use library authority as a simple means for securing objects. For example, assume that for the Order-Entry menu example, everyone who has authority to the Order Entry menu can use all of the programs in the ORDERPGM library. Rather than securing individual programs, you can set the public authority to the ORDERPGM library to *EXCLUDE. You can then grant *USE authority to the library to specific user profiles, which will allow them to use the programs in the library. (This assumes that public authority to the programs is *USE or greater.)

Library authority can be a simple, efficient method for administering object authority. However, you must ensure that you are familiar with the contents of the libraries that you are securing so that you do not provide unintended access to objects.

Configure object ownership

The ownership of objects on your system is an important part of your object authority scheme. By default, the owner of an object has *ALL authority to the object. Chapter 5 in the *iSeries Security Reference* book provides recommendations and examples for planning object ownership. Following are a few tips:

- In general, group profiles should not own objects. If a group profile owns an object, all group members have *ALL authority to the object unless the group member is explicitly excluded.
- If you use adopted authority, consider whether the user profiles that own programs should also own application objects, such as files. You may not want the users who run the programs that adopt authority to have *ALL authority to files.

If you are using iSeries Navigator, this can be accomplished by completing the changes using the security **policies** function. For more information, refer to the iSeries Information Center (see “Prerequisite and related information” on page xii for details).

Object authority to system commands and programs

Following are several suggestions when you restrict authority to IBM-supplied objects:

- When you have more than one national language on your system, your system has more than one system (QSYS) library. Your system has a QSYSxxxx library for each national language on your system. If you are using object authority to control access to system commands, remember to secure the command in the QSYS library and in every QSYSxxx library on your system.
- The System/38™ library sometimes provides a command with function that is equivalent to the commands that you want to restrict. Be sure you restrict the equivalent command in the QSYS38 library.
- If you have the System/36™ environment, you may need to restrict additional programs. For example, the QY2FTML program provides System/36 file transfer.

Audit security functions

This chapter describes techniques for auditing the effectiveness of security on your system. People audit their system security for several reasons:

- To evaluate whether the security plan is complete.
- To make sure that the planned security controls are in place and working. This type of auditing is usually performed by the security officer as part of daily security administration. It is also performed, sometimes in greater detail, as part of a periodic security review by internal or external auditors.
- To make sure that system security is keeping pace with changes to the system environment. Some examples of changes that affect security are:
  - New objects created by system users
  - New users admitted to the system
  - Change of object ownership (authorization not adjusted)
  - Change of responsibilities (user group changed)
  - Temporary authority (not timely revoked)
  - New products installed
- To prepare for a future event, such as installing a new application, moving to a higher security level, or setting up a communications network.

The techniques described in this chapter are appropriate for all these situations. Which things you audit and how often depends on the size and security needs of your organization. The purpose of this chapter is to discuss what information is available, how to obtain it, and why it is needed, rather than to give guidelines for the frequency of audits.

This information has three parts:

- A checklist of security items that can be planned and audited.
- Information about setting up and using the audit journal provided by the system.
- Other techniques that are available to gather security information on the system.

Security auditing involves using commands on the iSeries system and accessing log and journal information on the system. You may want to create a special profile to be used by someone doing a security audit of your system. The auditor profile will need *AUDIT special authority to be able to change the audit characteristics of your system. Some of the auditing tasks suggested in this chapter require a user profile with *ALLOBJ and *SECADM special authority. Be sure that you set the password for the auditor profile to *NONE when the audit period has ended.

For more details on security auditing see Chapter 9, of the *Security Reference* book.

Analyze user profiles

You can display or print a complete list of all the users on your system with the Display Authorized Users (DSPAUTUSR) command. The list can be sequenced by profile name or group profile name. Following is an example of the group profile sequence:

```
                          Display Authorized Users

                       Password
Group       User       Last         No
Profile     Profile    Changed   Password  Text
DPTSM
            ANDERSOR   08/04/0x            Roger Anders
            VINCENTM   09/15/0x            Mark Vincent
DPTWH
            ANDERSOR   08/04/0x            Roger Anders
            WAGNERR    09/06/0x            Rose Wagner
QSECOFR
            JONESS     09/20/0x            Sharon Jones
            HARRISOK   08/29/0x            Ken Harrison
*NO GROUP
            DPTSM      09/05/0x    X       Sales and Marketing
            DPTWH      08/13/0x    X       Warehouse
            RICHARDS   09/05/0x            Janet Richards
            SMITHJ     09/18/0x            John Smith
```

Print selected user profiles

You can use the Display User Profile (DSPUSRPRF) command to create an output file, which you can process using a query tool.

```
DSPUSRPRF USRPRF(*ALL) +
          TYPE(*BASIC) OUTPUT(*OUTFILE)
```

You can use a query tool to create a variety of analysis reports of your output file, such as:

- A list of all users who have both *ALLOBJ and *SPLCTL special authority.
- A list of all users sequenced by a user profile field, such as initial program or user class.

You can create query programs to produce different reports from your output file. For example:

- List all user profiles that have any special authorities by selecting records where the field UPSPAU is not equal to *NONE.
- List all users who are allowed to enter commands by selecting records where the *Limit capabilities* field (called UPLTCP in the model database outfile) is equal to *NO or *PARTIAL.
- List all users who have a particular initial menu or initial program.
- List inactive users by looking at the date last sign-on field.

Examine large user profiles

User profiles with large numbers of authorities, appearing to be randomly spread over most of the system, can reflect a lack of security planning. Following is one method for locating large user profiles and evaluating them:

1. Use the Display Object Description (DSPOBJD) command to create an output file containing information about all the user profiles on the system:

   ```
   DSPOBJD OBJ(*ALL) OBJTYPE(*USRPRF) +
           DETAIL(*BASIC) OUTPUT(*OUTFILE)
   ```

2. Create a query program to list the name and size of each user profile, in descending sequence by size.

3. Print detailed information about the largest user profiles and evaluate the authorities and owned objects to see if they are appropriate:

```
DSPUSRPRF USRPRF(user-profile-name) +
          TYPE(*OBJAUT) OUTPUT(*PRINT)
DSPUSRPRF USRPRF(user-profile-name) +
          TYPE(*OBJOWN) OUTPUT(*PRINT)
```

   Some IBM-supplied user profiles are very large because of the number of objects they own. Listing and analyzing them is usually not necessary. However, you should check for programs adopting the authority of the IBM-supplied user profiles that have *ALLOBJ special authority, such as QSECOFR and QSYS.

For more details on security auditing see Chapter 9, of the *Security Reference* book.

Analyze object authorities

You can use the following method to determine who has authority to libraries on the system:

1. Use the DSPOBJD command to list all the libraries on the system:

```
DSPOBJD OBJ(QSYS/*ALL) OBJTYPE(*LIB) ASPDEV(*ALLAVL) OUTPUT(*PRINT)
```

   Note: Libraries in independent auxiliary storage pools that are not in AVAILABLE status will not be displayed by this command.

2. Use the Display Object Authority (DSPOBJAUT) command to list the authorities to a specific library:

```
DSPOBJAUT OBJ(QSYS/library-name) OBJTYPE(*LIB) +
          ASPDEV(asp-device-name) OUTPUT(*PRINT)
```

3. Use the Display Library (DSPLIB) command to list the objects in the library:

```
DSPLIB LIB(QSYS/library-name) ASPDEV(asp-device-name) OUTPUT(*PRINT)
```

Using these reports, you can determine what is in a library and who has access to the library. If necessary, you can use the DSPOBJAUT command to view the authority for selected objects in the library also.

Check for altered objects

You can use the Check Object Integrity (CHKOBJITG) command to look for objects that have been altered. An altered object is usually an indication that someone is attempting to tamper with your system. You may want to run this command after someone has:

- Restored programs to your system
- Used dedicated service tools (DST)

When you run the command, the system creates a database file containing information about any potential integrity problems. You can check objects owned by one profile, several different profiles, or all profiles. You can look for objects whose domain has been altered. You can also recalculate program validation values to look for objects of type *PGM, *SRVPGM, *MODULE, and *SQLPKG that have been altered.

Running the CHKOBJITG program requires *AUDIT special authority. The command may take a long time to run because of the scans and calculations it performs. You should run it at a time when your system is not busy.

Note: Profiles that own many objects with many private authorities can become very large. The size of an owner profile affects performance when displaying and working with the authority to owned objects, and when saving or restoring profiles. System operations can also be impacted. To prevent impacts to either performance or system operations, distribute ownership of objects to multiple profiles. **Do not assign all (or nearly all) objects to only one owner profile.**

Analyze programs that adopt authority

Programs that adopt the authority of a user with *ALLOBJ special authority represent a security exposure. The following method can be used to find and inspect those programs:

1. For each user with *ALLOBJ special authority, use the Display Programs That Adopt (DSPPGMADP) command to list the programs that adopt that user's authority:

   ```
   DSPPGMADP USRPRF(user-profile-name) +
             OUTPUT(*PRINT)
   ```

   Note: The topic "Print selected user profiles" on page 55 shows how to list users with *ALLOBJ authority.

2. Use the DSPOBJAUT command to determine who is authorized to use each adopting program and what the public authority is to the program:

   ```
   DSPOBJAUT OBJ(library-name/program-name) +
             OBJTYPE(*PGM) ASPDEV(library-name/program-name) +
             OUTPUT(*PRINT)
   ```

3. Inspect the source code and program description to evaluate:
   - Whether the user of the program is prevented from excess function, such as using a command line, while running under the adopted profile.
   - Whether the program adopts the minimum authority level needed for the intended function. Applications that use program failure can be designed using the same owner profile for objects and programs. When the authority of the program owner is adopted, the user has *ALL authority to application objects. In many cases, the owner profile does not need any special authorities.

4. Verify when the program was last changed, using the DSPOBJD command:

   ```
   DSPOBJD OBJ(library-name/program-name) +
           OBJTYPE(*PGM) ASPDEV(library-name/program-name) +
           DETAIL(*FULL)
   ```

Manage the audit journal and journal receivers

The auditing journal, QSYS/QAUDJRN, is intended solely for security auditing. Objects should not be journaled to the audit journal. Commitment control should not use the audit journal. User entries should not be sent to this journal using the Send Journal Entry (SNDJRNE) command or the Send Journal Entry (QJOSJRNE) API.

Special locking protection is used to ensure that the system can write audit entries to the audit journal. When auditing is active (the QAUDCTL system value is not *NONE), the system arbitrator job (QSYSARB) holds a lock on the QSYS/QAUDJRN journal. You cannot perform certain operations on the audit journal when auditing is active, such as:

- DLTJRN command
- ENDJRNxxx command

- APYJRNCHG command
- RMVJRNCHG command
- DMPOBJ or DMPSYSOBJ command
- Moving the journal
- Restoring the journal
- Operations that work with authority, such as the GRTOBJAUT command
- WRKJRN command

The information recorded in the security journal entries is described in *Security Reference* book. All security entries in the audit journal have a journal code of T. In addition to security entries, system entries also appear in the journal QAUDJRN. These are entries with a journal code of J, which relate to initial program load (IPL) and general operations performed on journal receivers (for example, saving the receiver).

If damage occurs to the journal or to its current receiver so that the auditing entries cannot be journaled, the QAUDENDACN system value determines what action the system takes. Recovery from a damaged journal or journal receiver is the same as for other journals.

You may want to have the system manage the changing of journal receivers. Specify MNGRCV(*SYSTEM) when you create the QAUDJRN journal, or change the journal to that value. If you specify MNGRCV(*SYSTEM), the system automatically detaches the receiver when it reaches its threshold size and creates and attaches a new journal receiver. This is called **System change-journal management**. See the iSeries Information Center—>Systems management–> Journal management–>Local journal management–>Manage journals for more information. See “Prerequisite and related information” on page xii for information on accessing the iSeries Information Center.

Chapter 7. Manage authority

A set of security reports are available to help you keep track of how the authority is set up on your system. When you run these reports initially, you can print everything (authority for all the files or for all the programs, for example).

After you have established your base of information, you can run the changed versions of reports regularly. The changed versions help you identify security-relevant changes on your system that require your attention. For example, you can run the report that shows the public authority for files every week. You can request only the changed version of the report. It will show you both new files on the system that are available to everyone and existing files whose public authority has changed since the last report.

Two menus are available to run security tools:

- Use the SECTOOLS menu for running programs interactively.
- Use the SECBATCH menu for running programs in batch. The SECBATCH menu has two parts: one for submitting jobs to the job queue immediately, and the other for placing jobs on the job scheduler.

If you are using iSeries Navigator, follow these steps to run the seurity tools:

1. In iSeries Navigator, expand your Server—>**Security**.
2. Right-click **Policies** and select **Explore** to display a list of policies you can create and manage.

Monitor public authority to objects

For both simplicity and performance, most systems are set up so that most objects are available to most users. Users are explicitly denied access to certain confidential, security-sensitive objects rather than having to be explicitly authorized to use every object. A few systems with high security requirements take the opposite approach and authorize objects on a need-to-know basis. On those systems, most objects are created with the public authority set to *EXCLUDE.

iSeries is an object-based system with many different types of objects. Most object types do not contain sensitive information or perform security-relevant functions. As a security administrator on an iSeries system with typical security needs, you probably want to focus your attention on objects that require protection, such as database files and programs. For other object types, you can just set public authority that is sufficient for your applications, which for most object types is *USE authority.

You can use the Print Public Authority (PRTPUBAUT) command to print information about objects that public users can access. (A **public user** is anyone with sign-on authority who does not have explicit authority to an object.) When you use the PRTPUBAUT command, you can specify the object types, and libraries or directories, that you want to examine. Options are available on the SECBATCH and SECTOOLS menus to print the Publicly Authorized Objects Report for the object types that most commonly have security implications. You can print the changed version of this report regularly to see what objects might require your attention.

© Copyright IBM Corp. 1996, 2002

Manage authority for new objects

OS/400 provides functions to help you manage the authority and ownership for new objects on your system. When a user creates a new object, the system determines the following:

- Who will own the object
- What the public authority for the object is
- Whether the object has any private authorities
- Where to put the object (what library or directory)
- Whether access to the object will be audited

The system uses system values, library parameters, and user profile parameters to make these decisions. "Assigning Authority and Ownership to New Objects" in chapter 5 of the *iSeries Security Reference* book provides several examples of the options that are available.

You can use the PRTUSRPRF command to print the user profile parameters that affect ownership and authority for new objects. Figure 5 on page 64 shows an example of this report.

Monitor authorization lists

You can group objects with similar security requirements by using an authorization list. Conceptually, an authorization list contains a list of users and the authority that the users have to the objects that are secured by the list. Authorization lists provide an efficient way to manage the authority to similar objects on the system. However, in some cases, they make it difficult to keep track of authorities to objects.

You can use the Print Private Authority (PRTPVTAUT) command to print information about authorization list authorities. Figure 3 shows a sample of the report.

```
                        Private Authorities (Full Report)
SYSTEM4
Authorization             Primary                              List  ----------Object-----------  ------------Data------------
List          Owner       Group       User        Authority  Mgt   Opr  Mgt  Exist  Alter  Ref  Read  Add  Upd  Dlt  Execute
LIST1         QSECOFR     *NONE       *PUBLIC     *EXCLUDE
LIST2         BUDNIKR     *NONE       BUDNIKR     *ALL        X     X    X     X      X     X    X     X    X    X      X
                                      *PUBLIC     *CHANGE           X                              X     X    X    X      X
LIST3         QSECOFR     *NONE       *PUBLIC     *EXCLUDE
LIST4         CJWLDR      *NONE       CJWLDR      *ALL        X     X    X     X      X     X    X     X    X    X      X
                                      GROUP1      *ALL              X    X     X      X     X    X     X    X    X      X
                                      *PUBLIC     *EXCLUDE
```

Figure 3. Private Authorities Report for Authorization Lists

This report shows the same information that you see on the Edit Authorization List (EDTAUTL) display. The advantage of the report is that it provides information about all authorization lists in one place. If you are setting up security for a new group of objects, for example, you can quickly scan the report to see if an existing authorization list meets your needs for those objects.

You can print a changed version of the report to see new authorization lists or authorization lists with authority changes since you last printed the report. You also have the option of printing a list of the objects that are secured by each authorization list. Figure 4 on page 61 shows an example of the report for one authorization list:

```
                    Display Authorization List Objects
Authorization list . . . . . :   CUSTAUTL
Library  . . . . . :     QSYS
Owner  . . . . . :   AROWNER
Primary group  . . . . . :   *NONE

                                                      Primary
Object         Library        Type         Owner     group          Text
CUSTMAS        CUSTLIB        *FILE        AROWNER   *NONE
CUSTORD        CUSTORD        *FILE        OEOWNER   *NONE
```

Figure 4. Display authorization list objects report

You can use this report, for example, to understand the effect of adding a new user to an authorization list (what authorities that user will receive).

Use authorization lists

iSeries Navigator provides security features designed to assist you in developing a security plan and policy, and configure your system to meet your company's needs. One of the functions available is the use of authorization lists.

Authorization lists have the following features.

- An authorization list group objects with similar security requirements.
- An authorization list conceptually contains a list of users and groups and the authority each has to the objects secured by the list.
- Each user and group can have a different authority to the set of object the list secures.
- Authority can be given by way of the list, rather than to individual users and groups.

Tasks that can be done using authorization lists include the following.

- Create an authorization list
- Change an authorization list.
- Add users and groups.
- Change user permissions.
- Display secured objects.

To use this function, perform the following steps:

1. From iSeries Navigator, expand your server—>Security. You will see **Authorization Lists** and **Policies**.
2. Right-click **Authorization Lists** and select **New Authorization List**. The **New Authorization List**allows you to do the following.
   - **Use:** Allows access to the object attributes and use of the object. The public may view, but not change the objects.
   - **Change:** Allows the contents of the object (with some exceptions) to be changed.
   - **All:** Allows all operations on the object, except those that are limited to the owner. The user or group can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user or group can also change ownership of the object.
   - **Exclude:**All operations on the object are prohibited. No access or operations are allowed to the object for the users and groups having this permission. Specifies the public is not allowed to use the object.

When working with authorization lists you will want to grant permissions for both objects and data. Object permissions you can choose are listed below.

- **Operational:** Provides the permission to look at the description of an object and use the object as determined by the data permission that the user or group has to the object.
- **Management:** Provides the permission to specify the security for the object, move or rename the object, and add members to the database files.
- **Existence:** Provides the permission to control the object's existence and ownership. The user or group can delete the object, free storage of the object, perform save and restore operations for the object, and transfer ownership of the object. If a user or group has special save permission, the user or group does not need object existence permission.
- **Alter** (used only for database files and SQL packages): Provides the permission needed to alter the attributes of an object. If the user or group has this permission on a database file, the user or group can add and remove triggers, add and remove referential and unique constraints, and change the attributes of the database file. If the user or group has this permission on an SQL package, the user or group can change the attributes of the SQL package. This permission is currently used only for database files and SQL packages.
- **Reference**(used only for database files and SQL packages): Provides the permission needed to reference an object from another object such that operations on that object may be restricted by the other object. If the user or group has this permission on a physical file, the user or group can add referential constraints in which the physical file is the parent. This permission is currently used only for database files.

Data permissions you can choose are listed below.

- **Read:** Provides the permission needed to get and display the contents of the object, such as viewing records in a file.
- **Add:** Provides the permission to add entries to an object, such as adding messages to a message queue or adding records to a file.
- **Update:** Provides the permission to change the entries in an object, such as changing records in a file.
- **Delete:** Provides the permission to remove entries from an object, such as removing messages from a message queue or deleting records from a file.
- **Execute:** Provides the permission needed to run a program, service program or SQL package. The user can also locate an object in a library or directory.

For more information on each process as you are creating or editing your authorization lists, use the online help available in iSeries Navigator.

Accessing Policies in iSeries Navigator

You can use iSeries Navigator to view and manage policies for your iSeries server. iSeries Navigator has five policy areas:

- **Audit policy**
  This allows you to set up monitoring for specific actions and access to specific resources on your system.
- **Security policy**
  This allows you to specify the level of security and additional options that relate to system security.
- **Password policy**
  This allows you to specify password level for the system.
- **Restore policy**
  This allows you to specify how certain objects are restored on the system.

- **Sign-on policy**
  This allows you to specify how user can sign-onto the system.

To view or change policies with iSeries Navigator, follow these steps:

1. From iSeries Navigator, expand your server—>**Security**.
2. Right-click **Policies** and select **Explore** to display a list of policies that you can create and manage. See iSeries Navigator help for specifics on these policies.

Monitor private authority to objects

SECBATCH menu options:
12 *to submit immediately* **41** *to use the job scheduler*

You can use the Print Private Authority (PRTPVTAUT) command to print a list of all the private authorities for objects of a specified type in a specified library.

You can use this report to help you detect new authorities to objects. It can also help you keep your private authority scheme from becoming convoluted and unmanageable.

Monitor access to output and job queues

Sometimes a security administrator does a great job of protecting access to files and then forgets about what happens when the contents of a file are printed. iSeries servers provide functions for you to protect sensitive output queues and job queues. You protect an output queue so that unauthorized users cannot, for example, view or copy confidential spooled files that are waiting to print. You protect job queues so that an unauthorized user cannot either redirect a confidential job to a nonconfidential output queue or cancel the job entirely.

SECBATCH menu options:
24 to submit immediately 63 *to use the job scheduler*

The *Basic system security and planning* in the Information Center and *iSeries Security Reference* books describe how to protect your output queues and job queues.

You can use the Print Queue Authority (PRTQAUT) command to print the security settings for the job queues and output queues on your system. You can then evaluate printing jobs that print confidential information and ensure that they are going to output queues and job queues that are protected.

For output queues and job queues that you consider to be security-sensitive, you can compare your security settings to the information in Appendix D of the *iSeries Security Reference* book. The tables in Appendix D tell what settings are required to perform different output queue and job queue functions.

Monitor special authorities

When users on your system have unnecessary special authorities, your efforts to develop a good object-authority scheme may be wasted. Object authority is meaningless when a user profile has *ALLOBJ special authority. A user with *SPLCTL special authority can see any spooled file on the system, no matter what efforts you make to secure your output queues. A user with *JOBCTL special authority can affect system operations and redirect jobs. A user with *SERVICE special authority may be able to use service tools to access data without going through the operating system.

SECBATCH menu options:
29 to submit immediately 68 *to use the job scheduler*

You can use the Print User Profile (PRTUSRPRF) command to print information about the special authorities and user classes for user profiles on your system. When you run the report, you have several options:
- All user profiles
- User profiles with specific special authorities
- User profiles that have specific user classes
- User profiles with a mismatch between user class and special authorities.

Figure 5 shows an example of the report that shows the special authorities for all user profiles:

```
                                    User Profile Information
  Report type  . . . . . . . . . . :   *AUTINFO
  Select by  . . . . . . . . . . . :   *SPCAUT
  Special authorities  . . . . . . :   *ALL
                    ------------Special Authorities-------------
                                *IO                                                    Group
User      Group     *ALL  *AUD  SYS  *JOB  *SAV  *SEC  *SER  *SPL  User               Group       Authority   Limited
Profile   Profiles   OBJ   IT   CFG   CTL   SYS   ADM  VICE   CTL  Class     Owner    Authority   Type        Capability
USERA     *NONE      X     X     X    X     X     X     X     X    *SECOFR   *USRPRF  *NONE       *PRIVATE    *NO
USERB     *NONE                        X     X                      *PGMR     *USRPRF  *NONE       *PRIVATE    *NO
USERC     *NONE      X     X     X    X     X     X     X     X    *SECOFR   *USRPRF  *NONE       *PRIVATE    *NO
USERD     *NONE                                                    *USER     *USRPRF  *NONE       *PRIVATE    *NO
```

Figure 5. User information report: Example 1

In addition to the special authorities, the report shows the following:
- Whether the user profile has limited capability.
- Whether the user or the user's group owns new objects that the user creates.
- What authority the user's group automatically receives to new objects that the user creates.

Figure 6 on page 65 shows an example of the report for mismatched special authorities and user classes:

```
User Profile Information
Report type  . . . . . :   *AUTINFO
Select by  . . . . . :   *MISMATCH
-------------------------------Special Authorities-------------
                               *IO                                                    Group
User      Group     *ALL  *AUD  SYS  *JOB  *SAV  *SEC  *SER  *SPL  User               Group       Authority   Limited
Profile   Profiles   OBJ   IT   CFG   CTL   SYS   ADM  VICE   CTL  Class      Owner    Authority   Type        Capability
USERX     *NONE       X                X     X                 X    *SYSOPR    *USRPRF  *NONE       *PRIVATE    *NO
USERY     *NONE                                    X               *USER      *USRPRF  *NONE       *PRIVATE    *NO
USERZ                                              X               *USER      *USRPRF  *NONE       *PRIVATE    *NO
          QPGMR                          X     X
```

Figure 6. User information report: Example 2

In Figure 6, notice the following:

- USERX has a system operator (*SYSOPR) user class but has *ALLOBJ and *SPLCTL special authorities.
- USERY has a user (*USER) user class but has *SECADM special authority.
- USERZ also has a user (*USER) class and *SECADM special authority. You can also see that USERZ is a member of the QPGMR group, which has *JOBCTL and *SAVSYS special authorities.

You can run these reports regularly to help you monitor the administration of user profiles.

Monitor user environments

One role of the user profile is to define the environment for the user, including the output queue, the initial menu, and the job description. The user's environment affects how the user sees the system and, to some extent, what the user is allowed to do. The user must have authority to the objects that are specified in the user profile. However, if your authority scheme is still in progress or is not very restrictive, the user environment that is defined in a user profile may produce results that you do not intend. Following are several examples:

SECBATCH menu options:

29 *to submit immediately* **68** *to use the job scheduler*

- The user's job description may specify a user profile that has more authority than the user.
- The user may have an initial menu that does not have a command line. However, the user's attention-key-handling program may provide a command line.
- The user may be authorized to run confidential reports. However, the user's output may be directed to an output queue that is available to users who should not see the reports.

You can use the *ENVINFO option of the Print User Profile (PRTUSRPRF) command to help you monitor the environments that are defined for system users. Figure 7 on page 66 shows an example of the report:

```
                                        User Profile Information
  Report type  . . . . . :   *ENVINFO
  Select by  . . . . . :   *USRCLS
                          Initial       Initial          Job          Message       Output        Attention
User          Current     Menu/         Program/      Description/    Queue/        Queue/        Program/
Profile       Library     Library       Library         Library       Library       Library       Library
AUDSECOFR     AUDITOR     MAIN          *NONE           QDFTJOBD      QSYSOPR       *WRKSTN       *SYSVAL
                          *LIBL                         QGPL           QSYS
USERA         *CRTDFT     OEMENU        *NONE           QDFTJOBD      USERA         *WRKSTN       *SYSVAL
                          *LIBL                         QGPL           QUSRSYS
USERB         *CRTDFT     INVMENU       *NONE           QDFTJOBD      USERB         *WRKSTN       *SYSVAL
                          *LIBL                         QGPL           QUSRSYS
USERC         *CRTDFT     PAYROLL       *NONE           QDFTJOBD      USERC         PAYROLL       *SYSVAL
                          *LIBL                         QGPL           QUSRSYS        PRPGMLIB
```

Figure 7. Print user profile-user environment example

Manage service tools

Service tools are used to configure, manage, and service your server. Service tools can be accessed from dedicated service tools (DST) or system service tools (SST). Service tools user IDs are required to access DST, SST, and to use iSeries Navigator functions for logical partition (LPAR) management and disk unit management.

DST is available when the Licensed Internal Code has been started, even if OS/400 has not been loaded. SST is available from OS/400. The following table outlines the basic differences between DST and SST.

Characteristic	DST	SST
How to access	Physical access through console during a manual IPL or by selecting option 21 on the control panel.	Access through interactive job with the ability to sign on with QSRV or the following authorizations: • Authorized to STRSST (Start SST) CL command. • Service special authority (*SERVICE) or all object special authority (*ALLOBJ). • Functional privilege to use SST.
When available	Available even when the server has limited capabilities. OS/400 is not required to access DST.	Available when OS/400 has been started. OS/400 is required to access SST.
How to authenticate	Requires service tools user ID and password.	Requires service tools user ID and password.

See the iSeries Information Center—>Security—>Service tools for information about using the Service tools to perform the following tasks:

- Access service tools with DST
- Access service tools with SST
- Access service tools with iSeries Navigator
- Create a service tools user ID
- Change the functional privileges for a service tools user ID
- Change the description for a service tools user ID

- Display a service tools user ID
- Enable or disable a service tools user ID
- Delete a service tools user ID
- Change service tools user IDs and passwords using SST or DST
- Change your service tools user ID password using STRSST
- Change service tools user IDs and passwords using
- Change Service Tools User ID (QSYCHGDS) API
- Reset the QSECOFR OS/400 user profile password
- Reset the QSECOFR service tools user ID and password
- Save service tools security data Restore service tools security data
- Create your own version of the QSECOFR service tools user ID
- Configure the service tools server for DST
- Configure the service tools server for OS/400
- Monitor service function use through DST
- Monitor service tools use through OS/400 security audit log

See “Prerequisite and related information” on page xii for information on accessing the iSeries Information Center.

Chapter 8. Use logical partitions security (LPAR)

Having multiple logical partitions on a single iSeries server could prove beneficial in the following scenarios.

- **Maintain independent systems**: Dedicating a portion of the resources (disk storage unit, processors, memory, and I/O devices) to a partition achieves logical isolation of software. Logical partitions also have some hardware fault tolerance if configured properly. Interactive and batch workloads which may not run well together on a single machine can be isolated and run efficiently in separate partitions.
- **Consolidation** : A logically partitioned system can reduce the number of iSeries server systems that are needed within an enterprise. You can consolidate several systems into a single logically partitioned system. This eliminates the need for, and expense of, additional equipment. You can shift resources from one logical partition to another as needs change.
- **Create a mixed production and test environment:** You can create a combination production and test environment. You can create a single production partition in the primary partition. For multiple production partitions, see *Creating a Multiple Production Partition Environment* below.

  A logical partition is either a test or production partition. A production partition runs the your main business applications. A failure in a production partition could significantly hinder business operations and cost the you time and money. A test partition tests software. A failure in a test partition, while not necessarily planned, will not disrupt normal business operations.
- **Create a multiple production partition environment:** You should create multiple production partitions only in your secondary partitions. In this situation, you dedicate the primary partition to partition management.
- **Hot backup:** When a secondary partition replicates to another logical partition within the same system, switching to the backup during partition failure would cause minimal inconvenience. This configuration also minimizes the effect of long save windows. You can take the backup partition off line and save, while the other logical partition continues to perform production work. You will need special software to use this hot backup strategy.
- **Integrated cluster:** Using OptiConnect/400, and high availability application software, your partitioned system can run as an integrated cluster. You can use an integrated cluster to protect your system from most unscheduled failures within a secondary partition.

Note: When setting up a secondary partition, additional considerations for card locations need to be made. If the Input/Output Processor (IOP) you select for the console also has a LAN card and the LAN card is not intended for use with Operations Console, it will be activated for use by the console and you may not be able to use it for your intended purposes. For more information on working with Operations Console, see Chapter 9, “iSeries Operations Console” on page 71.

Refer to ″Logical Partitions″ in the iSeries Information Center for more detailed information on this topic.

© Copyright IBM Corp. 1996, 2002

Manage security for logical partitions

The security-related tasks you perform on a partitioned system are the same as on a system without logical partitions. However, when you create logical partitions, you work with more than one independent system. Therefore you will have to perform the same tasks on each logical partition instead of just once on a system without logical partitions.

Here are some basic rules to remember when dealing with security on logical partitions:

- You add users to the system one logical partition at a time. You need to add users to each logical partition you want them to access.
- Limit the number of people who have authority to go to dedicated service tools (DST) and system service tools (SST) on the primary partition. Refer to the ″Managing logical partitions with Operations Navigator along with DST and SST″ topic in the iSeries Information Center for more information on DST and SST. Refer to “Manage service tools” on page 66 for information on using service tool user profiles to control access to partition activities.

  Note: You must initialize the Service Tools Server (STS) before using Operations Navigator to access LPAR functions. See the iSeries Information Center—>Security—>Service tools for related information. See “Prerequisite and related information” on page xii for information on accessing the iSeries Information Center.
- Secondary partitions cannot see or use main storage and disk units of another logical partition.
- Secondary partitions can only see their own hardware resources.
- The primary partition can see all system hardware resources in the Work with System Partitions displays of DST and SST.
- The primary partition operating system still only sees its resources available.
- The system control panel controls the primary partition. When you set the panel mode to Secure, no actions can be performed on the Work with Partition Status display from SST. To force DST from the system control panel, you must change the mode to Manual.
- When you set the operating mode of a secondary partition to secure, you restrict the usage of its Work with Partition Status in these ways:
  - You can only use DST on the secondary partition to change partition status; you cannot use SST to change partition status.
  - You can only force DST on the secondary partition from the primary partition Work with Partition Status display using either DST or SST.
  - You can only use DST on the primary partition to change a secondary partition mode from secure to any other value.

Once a secondary partition’s mode is no longer secure, you can use both DST and SST on the secondary partition to change partition status.

For more information on security on your iSeries server, refer to the Security Reference book and the Basic system security and planning pages of the iSeries Information Center.

Chapter 9. iSeries Operations Console

Operations Console allows you to use your PC to access and control your iSeries server. Operations Console includes support for remote PC dial-in to a iSeries servers without console devices, allowing remote PCs to become the consoles. When you use Operations Console, note the following:

- You can do any tasks that you could do from a traditional console from Operations Console. For example, user profiles that have *SERVICE or *ALLOBJ special authority are able to sign on to the Operations Console session, even if they are disabled.
- Operations Console uses Service Tools User Profiles and passwords to enable the connection to the iSeries server. This makes it especially important to change your Service Tools User Profiles and passwords. Hackers are likely to be familiar with the default Service Tools User Profiles userids and passwords, and could use them to attempt a remote console session to your iSeries server. See "Change known passwords" on page 25 and "Avoid default passwords" on page 30 for tips on passwords.
- To protect your information when using the Remote Console, use the call back option of Windows Dial-Up Networking.
- When setting up a secondary partition, additional considerations for card locations need to be made. If the Input/Output Processor (IOP) you select for the console also has a LAN card and the LAN card is not intended for use with Operations Console, it will be activated for use by the console and you may not be able to use it for your intended purposes.

In V5R1, Operations Console was enhanced to enable console activities to be performed across a local area network (LAN). Enhanced authentication and data encryption provide network security for console procedures. To use Operations Console with LAN connectivity, you are strongly encouraged to install the following products:

- Cryptographic Access Provider, 5722–AC2 or 5722–AC3 on your iSeries server
- Client Encryption, 5722–CE2 or 5722–CE3 on your Operations Console PC

In order for the console data to be encrypted, the iSeries server must have one of the Cryptographic Access Provider products installed **and** the PC must have one of the Client Encryption products installed.

Note: If no cryptographic products are installed, there won't be any data encryption.

The table below summarizes the encryption results of the available products:

Table 13. Encryption results

Cryptographic Access Provider on your iSeries server	**Client Encryption on your Operations Console PC**	**Resulting Data Encryption**
None	None	None
5722–AC2	5722–CE2	56 bit
5722–AC2	5722–CE3	56 bit
5722–AC3	5722–CE2	56 bit

© Copyright IBM Corp. 1996, 2002

Table 13. Encryption results (continued)

Cryptographic Access Provider on your iSeries server	Client Encryption on your Operations Console PC	Resulting Data Encryption
5722–AC3	5722–CE3	128 bit

For additional information about setting up and administering iSeries Operations Console, see the iSeries Information Center.

Operations Console security overview

Operations Console security consists of:

- console device authentication
- user authentication
- data privacy
- data integrity

Operations Console with direct connectivity has implicit device authentication, data privacy, and data integrity due to its point-to-point connection. User authentication security is required to sign on to the console display.

Console device authentication

Console device authentication assures which physical device is the console. Operations Console with direct connectivity uses a physical connection similar to a twinaxial console. Operations Console using a direct connection may be physically secured similar to a twinaxial connection to control access to the physical console device.

Operations Console with LAN connectivity uses a version of secure sockets layer (SSL) which supports device and user authentication but without using certificates. For this form of connection, device authentication is based on a service tools device profile. Refer to 73 for more details.

User authentication

User authentication provides assurance about who is using the console device. All issues related to user authentication are the same regardless of console type.

Data privacy

Data privacy provides confidence that the console data can only be read by the intended recipient. Operations Console with direct connectivity uses a physical connection similar to a twinaxial console or secure network connection for LAN connectivity to protect console data. Operations Console using a direct connection has the same data privacy of a twinaxial connection. If the physical connection is secure, the console data remains protected.

Operations Console with LAN connectivity uses a secure network connection if the appropriate cryptographic products are installed (ACx and CEx). The console session uses the strongest encryption possible depending on the cryptographic products installed on the iSeries server and the PC running Operations Console.

Note: If no cryptographic products are installed, there will not be any data encryption.

Data integrity

Data integrity provides confidence that the console data has not changed en route to the recipient. Operations Console with direct connectivity uses a physical connection similar to a twinaxial console or secure network connection for LAN connectivity to protect console data. Operations Console using a direct connection has the same data integrity of a twinaxial connection. If the physical connection is secure, the console data remains protected.

Operations Console with LAN connectivity uses a secure network connection if the appropriate cryptographic products are installed (ACx and CEx). The console session uses the strongest encryption possible depending on the cryptographic products installed on the iSeries server and the PC running Operations Console.

Note: If no cryptographic products are installed, there will not be any data encryption.

Use Operations Console with LAN connectivity

Note: Any Operations Console device can be a console, but only LAN-based configurations use the service tool user profile.

The iSeries server is shipped with a default service tools device profile of QCONSOLE with a default password of QCONSOLE. Operations Console with LAN connectivity will change the password during each successful connection. See "Use the Operations Console setup wizard" for more information.

For additional information about iSeries Operations Console with LAN connectivity, refer to the topic, Configure Operations Console with LAN Connectivity, in the Information Center.

Protect Operations Console with LAN connectivity

When using Operations Console with LAN connectivity, the items below are recommended:

- Create another service tools device profile with console attributes and store the profile information in a safe place.
- Install Cryptographic Access Provider, 5722–AC2 or 5722–AC3 on your iSeries server and Client Encryption, 5722–CE2 or 5722–CE3 on your Operations Console PC.
- Choose a non-trivial service device information password.
- Protect the Operations Console PC in the same manner you would protect a twinaxial console or an Operations Console with direct connectivity.

Use the Operations Console setup wizard

The setup wizard will add the necessary information to the PC when using Operations Console with LAN connectivity. The setup wizard asks for the service tools device profile, the service tools device profile password, and a password to protect the service tools device profile information.

Note: The service tools device profile information password is used to lock and unlock the service tools device profile information (service tools device profile and password) on the PC.

When establishing a network connection, the Operations Console setup wizard will prompt you for the service device information password to access the encrypted service tools device profile and password. You will also be prompted for a valid service tools user identification and password.

Chapter 10. Detect suspicious programs

Recent trends in computer usage have increased the likelihood that your system has programs from untrusted sources or programs that perform unknown functions. Following are examples:

- A personal computer user sometimes obtains programs from other PC users. If the PC is attached to your iSeries system, that program can affect your iSeries server.
- Users who connect to networks can also obtain programs, for example from bulletin boards.
- Hackers have become more active and renowned. They often publish their methods and their results. This can lead to imitation by normally law-abiding programmers.

These trends have led to a problem in computer security that is called a **computer virus**. A virus is a program that can change other programs to include a copy of itself. The other programs are then said to be infected by the virus. Additionally, the virus can perform other operations that can take up system resources or destroy data.

The architecture of the iSeries server provides some protection from the infectious characteristics of a computer virus. “Protect against computer viruses” describes this. An iSeries server security administrator needs to be more concerned about programs that perform unauthorized functions. The remaining topics in this chapter describe ways that someone with ill intentions might set up harmful programs to run on your system. The topics provide tips for preventing programs from performing unauthorized functions.

Security tip

Object authority is always your first line of defense. If you do not have a good plan for protecting your objects, your system is defenseless. This information discusses ways that an authorized user might try to take advantage of loop-holes in your object authority scheme.

Protect against computer viruses

A computer that has a virus infection has a program that can change other programs. The object-based architecture of iSeries makes it more difficult for a mischief-maker to produce and spread this type of virus than it is with other computer architectures. On the iSeries server, you use specific commands and instructions to work on each type of object. You cannot use a file instruction to change an operable program object (which is what most virus-creators do). Nor can you easily create a program that changes another program object. To do this requires considerable time, effort, and expertise, and it requires access to tools and documentation that are not generally available.

However, as new iSeries server functions become available to participate in the open-systems environment, some of the object-based protection functions of iSeries servers no longer apply. For example, with the integrated file system (IFS), users can directly manipulate some objects in directories, such as stream files.

© Copyright IBM Corp. 1996, 2002

Also, although iSeries server architecture makes it difficult for a virus to spread among iSeries server programs, its architecture does not prevent the iSeries server from being a virus-carrier. As a file server, the iSeries server can store programs that many PC users share. Any one of these programs might contain a virus that the iSeries server does not detect. To prevent this type of virus from infecting the PCs that are attached to your iSeries server, you must use PC virus-scan software.

Several functions exist on the iSeries server to prevent someone from using a low-level language with pointer capability to alter an operable object program:

- If your system runs at security level 40 or higher, the integrity protection includes protections against changing program objects. For example, you cannot successfully run a program that contains blocked (protected) machine instructions.
- The program validation value is also intended to protect you when you restore a program that was saved (and potentially changed) on another system. Chapter 2 in the *iSeries Security Reference* book describes the integrity protection functions for security level 40 and higher, including program validation values.

  Note: The program validation value is not foolproof, and it is not a replacement for vigilance in evaluating programs that are restored to your system.

Several tools are also available to help you detect the introduction of an altered program into your system:

- You can use the Check Object Integrity (CHKOBJITG) command to scan objects (operable objects) that meet your search values to ensure that those objects have not been altered. This is similar to a virus-scan function.
- You can use the security auditing function to monitor programs that are changed or restored. The *PGMFAIL, *SAVRST, and *SECURITY values for the authority level system value provide audit records that can help you detect attempts to introduce a virus-type program into your system. Chapter 9 and Appendix F in the *iSeries Security Reference* book provide more information about audit values and the audit journal entries.
- You can use the force create (FRCCRT) parameter of the Change Program (CHGPGM) command to re-create any program that has been restored to your system. The system uses the program template to re-create the program. If the program object has been changed after it was compiled, the system re-creates the changed object and replaces it. If the program template contains blocked (protected) instructions, the system will not re-create the program successfully.
- You can use the QFRCCVNRST (force conversion on restore) system value to recreate any program as it restored to your system. The system uses the program template to recreate the program. This system value provides several choices on which programs to recreate.
- You can use the QVFYOBJRST (verify objects on restore) system value to prevent the restore of programs that do not have a digital signature or do not have a valid digital signature. When a digital signature is not valid, it means the program has been changed since it was signed by its developer. APIs exist that allow you to sign your own programs, save files, and stream files.

For more information on signing and how it can be used to protect your system from attack, see “Object signing” on page 86.

Monitor usage of adopted authority

On an iSeries server, you can create a program that adopts the authority of the owner of the program. This means that any user who runs the program has the same authorities (private authorities and special authorities) as the user profile that owns the program.

Adopted authority is a valuable security tool when it is used correctly. "Enhance menu access control with object security" on page 50, for example, describes how to combine adopted authority and menus to help you expand beyond menu access control. You can use adopted authority to protect your important files from being changed outside of your approved application programs while you still allow queries against the files.

As security administrator, you should make sure that adopted authority is used properly:

- Programs should adopt the authority of a user profile that has only enough authority to do the necessary functions, not excessive authority. You should be particularly cautious of programs that adopt the authority of a user profile that either has *ALLOBJ special authority or owns important objects.
- Programs that adopt authority should have a specific, limited function and should not provide command-entry capability.
- Programs that adopt authority should be secured properly.
- Excessive use of adopted authority may have a negative impact on your system performance. To help you avoid performance problems, review the authority-checking flowcharts and the suggestions for using adopted authority in Chapter 5 of the *iSeries Security Reference* book.

SECBATCH menu options:
1 *to submit immediately* **40 to use the job scheduler**

You can use the Print Adopting Objects (PRTADPOBJ) command (option 21 on the SECTOOLS menu) to help you monitor the use of adopted authority on your system.

The report displays special authorities of the specified user profile, programs that adopt the user profile's authority, as well as ASP devices that use the profile's authorities. After you have established a base of information, you can print the changed version of the adopted objects report regularly. It lists new programs that adopt authority and programs that have been changed to adopt authority since you last ran the report.

If you suspect that adopted authority is being misused on your system, you can set the QAUDLVL system value to include *PGMADP. When this value is active, the system creates an audit journal entry whenever someone starts or ends a program that adopts authority. The entry includes the name of the user who started the program and the name of the program.

Limit the use of adopted authority

When an iSeries program runs, the program can use adopted authority to gain access to objects in two different ways:

- The program itself can adopt the authority of its owner. This is specified in the user profile (USRPRF) parameter of the program or service program.
- The program can use (inherit) adopted authority from a previous program that is still in the job's call stack. A program can inherit the adopted authority from previous programs even if the program itself does not adopt authority. The use adopted authority (USEADPAUT) parameter of a program or a service program controls whether the program inherits adopted authority from previous programs in the program stack.

Following is an example of how using adopted authority from previous programs works.

Assume that the ICOWNER user profile has *CHANGE authority to the ITEM file and that the public authority to the ITEM file is *USE. No other user profiles have any explicitly defined authority to the ITEM file. Table 14 shows the attributes for three programs that use the ITEM file:

Table 14. Use Adopted Authority (USEADPAUT) Example

Program Name	Program Owner	USRPRF Value	USEADPAUT Value
PGMA	ICOWNER	*OWNER	*YES
PGMB	ICOWNER	*USER	*YES
PGMC	ICOWNER	*USER	*NO

Example 1–Adopting Authority:
1. USERA runs the PGMA program.
2. The PGMA program attempts to open the ITEM file with update capability.

Result: Attempt is successful. USERA has *CHANGE access to the ITEM file because PGMA adopts ICOWNER's authority.

Example 2–Using Adopted Authority:
1. USERA runs the PGMA program.
2. The PGMA program calls the PGMB program.
3. The PGMB program attempts to open the ITEM file with update capability.

Result: Attempt is successful. Although the PGMB program does not adopt authority (*USRPRF is *USER), it allows the use of previous adopted authority (*USEADPAUT is *YES). The PGMA program is still in the program stack. Therefore, USERA gets *CHANGE access to the ITEM file because PGMA adopts ICOWNER's authority.

Example 3–Not Using Adopted Authority:
1. USERA runs the PGMA program.
2. The PGMA program calls the PGMC program.
3. The PGMC program attempts to open the ITEM file with update capability.

Result: Authority failure. The PGMC program does not adopt authority. The PGMC program also does not allow the use of adopted authority from previous programs. Although PGMA is still in the call stack, its adopted authority is not used.

Prevent new programs from using adopted authority

The passing of adopted authority to later programs in the stack provides an opportunity for a knowledgeable programmer to create a Trojan horse program.

The Trojan horse program can rely on previous programs in the stack to get the authority that it needs to perform mischief. To prevent this, you can limit which users are allowed to create programs that use the adopted authority of previous programs.

When you create a new program, the system automatically sets the USEADPAUT parameter to *YES. If you do not want the program to inherit adopted authority, you must use the Change Program (CHGPGM) command or the Change Service Program (CHGSRVPGM) to set the USEADPAUT parameter to *NO.

You can use an authorization list and the use adopted authority (QUSEADPAUT) system value to control who can create programs that inherit adopted authority. When you specify an authorization list name in the QUSEADPAUT system value, the system uses this authorization list to determine how to create new programs.

When a user creates a program or service program, the system checks the user's authority to the authorization list. If the user has *USE authority, the USEADPAUT parameter for the new program is set to *YES. If the user does not have *USE authority, the USEADPAUT parameter is set to *NO. The user's authority to the authorization list cannot come from adopted authority.

The authorization list that you specify in the QUSEADPAUT system value also controls whether a user can use a CHGxxx command to set the USEADPAUT value for a program or a service program.

Notes:

1. You do not need to call your authorization list QUESADPAUT. You can create an authority list with a different name. Then specify that authorization list for the QUSEADPAUT system value. In the commands in this example, substitute the name of your authorization list.
2. The QUSEADPAUT system value does not affect existing programs on your system. Use the CGHPGM command or the CHGSRVPGM command to set the USEADPAUT parameter for existing programs.

More Restrictive Environment: If you want most users to create new programs with the USEADPAUT parameter set to *NO, do the following:

1. To set the public authority for the authorization list to *EXCLUDE, type the following:

   ```
   CHGAUTLE AUTL(QUSEADPAUT) USER(*PUBLIC)
      AUT(*EXCLUDE)
   ```

2. To set up specific users to create programs that use the adopted authority of previous programs, type the following:

   ```
   ADDAUTLE AUTL(QUSEADPAUT) USER(user-name)
      AUT(*USE)
   ```

Less Restrictive Environment: If you want most users to create new programs with the USEADPAUT parameter set to *YES, do the following:

1. Leave the public authority for the authorization list set to *USE.
2. To prevent specific users from creating programs that use the adopted authority of previous programs, type the following:

   ```
   ADDAUTLE AUTL(QUSEADPAUT)
      USER(user-name) AUT(*EXCLUDE)
   ```

Monitor usage of trigger programs

DB2® UDB provides the capability to associate trigger programs with database files. Trigger-program capability is common across the industry for high-function database managers.

When you associate a trigger program with a database file, you specify when the trigger program runs. For example, you can set up the customer order file to run a trigger program whenever a new record is added to the file. When the customer's outstanding balance exceeds the credit limit, the trigger program can print a warning letter to the customer and send a message to the credit manager.

Trigger programs are a productive way both to provide application functions and to manage information. Trigger programs also provide the ability for someone with devious intentions to create a "Trojan horse" on your system. A destructive program may be sitting and waiting to run when a certain event occurs in a database file on your system.

Note: In history, the Trojan horse was a large hollow wooden horse that was filled with Greek soldiers. After the horse was introduced within the walls of Troy, the soldiers climbed out of the horse and fought the Trojans. In the computer world, a program that hides destructive functions is often called a Trojan horse.

SECBATCH menu options:
27 *to submit immediately* **66** *to use the job scheduler*

When your system ships, the ability to add a trigger program to a database file is restricted. If you are managing object authority carefully, the typical user will not have sufficient authority to add a trigger program to a database file. (Appendix D in the *iSeries Security Reference* book tells the authority that is required or all commands, including the Add Physical File Trigger (ADDPFTRG) command.

You can use the Print Trigger Programs (PRTTRGPGM) command to print a list of all the trigger programs in a specific library or in all libraries.

You can use the initial report as a base to evaluate any trigger programs that already exist on your system. Then, you can print the changed report regularly to see whether new trigger programs have been added to your system.

When you evaluate trigger programs, consider the following:

- Who created the trigger program? You can use the Display Object Description (DSPOBJD) command to determine this.
- What does the program do? You will have to look at the source program or talk to the program creator to determine this. For example, does the trigger program check to see who the user is? Perhaps the trigger program is waiting for a particular user (QSECOFR) in order to gain access to system resources.

After you have established a base of information, you can print the changed report regularly to monitor new trigger programs that have been added to your system.

Check for hidden programs

Trigger programs are not the only possible way to introduce a Trojan horse into your system. Trigger programs are an example of an **exit program**. When a certain event occurs, such as a file update in the case of a trigger program, the system runs the exit program that is associated with that event.

Table 15 describes other examples of exit programs that might be on your system. You should use the same methods for evaluating the use and content of these exit programs that you use for trigger programs.

Note: Table 15 is not a complete list of possible exit programs.

.

Table 15. System-Provided Exit Programs

Program name	When the program runs
User-specified name on the DDMACC network attribute.	When a user attempts to open a DDM file on your system or makes a DRDA connection.
User-specified name on the PCSACC network attribute.	When a user attempts to use Client Access™ functions using the Original Clients to access objects on your system.
User-specified name on the QPWDVLDPGM system value	When a user runs the Change Password function.
User-specified name on the QRMTSIGN system value.	When a user attempts to sign on interactively from a remote system.
QSYS/QEZUSRCLNP	When the automatic cleanup function runs.
User-specified name on the EXITPGM parameter of the CHGBCKUP command.	When you use the Operation Assistant backup function.
User-specified names on the CRTPRDLOD command.	Before and after you save, restore, or delete the product that was created with the command.
User-specified name on the DFTPGM parameter of the CHGMSGD command.	If a default program is specified for a message, the system runs the program when the message is issued. Because of the large number of message descriptions on a typical system, the use of default programs is difficult to monitor. To prevent public users from adding default programs for messages, consider setting the public authority for message files (*MSGF objects) to *USE.
User-specified name on the FKEYPGM parameter of the STREML3270 command.	When the user presses a function key during the 3270 device emulation session. The system returns control to the 3270 device emulation session when the exit program ends.
User-specified name on the EXITPGM parameter of the performance monitor commands	To process data that is collected by the following commands: STRPFRMON, ENDPFRMON, ADDPFRCOL, and CHGPFRCOL. The program runs when data collection ends.
User-specified name on the EXITPGM parameter of the RCVJRNE command.	For each journal entry or group of journal entries that it reads from the specified journal and journal receivers.
User-specified name on the QTNADDCR API.	During a COMMIT or ROLLBACK operation.
User-specified names on the QHFRGFS API.	To perform the file system functions.
User-specified name on the SEPPGM parameter of a printer device description	To determine what to print on the separator page before or after a spooled file or a print job.
QGPL/QUSCLSXT	When a database file is closed to allow the capture of file usage information.

Table 15. System-Provided Exit Programs (continued)

Program name	When the program runs
User-specified name on the FMTSLR parameter of a logical file.	When a record is written to the database file and a record format name is not included in the high-level language program. The selector program receives the record as input, determines the record format used, and returns it to the database.
User-specified name that is specified in the QATNPGM system value, the ATNPGM parameter in a user profile, or the PGM parameter of the SETATNPGM command.	When a user presses the Attention key.
User-specified name on the EXITPGM parameter of the TRCJOB command.	Before starting the Trace Job procedure.

For commands that allow you to specify an exit program, you should ensure that the command default has not been changed to specify an exit program. You should also ensure that the public authority for these commands is not sufficient to change the command default. The CHGCMDDFT command requires *OBJMGT authority to the command. You do not need *OBJMGT authority to run a command.

Evaluate registered exit programs

You can use the system registration function to register exit programs that should be run when certain events occur. To list the registration information on your system, type `WRKREGINF OUTPUT(*PRINT)`. Figure 8 shows an example of the report:

```
              Work with Registration Information
Exit point . . . . . . . . . . . . . . . :   QIBM_QGW_NJEOUTBOUND
Exit point format  . . . . . . . . . . . :   NJEO0100
Exit point registered  . . . . . . . . . :   *YES
Allow deregister . . . . . . . . . . . . :   *YES
Maximum number of exit programs  . . . . :   *NOMAX
Current number of exit programs  . . . . :   0
Preprocessing for add  . . . . . . . . . :   *NONE
  Library  . . . . . . . . . . . . . . . :
  Format . . . . . . . . . . . . . . . . :
Preprocessing for remove . . . . . . . . :   *NONE
  Library  . . . . . . . . . . . . . . . :
  Format . . . . . . . . . . . . . . . . :
Preprocessing for retrieve . . . . . . . :   *NONE
  Library  . . . . . . . . . . . . . . . :
```

Figure 8. Work with Registration Information-Example

For each exit point on the system, the report shows whether any exit programs are currently registered. When an exit point has programs that are currently registered, you can select option 8 (Display programs) from the display version of WRKREGINF to display information about the programs:

```
                    Work with Registration Information

Type options, press Enter.
5=Display exit point   8=Work with exit programs

                          Exit
     Exit                 Point
Opt  Point                Format    Registered  Text
     QIBM_QGW_NJEOUTBOUND NJEO0100     *YES     Network Job Entry outbound ex
8    QIBM_QHQ_DTAQ        DTAQ0100     *YES     Original Data Queue Server
     QIBM_QLZP_LICENSE    LICM0100     *YES     Original License Mgmt Server
     QIBM_QMF_MESSAGE     MESS0100     *YES     Original Message Server
     QIBM_QNPS_ENTRY      ENTR0100     *YES     Network Print Server - entry
     QIBM_QNPS_SPLF       SPLF0100     *YES     Network Print Server - spool
     QIBM_QNS_CRADDACT    ADDA0100     *YES     Add CRQ description activity
     QIBM_QNS_CRCHGACT    CHGA0100     *YES     Change CRQ description activi
```

Use the same method for evaluating these exit programs that you use for other exit programs and trigger programs.

Check scheduled programs

iSeries provides several methods for scheduling jobs to run at a later time, including the job scheduler. Normally, these methods do not represent a security exposure because the user who schedules the job must have the same authority that is required to submit the job to batch.

However, you should periodically check for jobs scheduled in the future. A disgruntled user who is no longer in the organization may use this method to schedule a disaster.

Restrict Save and Restore capability

Most users do not need to save and restore objects on your system. The save commands provide the possibility of copying important assets of your organization to media or to another system. Most save commands support save files that can be sent to another system (by using the SNDNETF file command) without having access to media or a save/restore device.

Restore commands provide the opportunity to restore unauthorized objects, such as programs, commands, and files, to your system. You can also restore information without access to media or to a save/restore device by using save files. Save files can be sent from another system by using the SNDNETF command or by using the FTP function.

Following are suggestions for restricting save and restore operations on your system:

- Control which users have *SAVSYS special authority. *SAVSYS special authority allows the user to save and restore objects even when the user does not have the necessary authority to the objects.
- Control physical access to save and restore devices.
- Restrict access to the save and restore commands. When you install OS/400 licensed programs, the public authority for the RSTxxx commands is *EXCLUDE. Public authority for the SAVxxx commands is *USE. Consider changing the public authority for SAVxxx commands to *EXCLUDE. Carefully limit the users that you authorize to the RSTxxx commands.

- Use the QALWOBJRST system value to restrict restoration of system-state programs, programs that adopt authority, and objects that have validation errors.
- Use the QVFYOBJRST system value to control restoring signed objects on your system.
- Use the QFRCCVNRST system value to control the recreation of certain objects being restored on your system.
- Use security auditing to monitor restore operations. Include *SAVRST in the QAUDLVL system value, and periodically print audit records that are created by restore operations. (Chapter 9 and Appendix F of the *iSeries Security Reference* book provide more information about the audit entries operations.)

Check for user objects in protected libraries

Every iSeries server job has a library list. The library list determines the sequence in which the system searches for an object if a library name is not specified with the object name. For example, when you call a program without specifying where the program is, the system searches your library list in order and runs the first copy of the program that it finds.

The *iSeries Security Reference* book provides more information about the security exposures of library lists and calling programs without a library name (called an **unqualified call**). It also provides suggestions for controlling the content of library lists and the ability to change the system library lists.

For your system to run properly, certain system libraries, such as QSYS and QGPL, must be in the library list for every job. You should use object authority to control who can add programs to these libraries. This helps to prevent someone from placing an imposter program in one of these libraries with the same name as a program that appears in a library later in the library list.

You should also evaluate who has authority to the CHGSYSLIBL command and monitor SV records in the security audit journal. A devious user could place a library ahead of QSYS in the library list and cause other users to run unauthorized commands with the same names as IBM-supplied commands.

SECBATCH menu options:
28 *to submit immediately* **67** *to use the job scheduler*

You can use the Print User Objects (PRTUSROBJ) command to print a list of user objects (objects not created by IBM) that are in a specified library. You can then evaluate the programs on the list to determine who created them and what function they perform.

User objects other than programs can also represent a security exposure when they are in system libraries. For example, if a program writes confidential data to a file whose name is not qualified, that program might be fooled into opening an imposter version of that file in a system library.

Chapter 11. Prevent and detect hacking attempts

This information is a collection of miscellaneous tips to help you to detect potential security exposures and mischief-makers.

Physical security

Your system unit represents an important business asset and a potential door into your system. Some system components inside the system are both small and valuable. You should place the system unit in a controlled location to prevent someone from removing valuable system components.

The system unit has a control panel that provides the ability to perform basic functions without a workstation. For example, you can use the control panel to do the following:

- Stop the system.
- Start the system.
- Load the operating system.
- Start service functions.

All of these activities can disrupt your system users. They also represent a potential security exposure to your system. You can use the keylock that comes with your system to control when these activities are allowed. To prevent the use of the control panel, place the keylock in the Secure position, remove the key, and store it in a safe place.

Notes:

1. If you need to perform remote IPLs or perform remote diagnostics on your system, you may need to choose another setting for the keylock. The Getting Started topic in the iSeries Information Center provides more information about keylock settings (see "Prerequisite and related information" on page xii for details).
2. Not all system models come with a keylock as a standard feature.

Monitor user profile activity

User profiles provide entry to your system. Parameters in the user profile determine a user's environment and a user's security characteristics. As a security administrator, you need to control and audit changes that occur to user profiles on your system.

You can set up security auditing so that your system writes a record of changes to user profiles. You can use the DSPAUDJRNE command to print a report of those changes.

You can create exit programs to evaluate requested actions to user profiles. Table 16 shows the exit points that are available for user profile commands.

Table 16. Exit points for user profile activity

User profile command	Exit point name
Create User Profile (CRTUSRPRF)	QIBM_QSY_CRT_PROFILE

© Copyright IBM Corp. 1996, 2002

Table 16. Exit points for user profile activity (continued)

User profile command	Exit point name
Change User Profile (CHGUSRPRF)	QIBM_QSY_CHG_PROFILE
Delete User Profile (DLTUSRPRF)	QIBM_QSY_DLT_PROFILE
Restore User Profile (RSTUSRPRF)	QIBM_QSY_RST_PROFILE

Your exit program can, for example, look for changes that might cause the user to run an unauthorized version of a program. These changes might be assigning either a different job description or a new current library. Your exit program might either notify a message queue or take some action (like changing or disabling the user profile) based on the information that the exit program receives.

The *iSeries Security Reference* book provides more information about the exit programs for user profile actions.

Object signing

All of the security precautions you take are meaningless if someone can bypass them by introducing tampered data into your system. The iSeries server has many built-in features which you can use to keep tampered software from being loaded onto your system, and to detect any such software already there. One of the techniques added in V5R1 is object signing.

Object signing is the iSeries server implementation of a cryptographic concept known as ″digital signatures.″ The idea is relatively straightforward: once a software producer is ready to ship software to customers, the producer ″signs″ the software. This signature does not guarantee that the software performs any specific function. However, it provides a way to prove that the software came from the producer who signed it, and that the software has not changed since it was produced and signed. This is particularly important if the software has been transmitted across the Internet or stored on media which you feel might have been modified.

Using digital signatures gives you greater control over which software can be loaded onto your system, and allows you more power to detect changes once it has been loaded. The new system value Verify Object Restore (QVFYOBJRST) provides a mechanism for setting a restrictive policy which requires all software loaded onto the system to be signed by known software sources. You can also choose a more open policy and simply verify signatures if they are present.

All OS/400 software, as well as the software for options and iSeries server licensed programs, has been signed by a system trusted source. These signatures help the system protect its integrity, and they are checked when fixes are applied to the system to ensure that the fix has come from a system trusted source and that it did not change in transit. These signatures can also be checked once the software is on the system. The CHKOBJITG (Check Object Integrity) command has been expanded to check signatures in addition to other integrity features of the objects on the system. Additionally, the Digital Certificate Manager has panels that you can use to check signatures on objects, including objects in the operating system.

Just as the operating system has been signed, you could use digital signatures to protect the integrity of software which is critical to your business. You might buy software which has been signed by a software provider, or you might sign software which you have purchased or written. Part of your security policy, then,

might be to periodically use CHKOBJITG, or the Digital Certificate Manager, to verify that the signatures on that software are still valid—that the objects have not changed since they were signed. You might further require that all software which gets restored on your system be signed by you or a known source. However, since most iSeries server software which is not produced by IBM is not currently signed, this might be too restrictive for your system. The new digital signature support gives you the flexibility to decide how best to protect your software integrity.

Digital signatures that protect software are just one use of digital certificates. Additional information on managing digital certificates can be found in the Digital certificate management topic in the Information Center (see "Prerequisite and related information" on page xii for details).

Monitor subsystem descriptions

When you start a subsystem on an iSeries server, the system creates an environment for work to enter the system and run. A subsystem description defines what that environment looks like. Subsystem descriptions, therefore, can provide an opportunity for devious users. A mischief-maker might use a subsystem description to start a program automatically or to make it possible to sign on without a user profile.

When you run the Revoke Public Authority (RVKPUBAUT) command, the system sets public authority to subsystem description commands to *EXCLUDE. This prevents users who are not specifically authorized (and who do not have *ALLOBJ special authority) from changing or creating subsystem descriptions.

The topics that follow provide suggestions for reviewing the subsystem descriptions that currently exist on your system. You can use the Work with Subsystem Descriptions (WRKSBSD) command to create a list of all the subsystem descriptions. When you select 5 (Display) from the list, a menu displays for the system description that you selected. It shows a list of the parts of a subsystem environment.

You select options to see details about the parts. Use the Change Subsystem Description (CHGSBSD) command to change the first two items on the menu. To change other items, use the appropriate add, remove, or change command for the entry type. For example, to change a workstation entry, use the Change Workstation Entry (CHGWSE) command.

The *Work Management* book provides more information about working with subsystem descriptions. It also lists the shipped values for IBM-supplied subsystem descriptions.

Autostart job entries

An autostart job entry contains the name of a job description. The job description may contain request data (RQSDTA) that causes a program or a command to run. For example, the RQSDTA might be `CALL LIB1/PROGRAM1`. Whenever the subsystem starts, the system will run the program PROGRAM1 in library LIB1.

Look at your autostart job entries and the associated job descriptions. Ensure that you understand the function of any program that runs automatically when a subsystem starts.

Workstation names and workstation types

When a subsystem starts, it allocates all unallocated workstations that are listed (specifically or generically) in its entries for workstation names and workstations types. When a user signs on, the user is signing on to the subsystem that has allocated the workstation.

The workstation entry tells what job description will be used when a job starts at that workstation. The job description may contain request data that causes a program or a command to run. For example, the RQSDTA parameter might be `CALL LIB1/PROGRAM1`. Whenever a user signs on to a workstation in that subsystem, the system will run PROGRAM1 in LIB1.

Look at your workstation entries and the associated job descriptions. Ensure that no one has added or updated any entries to run programs that you are not aware of.

A workstation entry might also specify a default user profile. For certain subsystem configurations, this allows someone to sign on simply by pressing the Enter key. If the security level (QSECURITY system value) on your system is less than 40, you should review your workstation entries for default users.

Job queue entries

When a subsystem starts, it allocates any unallocated job queues that are listed in the subsystem description. Job queue entries do not provide any direct security exposure. However, they do provide an opportunity for someone to tamper with system performance by causing jobs to run in unintended environments.

You should periodically review the job queue entries in your subsystem descriptions to ensure that batch jobs are running where you expect them to run.

Routing entries

A routing entry defines what a job does once it enters the subsystem. The subsystem uses routing entries for all job types: batch, interactive, and communications jobs. A routing entry specifies the following:

- The class for the job. Like job queue entries, the class that is associated with a job can affect its performance but does not represent a security exposure.
- The program that runs when the job starts. Look at the routing entries and ensure that no one has added or updated any entries to run programs that you are not aware of.

Communications entries and remote location names

When a communications job enters your system, the system uses the communications entries and the remote location name entries in the active subsystem to determine how the communications job will run. Look at the following for these entries:

- All subsystems are capable of running communications jobs. If a subsystem that you intend for communications is not active, a job that is trying to enter your system might find an entry in another subsystem description that meets its needs. You need to look at the entries in all subsystem descriptions.

- A communications entry contains a job description. The job description may contain request data that runs a command or program. Look at your communications entries and their associated job descriptions to ensure that you understand how jobs will start.
- A communications entry also specifies a default user profile that the system uses in some situations. Make sure that you understand the role of default profiles. If your system contains default profiles, you should ensure that they are profiles with minimal authority. See Chapter 13, "Secure APPC communications" for more information about default user profiles.

  You can use the Print Subsystem Description (PRTSBSDAUT) command to identify communications entries that specify a user profile name.

Prestart job entries

You can use prestart job entries to make a subsystem ready for certain kinds of jobs so that the jobs start more quickly. Prestart jobs may start when the subsystem starts or when they are needed. A prestart job entry specifies the following:

- A program to run
- A default user profile
- A job description

All of these provide the potential for security exposures. You should make sure that prestart job entries perform only authorized, intended functions.

Jobs and job descriptions

Job descriptions contain request data and routing data that can cause a specific program to run when that job description is used. When the job description specifies a program in the request data parameter, the system runs the program. When the job description specifies routing data, the system runs the program that is specified in the routing entry that matches the routing data.

The system uses job descriptions for both interactive and batch jobs. For interactive jobs, the workstation entry specifies the job description. Typically, the workstation entry value is *USRPRF, so the system uses the job description that is specified in the user profile. For batch jobs, you specify the job description when you submit the job.

You should periodically review job descriptions to make sure that they do not run unintended programs. You should also use object authority to prevent changes to job descriptions. *USE authority is sufficient to run a job with a job description. A typical user does not need *CHANGE authority to job descriptions.

SECBATCH menu options:
15 *to submit immediately* **54** *to use the job scheduler*

Job descriptions can also specify what user profile the job should run under. With security level 40 and higher, you must have *USE authority to the job description and to the user profile that is specified in the job description. With security levels lower than 40, you need *USE authority only to the job description.

You can use the Print Job Description Authority (PRTJOBDAUT) command to print a list of job descriptions that specify user profiles and have public authority of *USE.

The report shows the special authorities of the user profile that is specified in the job description. The report includes the special authorities of any group profiles that the user profile has. You can use the following command to display the user profile's private authorities:

```
DSPUSRPRF USRPRF(profile-name) TYPE(*OBJAUT)
```

The job description specifies the library list that the job uses when it runs. If someone can change a user's library list, that user might run an unintended version of a program in a different library. You should periodically review the library lists that are specified in the job descriptions on your system.

Finally, you should ensure that the default values for the Submit Job (SBMJOB) command and the Create User Profile (CRTUSRPRF) command have not been changed to point to unintended job descriptions.

Architected transaction program names

Some communications requests send a specific type of signal to your system. This request is called an **architecture transaction program name (TPN)** because the name of the transaction program is part of the APPC architecture for the system. A request for display station pass-through request is an example of an architecture TPN. Architecture TPNs are a normal way for communications to function and do not necessarily represent a security exposure. However, architecture TPNs may provide an unexpected entrance into your system.

Some TPNs do not pass a profile on the request. If the request becomes associated with a communications entry whose default user is *SYS, the request may be initiated on your system. However, the *SYS profile can run system functions only, not user applications.

If you do not want architecture TPNs to run with a default profile, you can change the default user from *SYS to *NONE in communications entries. "Architected TPN requests" on page 91 lists the architecture TPNs and the associated user profiles.

If you do not want a specific TPN to run on your system at all, do the following:

1. Create a CL program that accepts several parameters. The program should perform no function. It should simply have the Declare (DCL) statements for parameters and then end.

2. Add a routing entry for the TPN to each subsystem that has communications entries or remote location name entries. The routing entry should specify the following:
   - A *Compare value* (CMPVAL) value equal to the program name for the TPN (see Architected TPN requests) with a starting position of 37.
   - A *Program to call* (PGM) value equal to the name of the program that you created in step 1 on page 90. This prevents the TPN from locating another routing entry, such as *ANY.

Several TPNs already have their own routing entry in the QCMN subsystem. These have been added for performance reasons.

Architected TPN requests

Table 17. Programs and users for TPN requests

TPN request	Program	User profile	Description
X'30F0F8F1'	AMQCRC6A	*NONE	Message queuing
X'06F3F0F1'	QACSOTP	QUSER	APPC sign-on transaction program
X'30F0F2D1'	QANRTP	QADSM	ADSM/400 APPC configuration
X'30F0F1F9'	QCNPCSUP	*NONE	Shared folders
X'07F0F0F1'	QCNTEDDM	QUSER	DDM
X'07F6C4C2'	QCNTEDDM	QUSER	Remote SQL–DRDA1
X'30F0F7F7'	QCQNRBAS	QSVCCS	SNA CC_Server
X'30F0F1F4'	QDXPRCV	QUSER	DSNX–PC receiver
X'30F0F1F3'	QDXPSEND	QUSER	DSNX–PC sender
X'30F0F2C4'	QEVYMAIN	QUSER	ENVY**/400 Server
X'30F0F6F0'	QHQTRGT	*NONE	PC data queue
X'30F0F8F0'	QLZPSERV	*NONE	Client Access license manager
X'30F0F1F7'	QMFRCVR	*NONE	PC message receiver
X'30F0F1F8'	QMFSNDR	*NONE	PC message sender
X'30F0F6F6'	QND5MAIN	QUSER	APPN® 5394 workstation controller
DB2DRDA	QCNTEDDDM	QUSER	DB2DRDA
APINGD	QNMAPINGD	QUSER	APINGD
X'30F0F5F4'	QNMEVK	QUSER	System management utilities
X'30F0F2C1'	QNPSERVR	*NONE	PWS-I network print server
X'30F0F7F9'	QOCEVOKE	*NONE	Cross-system calendar
X'30F0F6F1'	QOKCSUP	QDOC	Directory shadowing
X'20F0F0F7'	QOQSESRV	QUSER	DIA Version 2
X'20F0F0F8'	QOQSESRV	QUSER	DIA Version 2
X'30F0F5F1'	QOQSESRV	QUSER	DIA Version 2
X'20F0F0F0'	QOSAPPC	QUSER	DIA Version 1
X'30F0F0F5'	QPAPAST2	QUSER	S/36—S/38 pass-through
X'30F0F0F9'	QPAPAST2	QUSER	Printer pass-through
X'30F0F4F6'	QPWFSTP0	*NONE	Shared Folders Type 2
X'30F0F2C8'	QPWFSTP1	*NONE	Client Access file server

Table 17. Programs and users for TPN requests (continued)

TPN request	Program	User profile	Description
X'30F0F2C9'	QPWFSTP2	*NONE	Windows** Client Access file server
X'30F0F6F9'	QRQSRVX	*NONE	Remote SQL–converged server
X'30F0F6F5'	QRQSRV0	*NONE	Remote SQL without commit
X'30F0F6F4'	QRQSRV1	*NONE	Remote SQL without commit
X'30F0F2D2'	QSVRCI	QUSER	SOC/CT
X'21F0F0F8'	QS2RCVR	QGATE	SNADS FS2 receiver
X'21F0F0F7'	QS2STSND	QGATE	SNADS FS2 sender
X'30F0F1F6'	QTFDWNLD	*NONE	PC transfer function
X'30F0F2F4'	QTIHNPCS	QUSER	TIE function
X'30F0F1F5'	QVPPRINT	*NONE	PC virtual print
X'30F0F2D3'	QWGMTP	QWGM	Ultimedia Mail/400 Server
X'30F0F8F3'	QZDAINIT	QUSER	PWS-I data access server
X'21F0F0F2'	QZDRCVR	QSNADS	SNADS receiver
X'21F0F0F1'	QZDSTSND	QSNADS	SNADS sender
X'30F0F2C5'	QZHQTRG	*NONE	PWS-I data queue server
X'30F0F2C6'	QZRCSRVR	*NONE	PWS-I remote command server
X'30F0F2C7'	QZSCSRVR	*NONE	PWS-I central server

Methods for Monitoring Security Events

Setting up security is not a one-time effort. You need to constantly evaluate both the changes on your system and your security failures. Then make adjustments to your security environment to respond to what you have discovered.

The security reports help you to monitor security-relevant changes that occur on your system. Following are other system functions that you can use to help you to detect security failures or exposures:

- Security auditing is a powerful tool that you can use to observe many different types of security-relevant events that occur on your system. For example, you can set up the system to write an audit record every time a user opens a particular database file for updating. You can audit all changes to system values. You can audit actions that happen when users restore objects.

  Chapter 9 in the *iSeries Security Reference* book provides complete information about the security auditing function. You can use the Change Security Auditing (CHGSECAUD) command to set up security auditing on your system. You can also use the Display Audit Journal Entries (DSPAUDJRNE) command to print selected information from the security audit journal.
- You can create the QSYSMSG message queue to capture critical system-operator messages. The QSYSOPR message queue receives many messages of varying importance throughout a typical business day. Critical, security-relevant messages may be overlooked because of the sheer volume of messages in the QSYSOPR message queue.

  If you create a QSYSMSG message queue in the QSYS library on your system, the system automatically directs certain critical messages to the QSYSMSG message queue instead of to the QSYSOPR message queue.

Either you can create a program to monitor the QSYSMSG message queue, or you can assign it in break mode to yourself or to another trusted user.

Part 4. Applications and network communications

© Copyright IBM Corp. 1996, 2002

Chapter 12. Use Integrated File System to secure files

The integrated file system provides you with multiple ways to store and view information on the iSeries server. The integrated file system is a part of the OS/400 operating system that supports stream input and output operations. It provides storage management methods that are similar to (and compatible with) personal computer operating systems and UNIX® operating systems.

With the integrated file system, all objects on the system can be viewed from the perspective of a hierarchical directory structure. However, in most cases, users view objects in the way that is most common for a particular file system. For example, ″traditional″ iSeries objects are in the QSYS.LIB file system. Typically, users view these objects from the perspective of libraries. Users typically view objects in the QDLS file system from the perspective of documents within folders. The root (/), QOpenSys, and user-defined file systems present a structure of hierarchical (nested) directories.

As a security administrator, you need to understand the following:

- Which file systems are used on your system
- The unique security characteristics of each file system

The topics that follow provide some general considerations for the security of the integrated file system.

The Integrated File System approach to security

The root file system acts as an umbrella (or a foundation) for all other file systems on iSeries servers. At a high level, it provides an integrated view of all of the objects on the system. Other file systems that can exist on iSeries servers provide varying approaches to object management and integration, depending on the underlying purpose of each file system. The QOPT (optical) file system, for example, allows iSeries applications and servers (including the iSeries Access for Windows file server) to access the CD-ROM drive on the iSeries server. Similarly, the QFileSvr.400 file system allows applications to access integrated file system data on remote iSeries servers. The QLANSrv file server allows access to files stored on Integrated xSeries Server for iSeries or other connected servers in the network.

The security approach for each file system depends on the data that the file system makes available. The QOPT file system, for example, does not provide object-level security because no technology exists to write authority information to a CD-ROM. For the QFileSvr.400 file system, access control occurs at the remote system (where the files are physically stored and managed). For file systems like QLANSrv, the Integrated xSeries Server for iSeries provides access control. Despite the differing security models, many file systems support consistent management of access control through the integrated file system commands, such as Change Authority (CHGAUT) and Change Owner (CHGOWN).

Here are some tips related to the nooks and crannies of integrated file system security. The integrated file system is designed to follow POSIX standards as closely as possible. This leads to some interesting behavior where iSeries server authority and POSIX permissions are ″blended″:

© Copyright IBM Corp. 1996, 2002

1. Do not remove the private authority for a user to a directory owned by that user, even if that user is authorized through the public authority, a group, or authorization list. When working with libraries or folders in the standard iSeries server security model, removing the owner's private authority would reduce the amount of authority information stored for a user profile and would not affect other operations. But, because of the way the POSIX standard defines permission inheritance for directories, the owner of a newly-created directory will have the same object authorities to that directory as the owner of the parent has to the parent, even if the owner of the newly-created directory has other private authorities to the parent. That may be hard to understand, so here's an example: USERA owns directory /DIRA, but USERA's private authorities have been removed. USERB has private authority to /DIRA. USERB creates directory /DIRA/DIRB. Because USERA has no object authorities to /DIRA, USERB will have no object authorities to /DIRA/DIRB. USERB will be unable to rename or delete /DIRA/DIRB without further action to change USERB's object authorities. This also comes into play when creating files with the open() API using the O_INHERITMODE flag. If USERB created a file /DIRA/FILEB, USERB would have no object authorities AND no data authorities to it. USERB could not write to the new file.
2. Adopted authority is not honored by most physical file systems. This includes the root (/), QOpenSys, QDLS, and user-defined file systems.
3. Any objects are owned by the user profile which created the objects, even if the OWNER field of the user profile is set to *GRPPRF.
4. Many file system operations require *RX data authority to every component of the path, including the root (/) directory. When experiencing authority problems, make sure to check the user's authorization to the root itself.
5. Displaying or retrieving the current working directory (DSPCURDIR, getcwd(), etc.) requires *RX data authority to every component in the path. However, changing the current working directory (CD, chdir(), etc.) only requires *X data authority to every component. Therefore, a user may change the current working directory to a certain path and then be unable to display that path.
6. The intent of the COPY command is to duplicate an object. The authority settings on the new file will be the same as the original except for the owner. The intent of the CPYTOSTMF command, however, is simply to duplicate data. The authority settings on the new file cannot be controlled by the user. The creator/owner will have *RWX data authority, but the group and public authorities will be *EXCLUDE. The user must use another means (CHGAUT, chmod(), etc.) to assign the desired authorities.
7. A user must be the owner or have *OBJMGT object authority to an object to retrieve authority information about the object. This pops up in some unexpected places, like COPY, which must retrieve the authority information on the source object to set the equivalent authorities on the target object.
8. When changing the owner or group of an object, the user must not only have appropriate authority to the object, but also must have *ADD data authority to the new owner/group user profile and *DELETE data authority to the old owner/group profile. These data authorities are not related to the file system data authorities. These data authorities can be displayed using the DSPOBJAUT command and changed using the EDTOBJAUT command. This also pops up unexpectedly on COPY when it tries to set the group ID for a new object.
9. The MOV command is prone to puzzling authority errors, especially when moving from one physical file system to another, or when performing data conversion. In these cases, the move actually becomes a copy-and-delete operation. Therefore, the MOV command can be affected by all of the same

authority considerations as the COPY command (see 7 and 8 above) and the RMVLNK command, in addition to other specific MOV considerations.

Following sections provide you with some considerations for several representative file systems. For more information about a specific file system on your iSeries server, you will need to consult the documentation for the licensed program that uses the file system.

Root (/), QOpenSys, and user-defined file systems

Following are security considerations for the root, QOpenSys, and user-defined file systems.

How authority works

The root, QOpenSys, and user-defined file systems provide a blending of iSeries server, PC, and UNIX** capabilities both for object management and for security. When you use the integrated file system commands from an iSeries server session (WRKAUT and CHGAUT), you can set all the normal iSeries server object authorities. This includes the *R, *W, and *X authorities that are compatible with Spec 1170 (UNIX-type operating systems).

Note: The root, QOpenSys, and user-defined file systems are functionally equivalent. The QOpenSys file system is case-sensitive. The root file system is not. User-defined file systems can be defined as case-sensitive. Because these file systems have the same security characteristics, you can assume in the topics that follow that their names are used interchangeably.

When you access the root file system as an administrator from a PC session, you can set object attributes that the PC uses to restrict certain types of access:
- System
- Hidden
- Archive
- Read-only

These PC attributes are in addition to, not replacements for, iSeries server object authority values.

When a user attempts to access an object in the root file system, OS/400 enforces all of the object authority values and attributes for the object, whether or not those authorities are "visible" from the user's interface. For example, assume that the read-only attribute for an object is set on. A PC user cannot delete the object through a iSeries Access interface. An iSeries server user with a fixed function workstation cannot delete the object either, even if the iSeries server user has *ALLOBJ special authority. Before the object can be deleted, an authorized user must use a PC function to reset the read-only value to `off`. Similarly, a PC user might not have sufficient OS/400 authority to change the PC-relevant security attributes of an object.

UNIX-type applications that run on iSeries servers use UNIX-like application programming interfaces (APIs) to access data in the root file system. With UNIX-like APIs, applications can recognize and maintain the following security information:
- Object owner
- Group owner (iSeries server primary group authority)
- Read (files)
- Write (change contents)

- Execute (run programs or search directories)

The system maps these data authorities to existing iSeries server object and data authorities:

- Read (*R) = *OBJOPR and *READ
- Write (*W) = *OBJOPR, *ADD, *UPD, *DLT
- Execute (*X) = *OBJOPR and *EXECUTE

The concepts for other object authorities (*OBJMGT, *OBJEXIST, *OBJALTER, and *OBJREF) do not exist in a UNIX-type environment.

However, these object authorities do exist for all of the objects in the root file system. When you create an object using a UNIX-like API, that object inherits these authorities from the parent directory, resulting in the following:

- The new object's owner has the same object authority as the parent directory's owner.
- The new object's primary group has the same object authority as the parent directory's primary group.
- The new object's public has the same object authority as the parent directory's public.

The new object's data authority for owner, primary group, and public are specified on the API with the mode parameter. When all of the object authorities are set 'on', you get the authority behavior that you would expect in a UNIX-type environment. It is best to leave them set 'on', unless you do not want the POSIX-like behavior.

When you run applications that use UNIX-like APIs, the system enforces all object authorities, whether or not they are "visible" to UNIX-type applications. For example, the system will enforce the authority of authorization lists even though the concept of authorization lists does not exist in UNIX-type operating systems.

When you have a mixed-application environment, you need to ensure that you do not make authority changes in one environment that will break your applications in another environment.

Work with security for the Root (/), QOpenSys, and user-defined file systems

With the introduction of the integrated file system, iSeries servers also provided a new set of commands for working with objects in multiple file systems. This command set includes commands for working with security:

- Change Auditing (CHGAUD)
- Change Authority (CHGAUT)
- Change Owner (CHGOWN)
- Change Primary Group (CHGPGP)
- Display Authority (DSPAUT)
- Work with Authority (WRKAUT)

These commands group the underlying data and object authorities into the UNIX-like authority subsets:

***RWX** Read/write/execute
***RW** Read/write
***R** Read
***WX** Write/execute
***W** Write
***X** Execute

In addition, UNIX-like APIs are available to work with security.

Public authority to the root directory

When your system ships, the public authority to the root directory is *ALL (all object authorities and all data authorities). This setting provides flexibility and compatibility with both what UNIX-like applications expect and what typical iSeries server users expect. An iSeries server user with command-line capability can create a new library in the QSYS.LIB file system simply by using the CRTLIB command. Normally, authority on a typical iSeries server allows this. Similarly, with the shipped setting for the root file system, a typical user can create a new directory in the root file system (just like you can create a new directory on your PC).

As a security administrator, you must educate your users about adequately protecting the objects that they create. When a user creates a library, probably the public authority to the library should not be *CHANGE (the default). The user should set public authority either to *USE or to *EXCLUDE, depending on the contents of the library.

If your users need to create new directories in the root (/), QOpenSys, or user-defined file systems, you have several security options:

- You can educate your users to override the default authority when they create new directories. The default is to inherit authority from the immediate parent directory. In the case of a newly created directory in the root directory, by default the public authority will be *ALL.
- You can create a "master" subdirectory under the root directory. Set the public authority on that master directory to an appropriate setting for your organization. Then instruct users to create any new personal directories in this master subdirectory. Their new directories will inherit its authority.
- You can consider changing the public authority for the root directory to prevent users from creating objects in that directory. (Remove *W, *OBJEXIST, *OBJALTER, *OBJREF, and *OBJMGT authorities.) However, you need to evaluate whether this change will cause problems for any of your applications. You might, for example, have UNIX-like applications that expect to be able to delete objects from the root directory.

Print private authorities objects (PRTPVTAUT) command

The Print Private Authorities (PRTPVTAUT) command allows you to print a report of all the private authorities for objects of a specified type in a specified library, folder, or directory. The report lists all objects of the specified type and the users that are authorized to the object. This is a way to check for different sources of authority to objects.

This command prints three reports for the selected objects. The first report (Full Report) contains all of the private authorities for each of the selected objects. The second report (Changed Report) contains additions and changes to the private authorities to the selected objects if the PRTPVTAUT command was previously run for the specified objects in the specified library, folder, or directory. Any new objects of the selected type, new authorities to existing objects, or changes to existing authorities to the existing objects are listed in the 'Changed Report'. If the PRTPVTAUT command was not previously run for the specified objects in the specified library, folder, or directory, there will be no 'Changed Report'. If the

command has been previously run but no changes have been made to the authorities on the objects, then the 'Changed Report' is printed but there are no objects listed.

The third report (Deleted Report) contains any deletions of privately authorized users from the specified objects since the PRTPVTAUT command was previously run. Any objects that were deleted or any users that were removed as privately authorized users are listed in the 'Deleted Report'. If the PRTPVTAUT command was not previously run, there will be no 'Deleted Report'. If the command has been previously run but no delete operations have been done to the objects, then the 'Deleted Report' is printed but there are no objects listed.

Restriction: You must have *ALLOBJ special authority to use this command.

Examples:

This command creates the full, changed, and deleted reports for all file objects in the PAYROLLLIB:

```
PRTPVTAUT OBJTYPE(*FILE) LIB(PAYROLLLIB)
```

This command creates the full, changed, and deleted reports for all the stream file objects in the directory garry:

```
PRTPVTAUT OBJTYPE(*STMF) DIR(/GARRY) SCHSUBDIR(*NO)
```

This command creates the full, changed, and deleted reports for all the stream file objects in the subdirectory structure that starts at the directory garry:

```
PRTPVTAUT OBJTYPE(*STMF) DIR(/GARRY) SCHSUBDIR(*YES)
```

Print publicly authorized objects (PRTPUBAUT) command

The Print Publicly Authorized Objects (PRTPUBAUT) command allows you to print a report of the specified objects that do not have public authority of *EXCLUDE. For *PGM objects, only the programs that do not have public authority of *EXCLUDE that a user can call (the program is either user domain or the system security level (QSECURITY system value) is 30 or below) will be included in the report. This is a way to check for objects that every user on the system is authorized to access.

This command will print two reports. The first report (Full Report) will contain all of the specified objects that do not have public authority of *EXCLUDE. The second report (Changed Report) will contain the objects that now do not have public authority of *EXCLUDE that did have public authority of *EXCLUDE or did not exist when the PRTPUBAUT command was previously run. If the PRTPUBAUT command was not previously run for the specified objects and library, folder, or directory, there will be no 'Changed Report'. If the command has been previously run, but no additional objects do not have public authority of *EXCLUDE, then the 'Changed Report' will be printed but there will be no objects listed.

Restrictions: You must have *ALLOBJ special authority to use this command.

Examples:

This command creates the full, and changed reports for all the file objects in the library GARRY that do not have a public authority of *EXCLUDE:

```
PRTPUBAUT OBJTYPE(*FILE) LIB(GARRY)
```

This command creates the full, changed, and deleted reports for all the stream file objects in the subdirectory structure that starts at the directory garry that do not have a public authority of *EXCLUDE:

```
PRTPUBAUT OBJTYPE(*STMF) DIR(GARRY) SCHSUBDIR(*YES)
```

Restrict access to the QSYS.LIB file system

Because the root file system is the umbrella file system, the QSYS.LIB file system appears as a subdirectory within the root directory. Therefore, any PC user with access to your iSeries server can manipulate objects stored in iSeries server libraries (the QSYS.LIB file system) with normal PC commands and actions. A PC user could, for example, drag a QSYS.LIB object (such as the library with your critical data files) to the shredder.

As you learned in "Root (/), QOpenSys, and user-defined file systems" on page 99, the system enforces all object authority whether or not it is visible to the interface. Therefore, a user cannot shred (delete) an object unless the user has *OBJEXIST authority to the object. However, if your iSeries depends on menu access security rather than object security, the PC user might very well discover objects in the QSYS.LIB file system that are available for shredding.

As you expand the uses of your system and the different methods of access that you provide, you will soon discover that menu access security is not sufficient. Chapter 6, "Protect information assets with object authority" on page 49 discusses your strategies for supplementing menu access control with object security. However, iSeries servers also provide a simple way for you to prevent access to the QSYS.LIB file system through the root file system directory structure. You can use the QPWFSERVER authorization list to control which users can access the QSYS.LIB file system through the root directory.

When a user's authority to the QPWFSERVER authorization list is *EXCLUDE, the user cannot enter the QSYS.LIB directory from the root directory structure. When a user's authority is *USE, the user can enter the directory. Once the user has authority to enter the directory, normal object authority applies for any action the user attempts to perform on an object within the QSYS.LIB file system. In other words, the authority to the QPWFSERVER authorization list acts like a door to the entire QSYS.LIB file system. For the user with *EXCLUDE authority, the door is locked. For the user with *USE authority (or any greater authority), the door is open.

For most situations, users do not need to use a directory interface to access objects in the QSYS.LIB file system. Probably, you will want to set the public authority to the QPWFSERVER authorization list to *EXCLUDE. Keep in mind, that authority to the authorization list opens or closes the door to all libraries within the QSYS.LIB file system, including user libraries. If you encounter users who object to this exclusion, you can evaluate their requirements on an individual basis. If appropriate, you can explicitly authorize an individual user to the authorization list. However, you need to ensure that the user has appropriate authority to objects within the QSYS.LIB file system. Otherwise, the user might unintentionally delete objects or entire libraries.

Notes:

1. When your system ships, the public authority to the QPWFSERVER authorization list is *USE.

2. If you explicitly authorize an individual user, the authorization list controls access only with iSeries Access file serving, NetServer file serving and file serving between iSeries servers. This does not prevent access to the same directories via FTP, ODBC, and other networks.

Secure directories

To access an object within the root file system, you read through the entire path to that object. To search a directory, you must have *X (*OBJOPR and *EXECUTE) authority to that directory. Assume, for example, that you want to access the following object:

```
/companya/customers/custfile.dat
```

You must have *X authority to the `companya` directory and to the `customers` directory.

With the root file system, you can create a symbolic link to an object. Conceptually, a symbolic link is an alias for a path name. Usually, it is shorter and easier to remember than the full path name. A symbolic link does not, however, create a different physical path to the object. The user still needs *X authority to every directory and subdirectory in the physical path to the object.

For objects in the root file system, you can use directory security just as you might use library security in the QSYS.LIB file system. You can, for example, set the public authority of a directory to *EXCLUDE to prevent public users from accessing any objects within that tree.

Security for new objects

When you create a new object in the root file system, the interface that you use to create it determines its authorities. For example, if you use the CRTDIR command and its defaults, the new directory inherits all of the authority characteristics of its parent directory, including private authorities, primary group authority, and authorization list association. The following sections describe how authorities are determined for each type of interface.

Authority comes from the immediate parent directory, not from directories higher up in the tree. Therefore, as a security administrator, you need to view the authority that you assign to directories in a hierarchy from two perspectives:

- How the authority affects access to objects in the tree (like library authority).
- How the authority affects newly created objects (like the CRTAUT value for libraries).

Recommendation: You may want to give users who work in the integrated file system a home directory (for example, /home/usrxxx), then set the security appropriately (such as PUBLIC *EXCLUDE). Any directories the user creates under their home directory will then inherit the authorities.

Following are the descriptions of authority inheritance for different interfaces:

Use the Create Directory command

When you create a new subdirectory by using the CRTDIR command, you have two options for specifying authority:

- You can specify the public authority (data authority, object authority, or both).
- You can specify *INDIR for the data authority, object authority, or both. When you specify *INDIR for both data authority and object authority, the system makes an exact copy of all the authority information from the parent directory to the new object, including authorization list, primary group, public authority, and private authorities. (The system does not copy private authority that the QSYS profile or the QSECOFR profile has to the object.)

Create a directory with an API

When you create a directory by using the mkdir() API, you specify the data authorities for the owner, the primary group, and public (using the authority map of *R, *W, and *X). The system uses the information in the parent directory to set the object authorities for the owner, primary group, and public.

Because UNIX-type operating systems do not have the concept of object authorities, the mkdir() API does not support specifying object authorities. If you want different object authorities, you can use the iSeries server command (CHGAUT). However, when you remove some object authorities, the UNIX-like application might not work as you expect it to work.

Create a stream file with the open() or creat() API

When you use the creat() API to create a stream file, you can specify the data authorities for the owner, the primary group, and public (using the UNIX-like authorities of *R, *W, and *X). The system uses the information in the parent directory to set the object authorities for the owner, primary group, and public.

You can also specify these authorities when you use the open() API to create a stream file. Alternatively, when you use the open() API you can specify that the object should inherit all authorities from the parent directory. This is called `inherit mode`. When you specify inherit mode, the system then creates a complete match for the parent authorities, including authorization list, primary group, public authority, and private authorities. This option works like specifying *INDIR on the CRTDIR command.

Create an object by using a PC interface

When you use a PC application to create an object in the root file system, the system automatically inherits all authority from the parent directory. This includes authorization list, primary group, public authority, and private authorities. PC applications do not have any equivalent to specifying authority when you create an object.

QFileSvr.400 file system

With the QFileSvr.400 file system, a user (USERX) on one iSeries system (SYSTEMA) can access data on another connected iSeries system (SYSTEMB). The USERX has an interface that is just like the Client Access interface. The remote iSeries server (SYSTEMB) appears as a directory with all its file systems as subdirectories.

When USERX attempts to access SYSTEMB with this interface, SYSTEMA sends USERX's user profile name and encrypted password to SYSTEMB. The same user profile and password must exist on SYSTEMB or SYSTEMB rejects the request.

If SYSTEMB accepts the request, USERX appears to SYSTEMB just like any Client Access user. The same authority-checking rules apply to any actions that USERX attempts.

As a security administrator, you need to be aware that the QFileSvr.400 file system represents another possible door to your system. You cannot assume that you are limiting your remote users to an interactive sign on with display station passthrough. If you have the QSERVER subsystem running and your system is connected to another iSeries system, remote users can access your system as if they are on a local PC running Client Access. More than likely, your system will have a connection that needs to have the QSERVER subsystem running. This is yet another reason why a good object authority scheme is essential.

Network file system

The Network File System (NFS) provides access to and from systems that have NFS implementations. NFS is an industry-standard method for sharing information among users on networked systems. Most major operating system (including PC operating systems) provide NFS. For UNIX systems, NFS is the primary method for accessing data. iSeries servers can act as both an NFS client and an NFS server.

When you are the security administrator of an iSeries system that acts as an NFS server, you need to understand and manage the security aspects of NFS. Following are suggestions and considerations:

- You must explicitly start the NFS server function by using the STRNFSSVR command. Control who has authority to use this command.
- You make a directory or an object available to NFS clients by `exporting` it. Therefore, you have very specific control over which parts of your system you will make available to NFS clients in your network.
- When you export, you can specify which clients have access to the objects. You identify a client by system name or IP address. A client can be an individual PC or an entire iSeries server or UNIX system. In NFS terminology, the client (IP address) is called a `machine`.
- When you export, you can specify read-only access or read/write access for each machine that has access to an exported directory or object. In most cases, you will probably want to provide read-only access.
- The NFS does not provide password protection. It is designed and intended for data sharing within a trusted community of systems. When a user requests access, the server receives the user's `uid`. Following are some `uid` considerations:
  - The iSeries server attempts to locate a user profile with the same `uid`. If it finds a matching `uid`, it uses the credentials of the user profile. Credentials is an NFS term to describe using the authority of a user. This is similar to profile swapping in other iSeries server applications.
  - When you export a directory or object, you can specify whether you will allow access by a profile with root authority. The NFS server on iSeries servers equates root authority to *ALLOBJ special authority. If you specify that you will not allow root authority, an NFS user with a `uid` that maps to a user profile with *ALLOBJ special authority will not be able to will not be able to access the object under that profile. Instead, if anonymous access is allowed, the requester will be mapped to the anonymous profile.
  - When you export a directory or object, you can specify whether you will allow anonymous requests. An anonymous request is a request with a `uid` that does not match any `uid` on your system. If you choose to allow anonymous requests, the system maps the anonymous user to the

IBM-supplied QNFSANON user profile. This user profile does not have any special authorities or explicit authority. (On the export, you can specify a different user profile for anonymous requests if you want.)

- When your iSeries server participates in an NFS network (or any network with UNIX systems that depend on uids), you probably need to manage your own uids instead of letting the system assign them automatically. You will need to coordinate uids with other systems in your network.

  You might discover that you need to change uids (even for IBM-supplied user profiles) to have compatibility with other systems in your network. A program is available to make it simpler to change the uid for a user profile. (When you change the uid for a user profile, you also need to change the uid for all the objects that the profile owns in either the root directory or the QOpenSrv directory.) The QSYCHGID program automatically changes the uid in both the user profile and all the owned objects. For information about how to use this program, see the *iSeries System API Reference* book.

Chapter 13. Secure APPC communications

When your system participates in a network with other systems, a new set of doors and windows to your system becomes available. As security administrator, you should be aware of the options that you can use to control the entrances to your system in an APPC environment.

Advanced program-to-program communications (APPC) is a way that computers, including personal computers, communicate with each other. Display station passthrough, distributed data management, and iSeries Access for Windows can all use APPC communications.

The topics that follow provide some basic information about how APPC communications works and how you can set up appropriate security. These topics concentrate primarily on the security-relevant elements of an APPC configuration. To adapt this example to your situation, you will need to work with the people who manage your communications network and perhaps your application providers. Use this information as a foundation to help you understand the security issues and the options that are available for APPC.

Security is never "free". Some suggestions for making network security easier may make network administration more difficult. For example, this information does not emphasize APPN (Advanced Peer-to-Peer Networking®), because security is easier to understand and manage without APPN. However, without APPN, the network administrator must manually create configuration information that APPN creates automatically.

PCs Use Communications, too

Many methods for connecting PCs to your iSeries servers depend on communications, such as APPC or TCP/IP. When you read the topics the following, be sure to consider the security issues for connecting both to other systems and to PCs. When you plan your network protection, make sure that you do not adversely affect the PCs that are attached to your system.

APPC Terminology

APPC provides the ability for a user on one system to perform work on another system. The system from which the request starts is called any of the following:

- **Source system**
- **Local system**
- **Client**

The system that receives the request is called any of the following:

- **Target system**
- **Remote system**
- **Server**

© Copyright IBM Corp. 1996, 2002

Basic elements of APPC communications

From the perspective of a security administrator, the following must happen before a user on one system (SYSTEMA) can perform meaningful work on another system (SYSTEMB):

- The source system (SYSTEMA) must provide a path to the target system (SYSTEMB). This path is called an **APPC session**.
- The target system must identify the user and associate the user with a user profile. The target system must support the encryption algorithm of the source system (see "Password levels" on page 20for more information).
- The target system must start a job for the user with an appropriate environment (work management values).

The topics that follow discuss these elements and how they relate to security. The security administrator on the target system has primary responsibility for ensuring that APPC users do not violate security. However, when the security administrators on both systems work together, the job of managing APPC security is much easier.

Example: A basic APPC session

In an APPC environment, when a user or application on one system requests access to another system, the two systems set up a session. To establish the session, the systems must link two matching APPC device descriptions. The remote location name (RMTLOCNAME) parameter in the SYSTEMA device description must match the local location name (LCLLOCNAME) parameter in the SYSTEMB device description and vice versa.

For the two systems to establish an APPC session, the location passwords in the APPC device descriptions on SYSTEMA and SYSTEMB must be identical. Both must specify *NONE, or both must specify the same value.

If the passwords are a value other than *NONE, they are stored and transmitted in encrypted format. If the passwords match, the systems establish a session. If the passwords do not match, the user's request is rejected. When systems specify location passwords to establish a session, this is called a **secure bind**.

Note: Not all computer systems provide support for the secure bind function.

Restrict APPC sessions

As security administrator on a source system, you can use object authority to control who can attempt to access other systems. Set the public authority for APPC device descriptions to *EXCLUDE and give *CHANGE authority to specific users. Use the QLMTSECOFR system value to prevent users with *ALLOBJ special authority from using APPC communications.

As security administrator on a target system, you can also use authority to APPC devices to prevent users from starting an APPC session on your system. However, you need to understand what user ID will be attempting to access the APPC device description. "APPC user access to the target system" on page 111 describes how iSeries servers associates a user ID with a request for an APPC session.

Note: You can use the Print Publicly Authorized Objects (PRTPUBAUT *DEVD) command and the Print Private Authorities (PRTPVTAUT *DEVD) command to find out who has authority to device descriptions on your system.

When your system uses APPN, it automatically creates a new APPC device when no existing device is available for the route that the system has chosen. One method for restricting access to APPC devices on a system that is using APPN is to create an authorization list. The authorization list contains the list of users who should be authorized to APPC devices. You then use the Change Command Default (CHGCMDDFT) command to change the CRTDEVAPPC command. For the authority (AUT) parameter on the CRTDEVAPPC command, set the default value to the authorization list that you created.

Note: If your system has a language other than English, you need to change the command default in the QSYSxxxx library for each national language that is on your system.

You use the location password (LOCPWD) parameter in the APPC device description to validate the identity of another system that is requesting a session on your system (on behalf of a user or an application). The location password can help you detect an imposter system.

When you use location passwords, you must coordinate with security administrators for other systems in the network. You must also control who can create or change APPC device descriptions and configuration lists. The system requires *IOSYSCFG special authority to use the commands that work with APPC devices and configuration lists.

Note: When you use APPN, the location passwords are stored in the QAPPNRMT configuration list rather than in device descriptions.

APPC user access to the target system

When the systems establish the APPC session, they create a path for the requesting user to get to the door of the target system. Several other elements determine what the user must do to gain entrance to the other system.

The topics that follow describe the elements that determine how an APPC user gains entrance to the target system.

System methods for sending information about a user

APPC architecture provides three methods for sending security information about a user from the source system to the target system. These methods are referred to as the **architected security values**. Table 18 shows these methods:

Note: The *APPC Programming* book provides more information about the architected security values.

Table 18. Security values in the APPC architecture

Architected security value	User ID sent to target system	Password sent to target system
None	No	No
Same	Yes[1]	See note 2.
Program	Yes	Yes[3]

Table 18. Security values in the APPC architecture (continued)

Architected security value	User ID sent to target system	Password sent to target system
Notes: 1. The source system sends the user ID if the target system specifies SECURELOC(*YES) or SECURELOC(*VFYENCPWD). 2. The user does not enter a password on the request because the password is already verified by the source system. For SECURELOC(*YES) and SECURELOC(*NO), the source system does not send the password. For SECURELOC(*VFYENCPWD), the source system retrieves the stored, encrypted password and sends it (in encrypted form). 3. The system sends the password in encrypted form if both the source and target systems support password encryption. Otherwise, the password is not encrypted.		

The application that the user requests determines the architected security value. For example, SNADS always uses SECURITY(NONE). DDM uses SECURITY(SAME). With display station passthrough, the user specifies the security value by using parameters on the STRPASTHR command.

In all cases, the target system chooses whether to accept a request with the security value that is specified on the source system. In some situations, the target system may reject the request completely. In other situations, the target system may force a different security value. For example, when a user specifies both a user ID and a password on the STRPASTHR command, the request uses SECURITY(PGM). However, if the QRMTSIGN system value is *FRCSIGNON on the target system, the user still sees a Sign On display. With the *FRCSIGNON setting, the systems always use SECURITY(NONE), which is the equivalent of the user entering no user ID and password on the STRPASTHR command.

Notes:

1. The source and target systems negotiate the security value before data is sent. In the situation where the target system specifies SECURELOC(*NO) and the request is SECURITY(SAME), for example, the target system tells the source system to use SECURITY(NONE). The source system does not send the user ID.
2. The target system rejects a session request when the user's password on the target system has expired. This applies only to connection requests that send a password, including the following:
   - Session requests of type SECURITY(PROGRAM).
   - Session requests of type SECURITY(SAME) when the SECURELOC value is *VFYENCPWD.

Options for dividing network security responsibility

When your system participates in a network, you must decide whether to trust the other systems to validate the identity of a user who is trying to enter your system. Will you trust SYSTEMA to ensure that USERA is really USERA (or QSECOFR is really QSECOFR)? Or will you require a user to provide a user ID and password again?

The secure location (SECURELOC) parameter on the APPC device description on the target system specifies whether the source system is a secure (trusted) location.

When both systems are running a release that supports *VFYENCPWD, SECURELOC(*VFYENCPWD) provides additional protection when applications use SECURITY(SAME). Although the requester does not enter a password on the request, the source system retrieves the user's password and sends it with the request. For the request to be successful, the user must have the same user ID and password on both systems.

When the target system specifies SECURELOC(*VFYENCPWD) and the source system does not support this value, the target system handles the request as SECURITY(NONE).

Table 19 shows how the architected security value and the SECURELOC value work together:

Table 19. How the APPC security value and the SECURELOC value work together

Source system	**Target system**	
Architected security value	**SECURELOC value**	**User profile for job**
None	Any	Default user[1]
Same	*NO	Default user[1]
	*YES	Same user profile name as requester from source system
	*VFYENCPWD	Same user profile name as requester from source system. The user must have the same password on both systems.
Program	Any	The user profiles that is specified on the request from the source system.

Notes:

1. The default user is determined by the communications entry in the subsystem description. "Target system assignment of user profiles for jobs" describes this.

Target system assignment of user profiles for jobs

When a user requests an APPC job on another system, the request has a mode name associated with it. The mode name may come from the user's request, or it may be a default value from the network attributes of the source system.

The target system uses the mode name and the APPC device name to determine how the job will run. The target system searches the active subsystems for a communications entry that is the best match for the APPC device name and the mode name.

The communications entry specifies what user profile the system will use for SECURITY(NONE) requests. Following is an example of a communications entry in a subsystem description:

```
                        Display Communications Entries

Subsystem description:   QCMN           Status:   ACTIVE

                          Job                            Default         Max
Device        Mode        Description    Library         User          Active
*ALL          *ANY        *USRPRF                        *SYS          *NOMAX
*ALL          QPCSUPP     *USRPRF                        *NONE         *NOMAX
```

Table 20 shows the possible values for the default user parameter in a communications entry:

Table 20. Possible values for the default user parameter

Value	Result
***NONE**	No default user is available. If the source system does not supply a user ID on the request, the job will not run.
***SYS**	Only IBM-supplied programs (system jobs) will run. No user applications will run.
user-name	If the source system does not send a user ID, the job runs under this user profile.

You can use the Print Subsystem Description (PRTSBSDAUT) command to print a list of all subsystems that have communications entries with a default user profile.

Display station passthrough options

Display station passthrough is an example of an application that uses APPC communications. You can use display station passthrough to sign on to another system that is connected to your system through a network.

Table 21 shows examples of passthrough requests (STRPASTHR command) and how the target system handles them. For display station passthrough, the system uses the basic elements of APPC communications and the remote sign-on (QRMTSIGN) system value.

Note: Display Station Passthrough requests are no longer routed through the QCMN or QBASE subsystems. Beginning with V4R1, they are routed through the QSYSWRK subsystem. Prior to V4R1 you could assume that by not having QCMD or QBASE subsystems started, Display Station Passthrough would not work. This is no longer true. You can force Display Station Passthrough to go through QCMN (or QBASE if it is active) by changing the QPASTHRSVR system value to 0.

Table 21. Sample pass-through sign-on requests

Values on STRPASTHR command		Target system		
User ID	**Password**	**SECURELOC value**	**QRMTSIGN value**	**result**
*NONE	*NONE	Any	Any	The user must sign on the target system.
A user profile name	Not entered	Any	Any	The request fails.

Table 21. Sample pass-through sign-on requests (continued)

Values on STRPASTHR command		Target system		
User ID	**Password**	**SECURELOC value**	**QRMTSIGN value**	**result**
*CURRENT	Not entered	*NO	Any	The request fails
		*YES	*SAMEPRF	An interactive job starts with the same user profile name as the user profile on the source system. No password is passed to the remote system. The user profile name must exist on the target system.
			*VERIFY	
			*FRCSIGNON	The user must sign on the target system.
		*VFYENCPWD	*SAMEPRF	An interactive job starts with the same user profile name as the user profile on the source system. The source system retrieves the user's password and sends it to the remote system. The user profile name must exist on the target system.
			*VERIFY	
			*FRCSIGNON	The user must sign on the target system.
*CURRENT (or the name of the current user profile for the job)	Entered	Any	*SAMEPRF	An interactive job starts with the same user profile name as the user profile on the source system. The password *is* sent to the remote system. The user profile name must exist on the target system.
			*VERIFY	
			*FRCSIGNON	The user must sign on the target system.
A user profile name (a name different from the current user profile for the job)	Entered	Any	*SAMEPRF	The request fails.
			*VERIFY	An interactive job starts with the same user profile name as the user profile on the source system. The password *is* sent to the remote system. The user profile name must exist on the target system.
			*FRCSIGNON	An interactive job starts with the specified user profile name. The password is sent to the target system. The user profile name must exist on the target system.

Avoid unexpected device assignments

When a failure occurs on an active device, the system attempts to recover. In some circumstances, when the connection is broken, another user can unintentionally reestablish the session that had the failure. For example, assume that USERA powered off a workstation without signing off. USERB could power on the workstation and restart USERA's session without signing on.

To prevent this possibility, set the Device I/O Error Action (QDEVRCYACN) system value to *DSCMSG. When a device fails, the system will end the user's job.

Control remote commands and batch jobs

Several options are available to help you control what remote commands and jobs can run on your system, including the following:

- If your system uses DDM, you can restrict access to DDM files to prevent users from using the Submit Remote Command (SBMRMTCMD) command from another system. To use the SBMRMTCMD, the user must be able to open a DDM file. You also need to restrict the ability to create DDM files.
- You can specify an exit program for the DDM request access (DDMACC) system value. In the exit program, you can evaluate all DDM requests before allowing them.
- You can use the network job action (JOBACN) network attribute to prevent network jobs from being submitted or to prevent them from running automatically.
- You can specify explicitly which program requests can run in a communications environment by removing the PGMEVOKE routing entry from subsystem descriptions. The PGMEVOKE routing entry allows the requester to specify the program that runs. When you remove this routing entry from subsystem descriptions, such as the QCMN subsystem description, you must add routing entries for the communications requests that need to run successfully.

  "Architected TPN requests" on page 91 lists the program names for the communications requests by IBM-supplied applications. For each request that you want to allow, you can add a routing entry with the compare value and the program name both equal to the program name.

  When you use this method, you need to understand the work management environment on your system and the types of communications requests that occur on your system. If possible, you should test all types of communications requests to ensure that they work properly after you change the routing entries. When a communications request does not find an available routing entry, you receive a CPF1269 message. Another alternative (less error-prone but perhaps slightly less effective) is to set the public authority to *EXCLUDE for the transaction programs that you do not want to run on your system.

  Note: The *Work Management* book provides more information about routing entries and how the system handles program-start requests.

Evaluate your APPC configuration

You can use the Print Communications Security (PRTCMNSEC) command or menu options to print the security-relevant values in your APPC configuration. The topics that follow describe the information on the reports.

Relevant parameters for APPC devices

Figure 9 shows an example of the Communications Information Report for device descriptions. Figure 10 shows an example of the report for configuration lists. Following the reports are explanations of fields on the reports.

```
                          Communications Information (Full Report)
                                                                                SYSTEM4
Object type  . . . . . . . . . :   *DEVD

                                                                                    Pre           SNUF
Object         Object        Device         Secure      Location     APPN       Single      Establish     Program
Name           Type          Category       Location    Password     Capable    Session      Session       Start

CDMDEV1        *DEVD         *APPC           *NO          *NO         *NO         *YES         *NO
CDMDEV2        *DEVD         *APPC           *NO          *NO         *NO         *YES         *NO
```

Figure 9. APPC Device Descriptions-Sample Report

```
                  Display Configuration List                          Page     1
                             SYSTEM4  12/17/95  07:24:36
Configuration list . . . . . . . . . :   QAPPNRMT
Configuration list type  . . . . . . :   *APPNRMT
Text . . . . . . . . . . . . . . . . :
----------------APPN Remote Locations------------------
          Remote              Remote    Control
Remote    Network   Local     Control   Point     Secure
Location  ID        Location  Point     Net ID     Loc
SYSTEM36  APPN      SYSTEM4   SYSTEM36  APPN       *NO
SYSTEM32  APPN      SYSTEM4   SYSTEM32  APPN       *NO
SYSTEMU   APPN      SYSTEM4   SYSTEM33  APPN       *YES
SYSTEMJ   APPN      SYSTEM4   SYSTEMJ   APPN       *NO
SYSTEMR2  APPN      SYSTEM4   SYSTEM1   APPN       *NO
--------------------------APPN Remote Locations---------------------------
          Remote                                          Local      Pre-
Remote    Network   Local     Single     Number of      Control  established
Location  ID        Location  Session  Conversations     Point     Session
SYSTEM36  APPN      SYSTEM4    *NO             10        *NO        *NO
SYSTEM32  APPN      SYSTEM4    *NO             10        *NO        *NO
```

Figure 10. Configuration List Report-Example

Secure location field

The secure location (SECURELOC) field specifies whether the local system trusts the remote system to do password verification on behalf of the local system. The SECURELOC field applies only to applications that use the SECURITY(SAME) value, such as DDM and applications that use the CPI-Communications API.

SECURELOC(*YES) makes the local system vulnerable to possible weaknesses in the remote system. Any user that exists on both systems can call programs on the local system. This is particularly dangerous because the QSECOFR (security officer) user profile exists on all iSeries systems and has *ALLOBJ special authority. If a system in the network does not do a good job of protecting the QSECOFR password, other systems that treat that system as a secure location are at risk.

When you use SECURELOC(*VFYENCPWD), your system is less vulnerable to other systems that do not adequately protect passwords. A user who requests an application that uses SECURITY(SAME) must have the same user ID and password on both systems. SECURELOC(*VFYENCPWD) requires password administration policies across your network so that users have the same password on all systems.

Note: SECURELOC(*VFYENCPWD) is supported only between systems that are running V3R2, V3R7, or V4R1. If the target system specifies SECURELOC(*VFYENCPWD) and the source system does not support this function, the request is treated as SECURITY(NONE).

If a system specifies SECURELOC(*NO), applications that use SECURITY(SAME) will need a default user to run programs. The default user depends on both the device description and the mode that are associated with the request. (See "Target system assignment of user profiles for jobs" on page 113.)

Location password field

The location password field determines whether the two systems will exchange passwords to verify that the requesting system is not an imposter system. "Example: A basic APPC session" on page 110 provides more information about location passwords.

APPN Capable field

The APPN-capable (APPN) field specifies whether the remote system can support advanced networking functions or is limited to single-hop connections. APPN(*YES) means the following:

- If the remote system is a network node, the remote system may be capable of connecting the local system to other systems. This is called **intermediate node routing**. It means that users on your system may be able to use the remote system as a route to a larger network.
- If the local system is a network node, the remote system can use the local system to connect to other systems. Users on the remote system may be able to use your system as a route to a larger network.

  Note: You can use the DSPNETA command to determine whether a system is a network node or an end node.

Single session field

The single session (SNGSSN) field specifies whether the remote system can run more than one session at a time by using the same APPC device description. SNGSSN(*NO) is commonly used because it eliminates the need to create multiple device descriptions for a remote system. For example, a PC user often wants more than one 5250-emulation session and sessions for file-server and print-server functions. With SNGSSN(*NO), you can provide this function with one device description for the PC on the iSeries system.

SNGSSN(*NO) means that you must rely on the security-conscious operating procedures of PC users and other APPC users. Your system is vulnerable to someone on the remote system who starts an unauthorized session that uses the same device description as an existing session. (This practice is sometimes referred to as **piggy-backing**.)

Pre-establish session field

The pre-establish (PREESTSSN) session field for a single-session device controls whether the local system starts a session with the remote system when the remote system first contacts the local system. PREESTSSN(*NO) means that the local system waits to start a session until an application requests a session with the system. PREESTSSN(*YES) is useful for minimizing how long it takes for an application program to complete the connection.

PREESTSSN(*YES) prevents the system from disconnecting a switched (dial-up) line that is no longer being used. The application or the user must explicitly vary

off the line. PREESTSSN(*YES) may lengthen the time that the local system is vulnerable to piggy-backing on the session.

SNUF Program start field

The SNUF program start field specifies whether the remote system is allowed to start programs on the local system. *YES means that the object authority scheme on the local system must be adequate to protect objects when users on the remote system start jobs and run programs on the local system.

Parameters for APPC controllers

Figure 11 shows an example of the Communications Information Report for controller descriptions. Following the report, you will find explanations of fields on the report.

```
                         Communications Information (Full Report)
                                                                          SYSTEM4
Object type  . . . . . . . . . . :   *CTLD

Object    Object    Controller    Auto     Switched      Call       APPN       CP      Disconnect   Delete    Device
Name      Type      Category      Create   Controller    Direction  Capable    Sessions    Timer      Seconds   Name
CTL01     *CTLD     *APPC         *YES       *YES        *DIAL      *YES       *YES          0        1440    AARON
CTL02     *CTLD     *APPC         *YES       *YES        *DIAL      *YES       *YES          0        1440    BASIC
CTL03     *CTLD     *APPC         *YES       *YES        *DIAL      *YES       *YES          0        1440    *NONE
```

Figure 11. APPC Controller Descriptions-Sample Report

Auto-create field

On a line description, the auto-create (AUTOCRTCTL) field specifies whether the local system automatically creates a controller description when an incoming request cannot find a matching controller description. On a controller description, the auto-create (AUTOCRTDEV) field specifies whether the local system automatically creates a device description when an incoming request cannot find a matching device description.

For controllers that are APPN-capable, the auto-create field has no effect. The system automatically creates device descriptions when necessary, regardless of how you have set the auto-create field.

When you specify *YES for a line description, anyone with access to the line can connect to your system. This includes sites that are connected by bridges and routers.

Control point sessions field

For APPN-capable controllers, the control point sessions (CPSSN) field controls whether the system establishes an APPC connection with the remote system automatically. The system uses the CP session to exchange network information and status with the remote system. The exchange of up-to-date information between APPN network nodes is particularly important so that your network functions smoothly.

When you specify *YES, an idle switched line does not disconnect automatically. This makes your system more vulnerable to a piggy-back session.

Disconnect timer field

For an APPC controller, the disconnect timer field specifies how long a controller must be unused (no active sessions) before the system disconnects the line to the remote system. This field has two values. The first value specifies how long the controller will stay active from the time it is initially contacted. The second value

determines how long the system waits after the last session has ended on the controller before the system drops the line.

The system uses the disconnect timer only when the switched disconnect (SWTDSC) field is *YES.

If you make these values large, your system is more vulnerable to piggy-back sessions.

Parameters for line descriptions

Figure 12 shows an example of the Communications Information Report for line descriptions. Following the report, you will find explanations of fields on the report.

```
                    Communications Information (Full Report)

Object type  . . . . . . . . . :   *LIND
Auto
Object         Object         Line           Auto      Delete       Auto      Auto
Name           Type           Category       Create    Seconds      Answer    Dial
LINE01          *LIND          *SDLC           *NO            0       *NO       *NO
LINE02          *LIND          *SDLC           *NO            0       *YES      *NO
LINE03          *LIND          *SDLC           *NO            0       *NO       *NO
LINE04          *LIND          *SDLC           *NO            0       *YES      *NO
```

Figure 12. APPC Line Descriptions-Sample Report

Auto answer field

The auto answer (AUTOANS) field specifies whether the switched line will accept incoming calls without operator intervention.

When you specify *YES, your system is less secure because it can be accessed more easily. To minimize the security exposure when you specify *YES, you should vary off the line when you do not need it.

Auto dial field

The auto dial (AUTODIAL) field specifies whether the switched line can make outgoing calls without operator intervention. When you specify *YES, you allow local users who do not have physical access to communications lines and modems to connect to other systems.

Chapter 14. Secure TCP/IP communications

TCP/IP (Transmission Control Protocol/Internet Protocol) is a common way that computers of all types communicate with each other. TCP/IP applications are well-known and widely used throughout the "information highway".

This chapter provides tips for the following:

- Preventing TCP/IP applications from running on your system.
- Protecting system resources when you allow TCP/IP applications to run on your system.

The iSeries Information Center—>Networking—>TCP/IP website is a complete source for information about all of the TCP/IP applications. *SecureWay®: iSeries and the Internet* (iSeries Information Center—>Security—>SecureWay describes security considerations when you connect your iSeries server either to the Internet (a very large TCP/IP network) or to an intranet. See "Prerequisite and related information" on page xii for information on accessing the iSeries Information Center.

Keep in mind that iSeries servers support many possible TCP/IP applications. When you decide to allow one TCP/IP application on your system, you may also be enabling other TCP/IP applications. As security administrator, you need to be aware of the range of TCP/IP applications and the security implications of these applications.

Prevent TCP/IP processing

TCP/IP server jobs run in the QSYSWRK subsystem. You use the Start TCP/IP (STRTCP) command to start TCP/IP on your system. If you do not want any TCP/IP processing or applications to run, do not use the STRTCP command. Your system ships with the public authority for the STRTCP command set to *EXCLUDE.

If you suspect that someone with access to the command is starting TCP/IP (during off-hours, for example), you can set up object auditing on the STRTCP command. The system will write an audit journal entry whenever a user runs the command.

TCP/IP security components

You can take advantage of several TCP/IP security components that enhance your network security and add flexibility. Though some of these technologies are also found in firewall products, these TCP/IP security components for OS/400 are not intended to be used as a firewall. However, you may be able to use some of these features, in some instances to eliminate the need for a separate firewall product. You also may be able to use these TCP/IP features to provide additional security in environments where you already use a firewall.

The following components can be utilized to enhance TCP/IP Security:

- Packet Rules
- HTTP Proxy Server
- VPN (virtual private networking)
- SSL (secure sockets layer)

© Copyright IBM Corp. 1996, 2002

Use packet rules to secure TCP/IP traffic

Packet rules, which is the combination of IP filtering and network address translation (NAT) acts like a firewall to protect your internal network from intruders. IP filtering lets you control what IP traffic to allow into and out of your network. Basically, it protects your network by filtering packets according to rules that you define. NAT, on the other hand, allows you to hide your unregistered private IP addresses behind a set of registered IP addresses. This helps to protect your internal network from outside networks. NAT also helps to alleviate the IP address depletion problem, since many private addresses can be represented by a small set of registered addresses. See the iSeries Information Center for more details.

HTTP proxy server

The HTTP proxy server comes with the IBM HTTP Server for iSeries server. The HTTP Server is part of OS/400. The proxy server receives HTTP requests from Web browsers and resends them to Web servers. Web servers that receive the requests are only aware of the IP address of the proxy server and cannot determine the names or addresses of the PCs that originated the requests. The proxy server can handle URL requests for HTTP, FTP, Gopher and WAIS.

The proxy server caches returned Web pages from requests made by all proxy server users. Consequently, when users request a page, the proxy server checks whether the page is in the cache. If it is, the proxy server returns the cached page. By using cached pages, the proxy server is able to server Web pages more quickly, which eliminates potentially time-consuming requests to the Web server.

The proxy server can also log all URL requests for tracking purposes. You can then review the logs to monitor use and misuse of network resources.

You can use the HTTP proxy support in the IBM HTTP Server to consolidate Web access. Addresses of PC clients are hidden from the Web servers they access; only the IP address of the proxy server is known. Web page caching can also reduce communication bandwidth requirements and firewall workload. See the IBM HTTP Server for iSeries Homepage for more information: http://www-1.ibm.com/servers/eserver/iseries/software/http/index.html

Virtual Private Networking (VPN)

A virtual private network (VPN) allows your company to securely extend its private intranet over the existing framework of a public network, such as the Internet. With VPN, your company can control network traffic while providing important security features such as authentication and data privacy.

OS/400 VPN is an optionally-installable component of iSeries Navigator, the graphical user interface (GUI) for OS/400. It allows you to create a secure end-to-end path between any combination of host and gateway. OS/400 VPN uses authentication methods, encryption algorithms, and other precautions to ensure that data sent between the two endpoints of its connection remains secure.

VPN runs on the network layer of the TCP/IP layered communications stack model. Specifically, VPN uses the IP Security Architecture (IPSec) open framework. IPSec provides base security functions for the Internet, as well as furnishes flexible building blocks from which you can create robust, secure virtual private networks.

VPN also supports Layer 2 Tunnel Protocol (L2TP) VPN solutions. L2TP connections, which are also called virtual lines, provide cost-effective access for remote users by allowing a corporate network server to manage the IP addresses assigned to its remote users. Further, L2TP connections provide secure access to your system or network when you protect them with IPSec.

It is important that you understand the impact a VPN will have on your entire network. Proper planning and implementation are essential to your success. You should review the VPN topic in the iSeries Information Center to ensure that you know how VPNs work and how you might use them. For more information, see the iSeries Information Center—>Security—>Virtual Private Networking. See "Prerequisite and related information" on page xii for information on accessing the iSeries Information Center.

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) has become an industry standard for enabling applications for secure communication sessions over an unprotected network, such as the Internet. The SSL protocol establishes a secure connections between clients and server applications which provide authentication of one or both endpoints of the communication session. SSL also provides privacy and integrity of the data that client and server applications exchange. For more information, see the iSeries Information Center–>Security—>Secure Sockets Layer (SSL). See "Prerequisite and related information" on page xii for information on accessing the iSeries Information Center.

Secure your TCP/IP environment

This topic provides general suggestions for steps that you can take to reduce the security exposures in the TCP/IP environment on your system. These tips apply to your entire TCP/IP environment rather than to the specific applications that are discussed in the topics that follow.

- When you write an application for a TCP/IP port, make sure that the application is properly secure. You should assume that an outsider might try to access that application through that port. A knowledgeable outsider may attempt to TELNET to that application.
- Monitor the use of TCP/IP ports on your system. A user application that is associated with a TCP/IP port can provide "back-door" entry to your system without a user ID or a password. Someone with sufficient authority on your system can associate an application with a TCP or UDP port.
- As a security administrator, you should be aware of a technique called *IP spoofing* that is used by hackers. Every system in a TCP/IP network has an IP address. Someone who uses IP spoofing sets up a system (usually a PC) to pretend to be an existing IP address or a trusted IP address. Thus, the imposter can establish a connection with your system by pretending to be a system that you normally connect with.

  If you run TCP/IP on your system and your system participates in a network that is not physically protected (all nonswitched lines and predefined links), you are vulnerable to IP spoofing. To protect your system from damage by a "spoofer", start with the suggestions in this chapter, such as sign-on protection and object security. You should also ensure that your system has reasonable auxiliary storage limits set. This prevents a spoofer from flooding your system with mail or spooled files to the point that your system becomes inoperable.

In addition, you should regularly monitor TCP/IP activity on your system. If you detect IP spoofing, you can try to discover the weak points in your TCP/IP setup and to make adjustments.

- For your intranet (network of systems that do not need to connect directly to the outside), use IP addresses that are reusable. Reusable addresses are intended for use within a private network. The Internet backbone does not route packets that have a reusable IP address. Therefore, reusable addresses provide an added layer of protection inside your firewall.

  The iSeries Information Center—>Networking—>TCP/IP website provides more information about how IP addresses are assigned and about the ranges of IP addresses, as well as security information about TCP/IP.
- If you are considering connecting your system to the Internet or an intranet, review the security information at*SecureWay: iSeries and the Internet* (iSeries Information Center—>Security—>SecureWay). See "Prerequisite and related information" on page xii for information on accessing the iSeries Information Center.

Control which TCP/IP servers start automatically

As security administrator, you need to control which TCP/IP applications start automatically when you start TCP/IP. Two commands are available for starting TCP/IP. For each command, the system uses a different method to determine which applications (servers) to start.

Table 22 shows the two commands and security recommendations for them. Table 23 on page 125 shows the default autostart values for the servers. To change the autostart value for a server, use the CHGxxxA (Change xxx Attributes) command for the server. For example, the command for TELNET is CHGTELNA.

Table 22. How TCP/IP commands determine which servers to start

Command	What servers start	Security recommendations
Start TCP/IP (STRTCP)	The system starts every server that specifies AUTOSTART(*YES). Table 23 on page 125 shows the shipped value for each TCP/IP server.	• Assign *IOSYSCFG special authority carefully to control who can change the autostart settings. • Carefully control who has authority to use the STRTCP command. The default public authority for the command is *EXCLUDE. • Set up object auditing for the Change *server-name* Attributes commands (such as CHGTELNA) to monitor users who attempt to change the AUTOSTART value for a server.
Start TCP/IP Server (STRTCPSVR)	You use a parameter to specify which servers to start. The default when this command ships is to start all servers.	• Use the Change Command Default (CHGCMDDFT) command to set up the STRTCPSVR command to start only a specific server. This does not prevent users from starting other servers. However, by changing the command default, you make it less likely that users will start all servers by accident. For example, use the following command to set the default to start only the TELNET server:`CHGCMDDFT CMD(STRTCPSVR) NEWDFT('SERVER(*TELNET)')` **Note:** When you change the default value, you can specify only a single server. Choose either a server that you use regularly or a server that is least likely to cause security exposures (such as TFTP). • Carefully control who has authority to use the STRTCPSVR command. The default public authority for the command is *EXCLUDE.

The following table contains autostart values for TCP/IP servers. For more information on each of these servers, see the iSeries Information Center (**Networking—>TCP/IP**). See "Prerequisite and related information" on page xii for details on accessing the iSeries Information Center.

Table 23. Autostart values for TCP/IP servers

Server	Default value	Your value
TELNET	AUTOSTART(*YES)	
FTP (file transfer protocol)	AUTOSTART(*YES)	
BOOTP (Bootstrap Protocol)	AUTOSTART(*NO)	
TFTP (trivial file transfer protocol)	AUTOSTART(*NO)	
REXEC (Remote EXECution server)	AUTOSTART(*NO)	
RouteD (Route Daemon)	AUTOSTART(*NO)	
SMTP (simple mail transfer protocol)	AUTOSTART(*YES)	
POP (Post Office Protocol)	AUTOSTART(*NO)	
HTTP (Hypertext Transfer Protocol) [1]	AUTOSTART(*NO)	
ICS (Internet Connection Server)[1]	AUTOSTART(*NO)	
LPD (line printer daemon)	AUTOSTART(*YES)	
SNMP (Simple Network Management Protocol (SNMP))	AUTOSTART(*YES)	
DNS (domain name system)	AUTOSTART(*NO)	
DDM	AUTOSTART(*NO)	
DHCP (dynamic host configuration protocol)	AUTOSTART(*NO)	
NSMI	AUTOSTART(*NO)	
INETD	AUTOSTART(*NO)	

Notes:

1. With the IBM HTTP Server for iSeries server, you use the CHGHTTPA command to set the AUTOSTART value.

Security considerations for using SLIP

iSeries server TCP/IP support includes Serial Interface Line Protocol (SLIP). SLIP provides low-cost point-to-point connectivity. A SLIP user can connect to a LAN or a WAN by establishing a point-to-point connection with a system that is part of the LAN or WAN.

SLIP runs on an asynchronous connection. You can use SLIP for dial-up connection to and from iSeries servers. For example, you might use SLIP to dial in from your PC to an iSeries system. After the connection is established, you can use the TELNET application on your PC to connect to the iSeries TELNET server. Or, you can use the FTP application to transfer files between the two systems.

No SLIP configuration exists on your system when it ships. Therefore, if you do not want SLIP (and dial-up TCP/IP) to run on your system, do not configure any configuration profiles for SLIP. You use the Work with TCP/IP Point-to-Point (WRKTCPPTP) command to create SLIP configurations. You must have *IOSYSCFG special authority to use the WRKTCPPTP command.

If you want SLIP to run on your system, you create one or more SLIP (point-to-point) configuration profiles. You can create configuration profiles with the following operating modes:
- Dial in (*ANS)
- Dial out (*DIAL)

The topics that follow discuss how you can set up security for SLIP configuration profiles.

Note: A **user profile** is an iSeries server object that allows sign-on. Every iSeries server job must have a user profile to run. A **configuration profile** stores information that is used to establish a SLIP connection with an iSeries system. When you start a SLIP connection to iSeries servers, you are simply establishing a link. You have not yet signed on and started an iSeries server job. Therefore, you do not necessarily need a user profile to start a SLIP connection to iSeries servers. However, as you will see in the discussions that follow, the SLIP configuration profile may require a user profile to determine whether to allow the connection.

Control dial-in SLIP connections

Before someone can establish a dial-in connection to your system with SLIP, you must start a SLIP *ANS configuration profile. To create or change a SLIP configuration profile, you use the Work with TCP/IP Point-to-Point (WRKTCPPTP) command. To start a configuration profile, you use either the Start TCP/IP Point-to-Point (STRTCPPTP) command or an option from the WRKTCPPTP display. When your system ships, the public authority for the STRTCPPTP and ENDTCPPTP commands are *EXCLUDE. The options to add, change, and delete SLIP configuration profiles are available only if you have *IOSYSCFG special authority. As security administrator, you can use both command authority and special authority determine who can set up your system to allow dial-in connections.

Secure a dial-in SLIP connection

If you want to validate systems that dial in to your system, then you want the requesting system to send a user ID and a password. Your system can then verify the user ID and password. If the user ID and password are not valid, your system can reject the session request.

To set up dial-in validation, do the following:

__ Step 1. Create a user profile that the requesting system can use to establish the connection. The user ID and password that the requester sends must match this user profile name and password.

Note: For the system to perform password validation, the QSECURITY system value must be set to 20 or higher.

As additional protection, you probably want to create user profiles specifically for establishing SLIP connections. The user profiles should have limited authority on the system. If you do not plan to use the profiles for any function except establishing SLIP connections, you can set the following values in the user profiles:
- An initial menu (INLMNU) of *SIGNOFF
- An initial program (INLPGM) of *NONE.
- Limit capabilities (LMTCPB) of *YES

These values prevent anyone from signing on interactively with the user profile.

__ Step 2. Create an authorization list for the system to check when a requester tries to establish a SLIP connection.

Note: You specify this authorization list in the *System access authorization list* field when you create or change the SLIP profile. (See step 4.)

__ Step 3. Use the Add Authorization Entry (ADDAUTLE) command to add the user profile that you created in step 1 to the authorization list. You can create a unique authorization list for each point-to-point configuration profile, or you can create an authorization list that several configuration profiles share.

__ Step 4. Use the WRKTCPPTP command to set up a TCP/IP point-to-point *ANS profile that has the following characteristics:

- The configuration profile must use a connection dialog script that includes the user-validation function. User validation includes accepting a user ID and password from the requester and validating them. The system ships with several sample dialog scripts that provide this function.
- The configuration profile must specify the name of the authorization list that you created in step 2. The user ID that the connection dialog script receives must be in the authorization list.

Keep in mind that the value of setting up dial-in security is affected by the security practices and capabilities of the systems that dial in. If you require a user ID and password, then the connection dialog script on the requesting system must send that user ID and password. Some systems, such as iSeries servers, provide a secure method for storing the user IDs and passwords. (“Security and dial-out sessions” on page 128 describes the method.) Other systems store the user ID and password in the script which might be accessible to anyone who knows where to find the script on the system.

Because of the differing security practices and capabilities of your communications partners, you might want to create different configuration profiles for different requesting environments. You use STRTCPPTP command to set your system up to accept a session for a specific configuration profile. You can start sessions for some configuration profiles only at certain times of the day, for example. You might use security auditing to log the activity for the associated user profiles.

Prevent dial-in users from accessing other systems

Depending on your system and network configuration, a user who starts a SLIP connection might be able to access another system in your network without signing on to your system. For example, a user could establish a SLIP connection to your system. Then the user could establish an FTP connection to another system in your network that does not allow dial-in.

You can prevent a SLIP user from accessing other systems in your network by specifying N (No) for the *Allow IP datagram forwarding* field in the configuration profile. This prevents a user from accessing your network before the user logs on to your system. However, after the user has successfully logged on to your system, the datagram forwarding value has no effect. It does not limit the user’s ability to use a TCP/IP application on your iSeries system (such as FTP or TELNET), to establish a connection with another system in your network.

Control dial-out sessions

Before someone can use SLIP to establish a dial-out connection from your system, you must start a SLIP *DIAL configuration profile. To create or change a SLIP configuration profile, you use the WRKTCPPTP command. To start a configuration profile, you use either the Start TCP/IP Point-to-Point (STRTCPPTP) command or an option from the WRKTCPPTP display. When your system ships, the public authority for the STRTCPPTP and ENDTCPPTP commands are *EXCLUDE. The options to add, change, and delete SLIP configuration profiles are available only if you have *IOSYSCFG special authority. As security administrator, you can use both command authority and special authority determine who can set up your system to allow dial-out connections.

Security and dial-out sessions

Users on your iSeries system might want to establish dial-out connections to systems that require user validation. The connection dialog script on your iSeries server must send a user ID and a password to the remote system. iSeries servers provide a secure method for storing that password. The password does not need to be stored in the connection dialog script.

Notes:

1. Even though your system stores the connection password in encrypted form, your system decrypts the password before sending it. SLIP passwords, like FTP and TELNET passwords, are sent unencrypted ("in the clear"). However, unlike with FTP and TELNET, the SLIP password is sent before the systems establish TCP/IP mode.

   Because SLIP uses a point-to-point connection in asynchronous mode, the security exposure when sending unencrypted passwords is different from the exposure with FTP and TELNET passwords. Unencrypted FTP and TELNET passwords might be sent as IP traffic on a network and are, therefore, vulnerable to electronic sniffing. The transmission of your SLIP password is as secure as the telephone connection between the two systems.
2. The default file for storing SLIP connection dialog scripts is QUSRSYS/QATOCPPSCR. The public authority for this file is *USE, which prevents public users from changing the default connection dialog scripts.

When you create a connection profile for a remote session that requires validation, do the following:

__ Step 1. Ensure that the Retain Server Security Data (QRETSVRSEC) system value is 1 (Yes). This system value determines whether you will allow passwords that can be decrypted to be stored in a protected area on your system.

__ Step 2. Use the WRKTCPPTP command to create a configuration profile that has the following characteristics:

- For the mode of the configuration profile, specify *DIAL.
- For the *Remote service access name*, specify the user ID that the remote system expects. For example, if you are connecting to another iSeries server, specify the user profile name on that iSeries server.
- For the *Remote service access password*, specify the password that the remote system expects for this user ID. On your iSeries server, this password is stored in a protected area in a form that can be decrypted. The names and passwords that you assign for configuration profiles are associated with the QTCP user profile. The

names and passwords are not accessible with any user commands or interfaces. Only registered system programs can access this password information.

Note: Keep in mind that the passwords for your connection profiles are not saved when your save the TCP/IP configuration files. To save SLIP passwords, you need to use the Save Security Data (SAVSECDTA) command to save the QTCP user profile.

- For the connection dialog script, specify a script that sends the user ID and password. The system ships with several sample dialog scripts that provide this function. When the system runs the script, the system retrieves the password, decrypts it, and sends it to the remote system.

Security considerations for point-to-point protocol

Point-to-point protocol (PPP) is available as part of TCP/IP. PPP is an industry standard for point-to-point connections that provides additional function over what is available with SLIP.

With PPP, your iSeries server can have high-speed connections directly to an Internet Service Provider or to other systems in an intranet or extranet. Remote LANs can realistically make dial-in connections to your iSeries server.

Remember that PPP, like SLIP, provides a network connection to your iSeries server. A PPP connection essentially brings the requester to your system's door. The requester still needs a user ID and password to enter your system and connect to a TCP/IP server like TELNET or FTP. Following are security considerations with this new connection capability:

Note: You configure PPP by using iSeries Navigator on an IBM iSeries Access for Windows workstation.

- PPP provides the ability to have dedicated connections (where the same user always has the same IP address). With a dedicated address, you have the potential for IP spoofing (an imposter system that pretends to be a trusted system with a known IP address). However, the enhanced authentication capabilities that PPP provides help protect against IP spoofing.
- With PPP, as with SLIP, you create connection profiles that have a user name and an associated password. However, unlike SLIP, the user does not need to have a valid user profile and password. The user name and password are not associated with a user profile. Instead, validation lists are used for PPP authentication. Additionally, PPP does not require a connection script. The authentication (exchange of user name and password) is part of the PPP architecture and happens at a lower level than with SLIP.
- With PPP, you have the option to use CHAP (challenge handshake authentication protocol). You will no longer need to worry about an eavesdropper sniffing passwords because CHAP encrypts user names and passwords.

  Your PPP connection uses CHAP only if both sides have CHAP support. During the exchange signals to set up communications between two modems, the two systems negotiate. For example, if SYSTEMA supports CHAP and SYSTEMB does not, SYSTEMA can either deny the session or agree to use an unencrypted user name and password. Agreeing to use an unencrypted user name and password is referred to as negotiating down. The decision to negotiate down is a configuration option. On your intranet, for example, where you know that all

your systems have CHAP capability, you should configure your connection profile so that it will not negotiate down. On a public connection where your system is dialing out, you might be willing to negotiate down.

The connection profile for PPP provides the ability to specify valid IP addresses. You can, for example, indicate that you expect a specific address or range of addresses for a specific user. This capability, together with the ability for encrypted passwords, provides further protection against spoofing.

As additional protection against spoofing or piggy-backing on an active session, you can configure PPP to rechallenge at designated intervals. For example, while a PPP session is active, your iSeries server might challenge the other system for a user and password. It does this every 15 minutes to ensure that `it is the same connection profile. (The end-user will not be aware of this rechallenge activity. The systems exchange names and passwords below the level that the end-user sees.)

With PPP, it is realistic to expect that remote LANs might establish a dial-in connection to your iSeries server and to your extended network. In this environment, having IP forwarding turned on is probably a requirement. IP forwarding has the potential to allow an intruder to roam through your network. However, PPP has stronger protections (such as encryption of passwords and IP address validation). This makes it less likely that an intruder can establish a network connection in the first place.

For more information about PPP, see the iSeries Information Center..

Security considerations for using Bootstrap Protocol server

Bootstrap Protocol (BOOTP) provides a dynamic method for associating workstations with servers and assigning workstation IP addresses and initial program load (IPL) sources. BOOTP and trivial file transfer protocol (TFTP) together provide support for the IBM NetVista™ thin client.

BOOTP is a TCP/IP protocol used to allow a media-less workstation (client) to request a file containing initial code from a server on the network. The BOOTP server listens on the well known BOOTP server port 67. When a client request is received, the server looks up the IP address defined for the client and returns a reply to the client with the client's IP address and the name of the load file. The client then initiates a TFTP request to the server for the load file. The mapping between the client hardware address and IP address is kept in the BOOTP table on the iSeries server.

Prevent BOOTP Access

If you do not have any thin clients attached to your network, you do not need to run the BOOTP server on your system. It can be used for other devices, but the preferred solution for those devices is to use DHCP. Do the following to prevent the BOOTP server from running:

__ Step 1. To prevent BOOTP server jobs from starting automatically when you start TCP/IP, type the following:

```
CHGBPA AUTOSTART(*NO)
```

Notes:

a. AUTOSTART(*NO) is the default value.

b. "Control which TCP/IP servers start automatically" on page 124 provides more information about controlling which TCP/IP servers start automatically.

__ Step 2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for BOOTP, do the following:

Note: Because DHCP and BOOTP use the same port number, this will also inhibit the port that is used by DHCP. Do not restrict the port if you want to use DHCP.

__ Step a. Type `GO CFGTCP` to display the Configure TCP/IP menu.

__ Step b. Select option 4 (Work with TCP/IP port restrictions).

__ Step c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).

__ Step d. For the lower port range, specify `67`.

__ Step e. For the upper port range, specify `*ONLY`.

Notes:

1) The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.
2) RFC1700 provides information about common port number assignments.

__ Step f. For the protocol, specify `*UCD`.

__ Step g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

Secure the BOOTP server

The BOOTP server does not provide direct access to your iSeries system, and thus represents a limited security exposure. Your primary concern as a security administrator is to ensure that the correct information is associated with the correct thin client. In other words, a mischief-maker could alter the BOOTP table and cause your thin clients to work incorrectly or not at all.

To administer the BOOTP server and the BOOTP table, you must have *IOSYSCFG special authority. You need to carefully control the user profiles that have *IOSYSCFG special authority on your system. The *IBM Network Station Manager for AS/400* book describes the procedures for working with the BOOTP table.

Security considerations for using DHCP server

Dynamic host configuration protocol (DHCP) provides a framework for passing configuration information to hosts on a TCP/IP network. For your client workstations, DHCP can provide a function similar to auto configuration. A DHCP-enabled program on the client workstation broadcasts a request for configuration information. If the DHCP server is running on your iSeries server, the server responds to the request by sending the information that the client workstation needs to correctly configure TCP/IP.

You can use DHCP to make it simpler for users to connect to your iSeries server for the first time. This is because the user does not need to enter TCP/IP configuration information. You can also use DHCP to reduce the number of

internal TCP/IP addresses that you need in a subnetwork. The DHCP server can temporarily allocate IP addresses to active users (from its pool of IP addresses).

For thin clients, you can use DHCP in place of BOOTP. DHCP provides more function than BOOTP, and it can support dynamic configuration of both thin clients and PCs.

Prevent DHCP access

If you do *not* want anyone to use the DHCP server on your system, do the following:

1. To prevent DHCP server jobs from starting automatically when you start TCP/IP, type the following:

   ```
   CHGDHCPA AUTOSTART(*NO)
   ```

   Notes:

   a. AUTOSTART(*NO) is the default value.

   b. "Control which TCP/IP servers start automatically" on page 124 provides more information about controlling which TCP/IP servers start automatically.

2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for DHCP, do the following:

   a. Type `GO CFGTCP` to display the Configure TCP/IP menu.

   b. Select option 4 (Work with TCP/IP port restrictions).

   c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).

   d. For the lower port range, specify `67`.

   e. For the upper port range, specify `68`.

      Notes:

      1) The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.

      2) RFC1700 provides information about common port number assignments.

   f. For the protocol, specify `*UDP`.

   g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

Secure the DHCP server

Following are security considerations when you choose to run DHCP on your iSeries system:

- Restrict the number of users who have authority to administer DHCP. Administering DHCP requires the following authority:
  - *IOSYSCFG special authority
  - *RW authority to the following files:

    ```
    /QIBM/UserData/OS400/DHCP/dhcpsd.cfg
    /QIBM/UserData/OS400/DHCP/dhcprd.cfg
    ```

- Evaluate how physically accessible your LAN is. Could an outsider easily walk into your location with a laptop and physically connect it to your LAN? If this is an exposure, DHCP provides the capability to create a list of clients (hardware

addresses) that the DHCP server will configure. When you use this feature, you remove some of the productivity benefit that DHCP provides to your network administrators. However, you prevent the system from configuring unknown workstations.

- If possible, use a pool of IP addresses that is reusable (not architected for the Internet). This helps prevent a workstation from outside your network from gaining usable configuration information from the server.
- Use the DHCP exit points if you need additional security protection. Following is an overview of the exit points and their capabilities. The *iSeries System API Reference* describes how to use these exit points.

  Port entry
  The system calls your exit program whenever it reads a data packet from port 67 (the DHCP port). Your exit program receives the full data packet. It can decide whether the system should process or discard the packet. You can use this exit point when existing DHCP screening features are not sufficient for your needs.

  Address assignment
  The system calls your exit program whenever DHCP formally assigns an address to a client.

  Address release
  The system calls your exit program whenever DHCP formally releases an address and places it back in the address pool.

Security considerations for using TFTP server

Trivial file transfer protocol (TFTP) provides basic file transfer with no user authentication. TFTP works with either Bootstrap Protocol (BOOTP) or Dynamic Host Configuration Protocol (DHCP) to provide support for the IBM NetVista thin client.

The IBM NetVista thin client connects initially to either the BOOTP server or the DHCP server. The BOOTP server or the DHCP server replies with the client's IP address and the name of the load file. The client then initiates a TFTP request to the server for the load file. When the client completes downloading of the load file, it ends the TFTP session.

Prevent TFTP access

If you do not have any thin clients attached to your network, you probably do not need to run the TFTP server on your system. Do the following to prevent the TFTP server from running:

__ Step 1. To prevent TFTP server jobs from starting automatically when you start TCP/IP, type the following:

```
CHGTFTPA AUTOSTART(*NO)
```

Notes:

a. AUTOSTART(*NO) is the default value.

b. "Control which TCP/IP servers start automatically" on page 124 provides more information about controlling which TCP/IP servers start automatically.

__ Step 2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for TFTP, do the following:

__ Step a. Type `GO CFGTCP` to display the Configure TCP/IP menu.

__ Step b. Select option 4 (Work with TCP/IP port restrictions).

__ Step c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).

__ Step d. For the lower port range, specify `69`.

__ Step e. For the upper port range, specify `*ONLY`.

Notes:

1) The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.
2) RFC1700 provides information about common port number assignments.

__ Step f. For the protocol, specify `*UCD`.

__ Step g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

Secure the TFTP server

By default, the TFTP server provides very limited access to your iSeries system. It is specifically configured to provide the initial code for thin clients. As a security administrator, you should be aware of the following characteristics of the TFTP server:

- The TFTP server does not require authentication (a user ID and password). All TFTP jobs run under the QTFTP user profile. The QTFTP user profile does not have a password. Therefore, it is not available for interactive sign-on. The QTFTP user profile does not have any special authorities, nor is it explicitly authorized to system resources. It uses public authority to access the resources that it needs for the thin clients.
- When the TFTP server arrives, it is configured to access the directory that contains thin client information. You must have *PUBLIC or QTFTP authorized to read or write to that directory. To write to the directory you must have *CREATE specified on the ″Allow file writes″ parameter of the CHGTFTPA command. To write to an existing file you must have the *REPLACE specified on the ″Allow file writes″ parameter of CHGTFTPA. *CREATE allows you to replace existing files or create new files. *REPLACE only allows you to replace existing files.

  A TFTP client cannot access any other directory unless you explicitly define the directory with the Change TFTP Attributes (CHGTFTPA) command. Therefore, if a local or remote user does attempt to start a TFTP session to your system, the user's ability to access information or cause damage is extremely limited.
- If you choose to configure your TFTP server to provide other services in addition to handling thin clients, you can define an exit program to evaluate and authorize every TFTP request. The TFTP server provides a request validation exit similar to the exit that is available for the FTP server. For more information, see the iSeries Information Center—>Networking—>TCP/IP—>TFTP. See "Prerequisite and related information" on page xii for information on accessing the iSeries Information Center.

Security considerations for using REXEC server

The Remote EXECution server (REXEC) receives and runs commands from an REXEC client. A REXEC client is typically a PC or UNIX application that supports sending REXEC commands. The support that this server provides is similar to the capability that is available when you use the RCMD (Remote Command) sub-command for the FTP server.

Prevent REXEC access

If you do not want your iSeries server to accept commands from an REXEC client, do the following to prevent the REXEC server from running:

__ Step 1. To prevent REXEC server jobs from starting automatically when you start TCP/IP, type the following:

```
CHGRXCA AUTOSTART(*NO)
```

Notes:

a. AUTOSTART(*NO) is the default value.

b. "Control which TCP/IP servers start automatically" on page 124 provides more information about controlling which TCP/IP servers start automatically.

__ Step 2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for REXEC, do the following:

__ Step a. Type `GO CFGTCP` to display the Configure TCP/IP menu.

__ Step b. Select option 4 (Work with TCP/IP port restrictions).

__ Step c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).

__ Step d. For the lower port range, specify `512`.

__ Step e. For the upper port range, specify `*ONLY`.

__ Step f. For the protocol, specify `*TCP`.

__ Step g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

Notes:

a. The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.

b. RFC1700 provides information about common port number assignments.

Secure the REXEC server

Following are considerations when you choose to run the Remote EXECution server on your system:

- An REXCD request includes a user ID, a password, and the command to run. Normal iSeries server authentication and authority checking applies:
  - The user profile and password combination must be valid.
  - The system enforces the *Limit capabilities* (LMTCPB) value for the user profile.

  - The user must be authorized to the command and to all of the resources that the command uses.
- The REXEC server provides exit points similar to the exit points that are available for the FTP server. You can use the Validation exit point to evaluate the command and decide whether to allow it. For more information, see the iSeries Information Center—>Networking—>TCP/IP—>REXEC. See "Prerequisite and related information" on page xii for information on accessing the iSeries Information Center.
- When you choose to run the REXEC server, you are running outside any menu access control that you have on your system. You must ensure that your object authority scheme is adequate to protect your resources.

Security considerations for using RouteD

The Route Daemon (RouteD) server provides support for the Routing Information Protocol (RIP) on iSeries servers. RIP is the most widely used of routing protocols. It is an Interior Gateway Protocol that assists TCP/IP in the routing of IP packets within an autonomous system.

RouteD is intended to increase the efficiency of network traffic by allowing systems within a trusted network to update each other with current route information. When you run RouteD, your system can receive updates from other participating systems about how transmissions (packets) should be routed. Therefore, if your RouteD server is accessible to a hacker, the hacker might use it to reroute your packets through a system that can sniff or modify those packets. Following are suggestions for RouteD security:

- iSeries servers use RIPv1, which does not provide any method for authenticating routers. It is intended for use within a trusted network. If your system is in a network with other systems that you do not "trust," you should not run the RouteD server. To ensure that the RouteD server does not start automatically, type the following:

  ```
  CHGRTDA AUTOSTART(*NO)
  ```

  Notes:

  1. AUTOSTART(*NO) is the default value.
  2. "Control which TCP/IP servers start automatically" on page 124 provides more information about controlling which TCP/IP servers start automatically.
- Make sure that you control who can change the RouteD configuration, which requires *IOSYSCFG special authority.
- If your system participates in more than one network (for example, an intranet and the Internet), you can configure the RouteD server to send and accept updates only with the secure network.

Security considerations for using DNS server

The Domain Name System (DNS) server provides translation of host name to IP addresses and vice versa. On iSeries servers, the DNS server is intended to provide address translation for the internal, secure network (intranet).

Prevent DNS access

If you do *not* want anyone to use the DNS server on your system, do the following:

1. To prevent DNS server jobs from starting automatically when you start TCP/IP, type the following:

```
CHGDNSA AUTOSTART(*NO)
```

Notes:

a. AUTOSTART(*NO) is the default value.

b. "Control which TCP/IP servers start automatically" on page 124 provides more information about controlling which TCP/IP servers start automatically.

2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for DNS, do the following:

   a. Type `GO CFGTCP` to display the Configure TCP/IP menu.

   b. Select option 4 (Work with TCP/IP port restrictions).

   c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).

   d. For the lower port range, specify `53`.

   e. For the upper port range, specify `*ONLY`.

      Notes:

      1) The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.

      2) RFC1700 provides information about common port number assignments.

   f. For the protocol, specify `*TCP`.

   g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

   h. Repeat steps 2c through 2g for the *UDP (User datagram) protocol.

Secure the DNS server

Following are security considerations when you choose to run DNS on your iSeries system:

- The function that the DNS server provides is IP address translation and name translation. It does not provide any access to objects on your iSeries system. Your risk when an outsider accesses your DNS server is that the server provides an easy way to view the topology of your network. Your DNS might save a hacker some effort in determining the addresses of potential targets. However, your DNS does not provide information that will help to break into those target systems.
- Typically, you use the iSeries DNS server for your intranet. Therefore, you probably do not have a need to restrict the ability to query the DNS. However, you might, for example, have several subnetworks within your intranet. You might not want users from a different subnetwork to be able to query the DNS on your iSeries server. A security option of DNS lets you limit access to a primary domain. Use iSeries Navigator to specify IP addresses to which the DNS server should respond.

  Another security option lets you specify which secondary servers can copy information from your primary DNS server. When you use this option, your server will accept zone transfer requests (a request to copy information) only from the secondary servers that you explicitly list.
- Be sure to carefully restrict the ability to change the configuration file for your DNS server. Someone with malicious intent could, for example, change your

DNS file to point to an IP address outside your network. They could simulate a server in your network and, perhaps, gain access to confidential information from users that visit the server.

Security considerations for using HTTP server for iSeries

The HTTP server provides World Wide Web browser clients with access to iSeries server multimedia objects, such as HTML (Hypertext Markup Language) documents. It also supports the *Common Gateway Interface (CGI)* specification. Application programmers can write CGI programs to extend the functionality of the server.

The administrator can use Internet Connection Server or IBM HTTP server for iSeries to run multiple servers concurrently on the same iSeries server. Each server that is running is called a **server instance**. Each server instance has a unique name. The administrator controls which instances are started and what each instance can do.

Note: You must have the *ADMIN instance of the HTTP server running when you use a Web browser to configure or administer any of the following:

- Firewall for iSeries
- Network Station™
- Internet Connection Server
- Internet Connection Secure Server
- IBM HTTP Server for AS/400

A user (Web site visitor) never sees an iSeries server Sign On display. However, the iSeries server administrator must explicitly authorize all HTML documents and CGI programs by defining them in HTTP directives. In addition, the administrator can set up both resource security and user authentication (user ID and password) for some or all requests.

An attack by a hacker could result in a denial of service to your Web server. Your server can detect a denial-of-service attack by measuring the time-out of certain clients' requests. If the server does not receive a request from the client, then your server determines that a denial-of-service attack is in progress. This occurs after making the initial client connection to your server. The server's default is to perform attack detection and penalization.

Prevent HTTP access

If you *do not* want anyone to use the program to access your system, you should prevent the HTTP server from running. Do the following:

__ Step 1. To prevent HTTP server jobs from starting automatically when you start TCP/IP, type the following:

```
CHGHTTPA AUTOSTART(*NO)
```

Notes:

a. AUTOSTART(*NO) is the default value.

b. "Control which TCP/IP servers start automatically" on page 124 provides more information about controlling which TCP/IP servers start automatically.

__ Step 2. By default, the HTTP server job uses the QTMHHTTP user profile. To prevent the HTTP server from starting, set the status of the QTMHHTTP user profile to *DISABLED.

Control access to the HTTP server

The primary purpose of running an HTTP server is to provide access for visitors to a Web site on your iSeries system. You might think of someone who visits your Web site as you would think of someone who views an advertisement in a trade journal. The visitor is not aware of the hardware and software running your Web site, such as the type of server you are using, and where your server is physically located. Usually, you do not want to put any barrier (such as a Sign On display) between a potential visitor and your Web site. However, you might want to restrict access to some of the documents or CGI programs that your Web site provides.

You might also want a single iSeries system to provide multiple logical Web sites. For example, your iSeries system might support different branches of your business that have different customer sets. For each of these branches of the business, you want a unique Web site that appears totally independent to the visitor. Additionally, you might want to provide internal Web sites (an intranet) with confidential information about your business.

As a security administrator, you need to protect the contents of your Web site while, at the same time, you need to ensure that your security practices do not negatively affect the value of your Web site. In addition, you need to ensure that HTTP activity does not jeopardize the integrity of your system or your network. The topics that follow provide security suggestions when you use the program.

Administration considerations

Following are some security considerations for administering your Internet server.

- You perform setup and configuration functions by using a Web browser and the *ADMIN instance. For some functions, such as creating additional instances on the server, you *must* use the *ADMIN server.
- The default URL for the administration home page (the home page for the *ADMIN server) is published in the documentation for products that provide browser administration functions. Therefore, the default URL will probably be known by hackers and published in hacker forums, just like the default passwords for IBM-supplied user profiles are known and published. You can protect yourself from this exposure in several ways:
  - Only run the *ADMIN instance of the HTTP server when you need to perform administrative functions. Do not have the *ADMIN instance running all the time.
  - Activate SSL support for the *ADMIN instance (by using Digital Certificate Manager). The *ADMIN instance uses HTTP protection directives to require a user ID and password. When you use SSL, your user ID and password are encrypted (along with all the other information about your configuration that appears on the administration forms).
  - Use a firewall both to prevent access to the *ADMIN server from the Internet and to hide your system and domain names, which are part of the URL.
- When you perform administration functions, you must sign on with a user profile that has *IOSYSCFG special authority. You might also need authority to specific objects on the system, such as the following:
  - The libraries or directories that contain your HTML documents and CGI programs.

– Any user profiles that you plan to swap to within the directives for the server.
– The Access Control Lists (ACLs) for any directories that your directives use.
– A validation list object for creating and maintaining user IDs and passwords.

With both the *ADMIN server and TELNET, you have the capability to perform administration functions remotely, perhaps over an Internet connection. Be aware that if you perform administration over a public link (the Internet), you might be exposing a powerful user ID and password to sniffing. The ″sniffer″ can then use this user ID and password to attempt to access your system using, for example, TELNET or FTP.

Notes:

1. With TELNET, the Sign On display is treated like any other display. Although the password does not display when you type it, the system transmits it without any encryption or encoding.
2. With the *ADMIN server, the password is encoded not encrypted. The encoding scheme is an industry standard, and thus commonly known among the hacker community. Although the encoding is not easily understood by the casual ″sniffer,″ a sophisticated sniffer probably has tools to attempt to decode the password.

Security tip

If you plan to perform remote administration over the Internet, you should use the *ADMIN instance with SSL, so that your transmissions are encrypted. Do not use an insecure application, such as a pre-V4R4 version of TELNET (TELNET supports SSL beginning with V4R4). If you are using the *ADMIN server across an intranet of *trusted* users, you can probably safely use this for administration.

- The HTTP directives provide the foundation for all activity on your server. The shipped configuration provides the capability to serve a default Welcome page. A client cannot view any documents except the Welcome page until the server administrator defines directives for the server. To define directives, use a Web browser and the *ADMIN server or the Work with HTTP Configuration (WRKHTTPCFG) command. Both methods require *IOSYSCFG special authority. When you connect your iSeries server to the Internet, it becomes even more critical to evaluate and control the number of users in your organization who have *IOSYSCFG special authority.

Protect resources

The IBM HTTP server for iSeries includes HTTP directives that can provide detailed control of the information assets that the server uses. You can use directives to control from which directories the Web server serves URLs for both HTML files and CGI programs, to swap to other user profiles, and to require authentication for some resources.

Note: The documentation under ″Web serving″ in the Information Center provides complete descriptions of the available HTTP directives and how to use them. Following are some suggestions and considerations for using this support:

- The HTTP server starts from the basis of ″explicit authority.″ The server does not accept a request unless that request is explicitly defined in the directives. In

other words, the server immediately rejects any request for a URL unless that URL is defined in the directives (either by name or generically).

- You can use protection directives to require a user ID and password before accepting a request for some or all of your resources.
  - When a user (client) requests a protected resource, the server challenges the browser for a user ID and password. The browser prompts the user to enter a user ID and password, and then sends the information to the server. Some browsers store the user ID and password and send them automatically with subsequent requests. This frees the user from repeatedly entering the same user ID and password on each request.

    Because some browsers store the user ID and password, you have the same user education task that you have when users enter your system through the iSeries server Sign On display or through a router. An unattended browser session represents a potential security exposure.
  - You have three options for how the system handles user IDs and passwords (specified in the protection directives):
    1. You can use normal iSeries server user profile and password validation. This is most commonly used to protect resources in an intranet (secure network).
    2. You can create ″Internet users″: users that can be validated but do not have a user profile on the iSeries server. Internet users are implemented through an iSeries server object called a ″validation list″. Validation list objects contain lists of users and passwords that are specifically defined for use with a particular application.

       You decide how Internet user IDs and passwords are supplied (such as by an application, or by an administrator in response to an e-mail request), as well as how to manage Internet users. Use the HTTP server's browser-based interface to set this up.

       For nonsecure networks (the Internet), using Internet users provides better overall protection than using normal user profiles and passwords. The unique set of user IDs and passwords creates a built-in limitation on what those users can do. The user IDs and passwords are not available for normal sign-on (such as with TELNET or FTP). In addition, you are not exposing normal user IDs and passwords to sniffing.
    3. Lightweight directory access protocol (LDAP) is a directory service protocol that provides access to a directory over a Transmission Control Protocol (TCP). It lets you store information in that directory service and query it. LDAP is now supported as a choice for user authentication.

    Notes:

    1. When the browser sends the user ID and the password (whether for an user profile or an Internet user), they are encoded, not encrypted. The encoding scheme is an industry standard, and thus commonly known among the hacker community. Although the encoding is not easily understood by the casual ″sniffer,″ a sophisticated sniffer probably has tools to attempt to decode them.
    2. The iSeries server stores the validation object in a protected system area. You can access it only with defined system interfaces (APIs) and proper authorization.
  - You can use Digital Certificate Manager (DCM) to create your own intranet Certificate Authority. Digital Certificate automatically associates a certificate with the owner's user profile. The certificate has the same authorizations and permissions as the associated profile.

- When the server accepts a request, normal iSeries server resource security takes over. The user profile that requests the resource must have authority to the resource (such as the folder or source physical file that contains the HTML document). By default, jobs run under the QTMHHTTP user profile. You can use a directive to swap to a different user profile. The system then uses that user profile's authority to access objects. Following are some considerations for this support:
  - Swapping user profiles can be particularly useful when your server provides more than one logical Web site. You can associate a different user profile with the directives for each Web site, and thus use normal iSeries server resource security to protect the documents for each site.
  - You can use the ability to swap user profiles in combination with the validation object. The server uses a unique user ID and password (separate from your normal user ID and password) to evaluate the initial request. After the server has authenticated the user, the system then swaps to a different user profile and thus takes advantage of resource security. The user is, thus, not aware of the true user profile name and cannot attempt to use it in other ways (such as FTP).
- Some HTTP server requests need to run a program on the HTTP server. For example, a program might access data on your system. Before the program can run, the server administrator must map the request (URL) to a specific user-defined program that conforms to CGI user-interface standards. Following are some considerations for CGI programs:
  - You can use the protection directives for CGI programs just as you do for HTML documents. Thus, you can require a user ID and password before running the program.
  - By default, CGI programs run under the QTMHHTP1 user profile. You can swap to a different user profile before running the program. Therefore, you can set up normal iSeries server resource security for the resources that your CGI programs access.
  - As security administrator, you should perform a security review before authorizing the use of any CGI program on your system. You should know where the program came from and what functions the CGI program performs. You should also monitor the capabilities of the user profiles under which you run CGI programs. You should also perform testing with CGI programs to determine, for example, whether you can gain access to a command line. Treat CGI programs with the same vigilance that you treat programs that adopt authority.
  - In addition, be sure to evaluate what sensitive objects might have inappropriate public authority. A poorly designed CGI program might, in rare cases, allow a knowledgeable, devious user to attempt to roam your system.
  - Use a specific user library, such as CGILIB, to hold all your CGI programs. Use object authority to control both who can place new objects in this library and who can run programs in this library. Use the directives to limit the HTTP server to running CGI programs that are in this library.

    Note: If your server provides multiple logical Web sites, you might want to set up a separate library for the CGI programs for each site.

Other security considerations

Following are additional security considerations:

- HTTP provides read-only access to your iSeries system. HTTP server requests cannot update or delete data on your system directly. However, you might have CGI programs that update data. Additionally, you can enable the Net.Data® CGI

program to access your iSeries server database. The system uses a script (which is similar to an exit program) to evaluate requests to the Net.Data program. Therefore, the system administrator can control what actions the Net.Data program can take.

- The HTTP server provides an access log that you can use to monitor both accesses and attempted accesses through the server.
- The *HTTP Server for iSeries Webmaster's Guide* provides more information about security considerations.

Security considerations for using SSL with IBM HTTP Server for iSeries

IBM HTTP Server for iSeries can provide secure Web connections to your iSeries server. A **secure web site** means that transmissions between the client and the server (in both directions) are encrypted. These encrypted transmissions are safe both from the scrutiny of sniffers and from those who attempt either to capture or to alter the transmissions.

Note: Keep in mind that a secure Web site applies strictly to the security of the information that passes between client and server. The intent of this is not to reduce your server's vulnerability to hackers. However, it certainly limits the information that a would-be hacker can obtain easily through sniffing.

The topics on SSL and Webserving (HTTP) in the information center provides complete information for installing, configuring, and managing the encryption process. These topics provide both an overview of the server features and some considerations for using the server.

Internet Connection Server provides HTTP and HTTPS support when one of the following licensed programs is installed:

- 5722–NC1
- 5722–NCE

When these options are installed, the product is referred to as the Internet Connection Secure Server.

IBM HTTP Server for iSeries (5722–DG1) provides both http and https support. You must install one of the following cryptographic products to enable SSL:

- 5722–AC2
- 5722–AC3

Security that depends on encryption has several requirements:

- Both the sender and receiver (server and client) must ″understand″ the encryption mechanism and be able to perform encryption and decryption. The HTTP server requires an SSL-enabled client. (Most popular Web browsers are SSL-enabled.) The iSeries encryption licensed programs support several industry-standard encryption methods. When a client attempts to establish a secure session, the server and client negotiate to find the most secure encryption method that both of them support.
- The transmission must not be able to be decrypted by an eavesdropper. Thus, encryption methods require both parties to have an encryption/decryption **private key** that only they know. If you want to have a secure *external* Web site, you should use an independent certificate authority (CA) to create and issue digital certificates to users and servers. The certificate authority is known as a trusted party.

Encryption protects the confidentiality of transmitted information. However, for sensitive information, such as financial information, you want integrity and authenticity in addition to confidentiality. In other words, the client and (optionally) the server must trust the party on the other end (through an independent reference) and they must be sure that the transmission has not been altered. The digital signature that is provided by a certification authority (CA) provides these assurances of authenticity and integrity. The SSL protocol provides authentication by verifying the digital signature of the server's certificate (and optionally the client's certificate).

Encryption and decryption require processing time and will affect the performance of your transmissions. Therefore, iSeries servers provide the capability to run both the programs for secure and insecure serving at the same time. You can use the insecure HTTP server to serve documents that do not require security, such as your product catalog. These documents will have a URL that starts with `http://`. You can use a secure HTTP server for sensitive information such as the form where the customer enters credit card information. The program can serve documents whose URL starts either with `http://` or with `https://`.

Reminder

It is good Internet etiquette to inform your clients when transmissions are secure and not secure, particularly when your Web site only uses a secure server for some documents.

Keep in mind that encryption requires both a secure client and a secure server. Secure browsers (HTTP clients) have become fairly common.

Security considerations for LDAP

Lightweight Directory Access Protocol (LDAP) security features include Secure Sockets Layer (SSL), Acess Control Lists, and CRAM-MD5 password encryption. In V5R1, Kerberos connections and Security auditing support were added to enhance LDAP security.

For more information on these topics, refer to the iSeries Information Center—>Networking–>TCP/IP–>Directory Services (LDAP). See "Prerequisite and related information" on page xii for information on accessing the iSeries Information Center.

Security considerations for LPD

LPD (line printer daemon) provides the capability to distribute printer output to your system. The system does not perform any sign-on processing for LPD.

Prevent LPD access

If you *do not* want anyone to use LPD to access your system, you should prevent the LPD server from running. Do the following:

__ Step 1. To prevent LPD server jobs from starting automatically when you start TCP/IP, type the following:

```
CHGLPDA AUTOSTART(*NO)
```

Notes:

a. AUTOSTART(*YES) is the default value.

b. "Control which TCP/IP servers start automatically" on page 124 provides more information about controlling which TCP/IP servers start automatically.

__ Step 2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for LPD, do the following:

__ Step a. Type `GO CFGTCP` to display the Configure TCP/IP menu.

__ Step b. Select option 4 (Work with TCP/IP port restrictions).

__ Step c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).

__ Step d. For the lower port range, specify `515`.

__ Step e. For the upper port range, specify `*ONLY`.

Notes:

1) The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.
2) RFC1700 provides information about common port number assignments.

__ Step f. For the protocol, specify `*TCP`.

__ Step g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

__ Step h. Repeat steps 2c through 2g for the *UDP protocol.

Control LPD access

If you want to allow LPD clients to access your system, be aware of the following security issues:

- To prevent a user from swamping your system with unwanted objects, be sure that you have set adequate threshold limits for your auxiliary storage pools (ASPs). You can display and set the thresholds for ASPs by using either system service tools (SST) or dedicated service tools (DST). The *Backup and Recovery* book provides more information about ASP thresholds.
- You can use the authority to output queues to restrict who can send spooled files to your system. LPD users without a user ID use the QTMPLPD user profile. You can give this user profile access to only a few output queues.

Security considerations for SNMP

The iSeries server can act as a simple network management protocol (SNMP) agent in a network. SNMP provides a means for managing the gateways, routers, and hosts in a network environment. An SNMP agent gathers information about the system and performs functions that remote SNMP network managers request.

Prevent SNMP access

If you *do not* want anyone to use SNMP to access your system, you should prevent the SNMP server from running. Do the following:

__ Step 1. To prevent SNMP server jobs from starting automatically when you start TCP/IP, type the following:

```
CHGSNMPA AUTOSTART(*NO)
```

Notes:

a. AUTOSTART(*YES) is the default value.

b. "Control which TCP/IP servers start automatically" on page 124 provides more information about controlling which TCP/IP servers start automatically.

__ Step 2. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for SNMP, do the following:

__ Step a. Type `GO CFGTCP` to display the Configure TCP/IP menu.

__ Step b. Select option 4 (Work with TCP/IP port restrictions).

__ Step c. On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).

__ Step d. For the lower port range, specify `161`.

__ Step e. For the upper port range, specify `*ONLY`.

Notes:

1) The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restrictions, you should end TCP/IP and start it again.

2) RFC1700 provides information about common port number assignments.

__ Step f. For the protocol, specify `*TCP`.

__ Step g. For the user profile field, specify a user profile name that is protected on your system. (A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users.) By restricting the port to a specific user, you automatically exclude all other users.

__ Step h. Repeat steps 2c through 2g for the *UDP protocol.

Control SNMP access

If you want to allow SNMP managers to access your system, be aware of the following security issues:

- Someone who can access your network with SNMP can gather information about your network. Information that you have hidden by using aliases and a domain-name server becomes available to the would-be intruder through SNMP. Additionally, an intruder might use SNMP to alter your network configuration and disrupt your communications.
- SNMP relies on a community name for access. Conceptually, the community name is similar to a password. The community name is not encrypted. Therefore, it is vulnerable to sniffing. Use the Add Community for SNMP (ADDCOMSNMP) command to set the manager internet address (INTNETADR) parameter to one or more specific IP addresses instead of *ANY. You can also set the OBJACC parameter of the ADDCOMSNMP or CHGCOMSNMP commands to *NONE to prevent the managers in a community from accessing any MIB objects. This is intended to just be done temporarily to deny access to managers in a community without removing the community.

Security considerations for INETD server

Unlike most TCP/IP servers, the INETD server does not provide one single service to clients. Instead, it provides a variety of miscellaneous services that administrators can customize. For that reason, the INETD server is sometimes called ″the super server″. The INETD server has the following built-in services:

- time
- daytime
- echo
- discard
- changed

These services are supported for both TCP and UDP. For UDP, the echo, time, daytime, and changed services receive UDP packets, then send the packets back to the originator. The echo server echoes back packets that it receives, the time and daytime servers generate the time in a specific format and sends it back, and the changed server generates a packet of printable ASCII characters and sends it back.

The nature of these UDP services makes a system vulnerable to a denial of service attack. For example, assume that you have two iSeries servers: SYSTEMA and SYSTEMB. A malicious programmer could forge the IP header and the UDP header with a source address of SYSTEMA and a UDP port number of the time server. He can then send that packet to the time server on SYSTEMB, which will send the time to SYSTEMA, which will respond back to SYSTEMB, and so on, generating a continuous loop and consuming CPU resources on both systems, as well as network bandwidth.

Therefore, you should consider the risk of such an attack on your iSeries system, and only run these services on a secure network. The INETD server is shipped to not be auto started when you start TCP/IP. You can configure whether or not to start the services when INETD is started. By default, the TCP and UDP time servers and daytime servers are both started when you start the INETD server.

There are two configuration files for the INETD server:

```
/QIBM/UserData/OS400/inetd/inetd.conf
/QIBM/ProdData/OS400/inetd/inetd.conf
```

These files determine what programs start when the INETD server starts. They also determine what user profile these programs are running under when INETD starts them.

Note: The configuration file in proddata should never be modified. It is replaced each time the system is reloaded. Customer configuration changes should only be placed in the file, in the userdata directory tree, as this file in **not** updated during release upgrades.

If a malicious programmer got access to these files, she could configure them to start any program when INETD started. Therefore it is very important to protect these files. By default they require QSECOFR authority to make changes. You should not reduce the authority required to access them.

Note: Do not modify the configuration file in the ProdData directory. That file is replaced each time that the system is reloaded. Customer configuration changes should only be placed in the file in the UserData directory tree, as that file is not updated during release upgrades.

Security considerations for limiting TCP/IP roaming

If your system is connected to a network, you may want to limit your users' ability to roam the network with TCP/IP applications. One way to do this is to restrict access to the following client TCP/IP commands:

Note: These commands might exist in several libraries on your system. They are in both the QSYS library and the QTCP library, at a minimum. Be sure to locate and secure all occurrences.

- STRTCPFTP
- FTP
- STRTCPTELN
- TELNET
- LPR
- SNDTCPSPLF
- RUNRMTCMD (REXEC client)

Your users' possible destinations are determined by the following:

- Entries in your TCP/IP host table.
- *DFTROUTE entry in the TCP/IP route table. This allows users to enter the IP address of the next-hop system when their destination is an unknown network. A user can reach or contact a remote network by using the default route.
- Remote name server configuration. This support allows another server in the network to locate host names for your users.
- Remote system table.

You need to control who can add entries to these tables and change your configuration. You also need to understand the implications of your table entries and your configuration.

Be aware that a knowledgeable user with access to an ILE C compiler can create a socket program that can attach to a TCP or UDP port. You can make this more difficult by restricting access to the following sockets interface files in the QSYSINC library:

- SYS
- NETINET
- H
- ARPA
- sockets and SSL

For service programs, you can restrict use of socket and SSL applications that are already compiled by restricting use of these service programs:

- QSOSRV1
- QSOSRV2
- QSOSKIT(SSL)
- QSOSSLSR(SSL)

The service programs are shipped with public authority *USE, but the authority can be changed to *EXCLUDE (or another value as needed).

Chapter 15. Secure workstation access

Many of your system users have personal computers (PCs) on their desks as their workstations. They use tools that run on the PC, and they use the PC to connect to iSeries server.

Most methods of connecting a PC to iSeries servers provide more function than workstation emulation. The PC may look like a display to iSeries and provide the user with interactive sign-on sessions. In addition, the PC may look to iSeries servers like another computer and provide functions such as file transfer and remote procedure call.

As an iSeries server security administrator, you need to be aware of the following:

- Functions that are available to PC users who are connected to your system
- iSeries server resources that PC users can access.

You may want to prevent advanced PC functions (such as file transfer and remote procedure call) if your iSeries server security scheme is not yet prepared for those functions. Probably, your long-range goal is to allow advanced PC functions while you still protect the information on your system. The topics that follow discuss some of the security issues that are associated with PC access.

Prevent workstation viruses

This information suggests ways that security administrators can safeguard against PC viruses.

Secure workstation data access

Some PC client software uses shared folders to store information on the server. To access iSeries database files, the PC user has a limited, well-defined set of interfaces. With the file transfer capability that is part of most client/server software, the PC user can copy files between the server and the PC. With database access capability; such as a DDM file, remote SQL, or an ODBC driver; the PC user can access data on the server.

In this environment, you can create programs to intercept and evaluate PC-user requests to access server resources. When the requests use a DDM file, you specify the exit program in the distributed data management access (DDMACC) network attribute. For some methods of PC file transfer, you specify the exit program in the client request access (PCSACC) network attribute. Or, you can specify PCSACC (*REGFAC) to use the registration function. When the requests use other server functions to access data, you can use the WRKREGINF command to register exit programs for those server functions.

Exit programs, however, can be difficult to design, and they are rarely foolproof. Exit programs are not a replacement for object authority, which is designed to protect your objects from unauthorized access from any source.

Some client software, such as IBM iSeries Access for Windows, uses the integrated file system to store and access data on iSeries servers. With the integrated file system, the entire server becomes more easily available to PC users. Object

© Copyright IBM Corp. 1996, 2002

authority becomes even more essential. Through the integrated file system, a user with sufficient authority can view a server library as if it is a PC directory. Simple move and copy commands can instantly move data from an iSeries server library to a PC directory or vice versa. The system automatically makes the appropriate changes to the format of the data.

Notes:

1. You can use an authorization list to control the use of objects in the QSYS.LIB file system. See "Restrict access to the QSYS.LIB file system" on page 103 for more information.
2. Chapter 12, "Use Integrated File System to secure files" on page 97 provides more information about security issues with the integrated file system.

The strength of the integrated file system is its simplicity for users and developers. With a single interface, the user can work with objects in multiple environments. The PC user does not need special software or APIs to access objects. Instead, the PC user can use familiar PC commands or "point and click" to work with objects directly.

For all systems that have PCs attached, but particularly for systems that have client software that uses the integrated file system, a good object authority scheme is critical. Because security is integrated into the OS/400 product, any request to access data must go through the authority checking process. Authority checking applies to requests from any source and to data access that uses any method.

Object authority with workstation access

When you set up authority for objects, you need to evaluate what that authority provides for the PC user. For example, when a user has *USE authority to a file, the user can view or print data in the file. The user cannot change information in the file or delete the file. For the PC user, viewing is equivalent to "reading", which provides sufficient authority for the user to make a copy of a file on the PC. This may not be what you intend.

For some critical files, you may need to set the public authority to *EXCLUDE to prevent downloading. You can then provide another method to "view" the file on the server, such as using a menu and programs that adopt authority.

Another option to prevent downloading is to use an exit program that runs whenever a PC user starts a server function (other than interactive sign-on). You can specify an exit program in the PCSACC network attribute by using the Change Network Attribute (CHGNETA) command. Or, you can register exit programs by using the Work with Registration Information (WRKREGINF) command. The method that you use depends on how PCs are accessing data on your system and which client program the PCs use. The exit program (QIBM_QPWFS_FILE_SERV) applies to iSeries Access and Net Server access to IFS. It does not prevent access from a PC with other mechanisms, such as FTP or ODBC.

PC software typically provides upload capability also, so that a user can copy data from the PC to a server database file. If you have not set up your authority scheme correctly, a PC user might overlay all of the data in a file with data from a PC. You need to assign *CHANGE authority carefully. Review Appendix D in the *iSeries Security Reference* book to understand what authority is required for file operations.

The iSeries Information Center provides more information about the authority for PC functions and about using exit programs. See “Prerequisite and related information” on page xii for details.

Application Administration

Application Administration is an optionally-installable component of iSeries Navigator, the graphical user interface (GUI) for the iSeries server. Application Administration allows system administrators to control the functions or applications available to users and groups on a specific server. This includes controlling the functions available to users that access their server through clients. It is important to note here, that if you access the server from a Windows client, the iSeries server user and not the Windows user determines which functions are available for administration.

For complete documentation on iSeries Navigator Application Administration, refer to the iSeries Information Center–>Connecting to iSeries—>What to connect with—>iSeries Navigator
(../html/as400/v5r2/ic2924/info/rzaj3/rzaj3overview.htm).

Policy administration

Policies are a tool for administrators to use as they configure software on their client PCs. Policies can restrict which functions and applications a user has access to on the PC. Policies can also suggest or mandate configurations to be used by certain users or certain PCs.

Note: Polices do not offer control over server resources. Policies are not a substitute for server security. Policies can be used to affect how iSeries Access is able to access the server from a particular PC, by a particular user. However, they do not change how server resources can be accessed via other mechanisms.

Policies are stored on a file server. Each time the user signs on to their Windows workstation, the policies that apply to that Windows user are downloaded from the file server. The policies are applied to the registry before the user does anything on the workstation.

Microsoft® policies versus application administration

iSeries Access Express supports two different strategies for implementing administrative control within your network: Microsoft system policies and iSeries Navigator Application Administration. Consider the following when deciding which strategy is best suited for your needs.

Microsoft system policies

Policies are PC driven, not dependent upon specific OS/400 releases. Policies can apply to PCs, as well as Windows users. This means that users refer to the Windows user profile, not the server user profile. Policies can be used to ″configure″ as well as to restrict. Policies typically will offer more granularity than Application Administration, and can offer a larger breadth of function. This is because a connection to the server is not needed to determine whether the user can use the function or not. Implementing policies is more complicated than implementing Application Administration because the use of the Microsoft system policy editor is required and PCs must be individually configured to download policies.

iSeries Navigator application administration

Application Administration associates data with the user profile, instead of the Windows profile that Microsoft system policies associate with. While iSeries servers running V4R3 or later of the OS/400 product are required in order to use Application Administration, some functions are only available at V4R4 or later. Application Administration uses the graphical user interface of iSeries Navigator to administer, which is much easier to use than policy editor. Application Administration info applies to the user regardless of which PC he signs on from. Particular functions within iSeries Navigator can be restricted. Application Administration is preferable if all of the functions you want to restrict are Application Administration-enabled, and if the version of OS/400 being used supports Application Administration.

Use SSL with iSeries Access for Windows

For information about using iSeries Access Express with SSL, review the iSeries Information Center topics *Secure Sockets Layer Administration, Securing iSeries Access Express and iSeries Navigator, iSeries Developer Kit for Java , and iSeries Java Toolbox* under the Java main topic. You may also review this information on the CD provided to you with your system.

iSeries Navigator security

iSeries Navigator provides an easy-to-use interface to your server for users who have iSeries Access. With each new release of the OS/400 product, more server function becomes available through iSeries Navigator. An easy-to-use interface provides many benefits, including reduced technical support costs and an improved image for your system. It also presents security challenges.

As a security administrator, you can no longer rely on the ignorance of your users to protect resources. iSeries Navigator makes many functions easy and visible for your users. You need to ensure that you have designed and implemented security policies for user profiles and for object security to meet your security needs.

V4R4 and later versions of iSeries Access for Windows provide the following methods to control the functions that users can perform through iSeries Navigator:

- Selective install
- Application administration
- Windows NT® system policy support

iSeries Navigator is packaged into multiple components that you can install separately. This allows you to install only the functions that you require. Application Administration allows an administrator to control the functions that a user or group can access through iSeries Navigator. Application Administration organizes applications into the following categories:

iSeries Navigator
Includes iSeries Navigator and any plug-ins.

Client applications
Includes all other client applications, including iSeries Access, that provide functions on clients that are administered through Application Administration.

Host applications
Includes all applications that reside entirely on your server and provide functions that are administered through Application Administration.

You can use selective install, application administration, and policies to limit the iSeries Navigator functions that a user can access. None of these, however, should be used for resource security.

Beginning in V4R4, iSeries Access for Windows also supports using the Windows NT System Policy Editor to control what functions can be performed from a particular PC client, regardless of who is using that PC.

See the iSeries Information Center for additional information on selective installation, Application Administration and Policy Administration. The "Limit access to program function" on page 9 section of this book also contains some discussion of application administration.

Prevent ODBC access

Open database connectivity (ODBC) is a tool that PC applications can use to access iSeries data as if the data is PC data. The ODBC programmer can make the physical location of the data transparent to the user of the PC application. For more information regarding ODBC security considerations, go to the ″iSeries Access for Windows ODBC security″ information (/rzaii/rzaiiodbc09.HTM), located in the iSeries Information Center.

Security considerations for workstation session passwords

Typically, when a PC user starts the connection software, such as iSeries Access, the user types the user ID and password for the server once. The password is encrypted and stored in PC memory. Whenever the user establishes a new session to the same server, the PC sends the user ID and password automatically.

Some client/server software also provides the option of bypassing the Sign On display for interactive sessions. The software will send the user ID and encrypted password when the user starts an interactive (5250 emulation) session. To support this option, the QRMTSIGN system value on the server must be set to *VERIFY.

When you choose to allow bypassing the Sign On display, you need to consider the security trade-offs.

Security exposure: For 5250 emulation or any other type of interactive session, the Sign On display is the same as any other display. Although the password is not displayed on the screen when it is typed, the password is sent over the link in unencrypted form just like any other data field. For some types of links, this may provide the opportunity for a would-be intruder to monitor the link and to detect a user ID and password. Monitoring a link by using electronic equipment is often referred to as **sniffing**. Beginning with V4R4, you can use secure sockets layer (SSL) to encrypt communication between iSeries Access and the iSeries server. This protects your data, including passwords, from sniffing.

When you choose the option to bypass the Sign On display, the PC encrypts the password before it is sent. Encryption avoids the possibility of having a password stolen by sniffing. However, you must ensure that your PC users practice operational security. An unattended PC with an active session to the iSeries system provides the opportunity for someone to start another session without knowing a user ID and password. PCs should be set up to lock when the system is inactive for an extended period, and they should require a password to resume the session.

Even if you do not choose to bypass the Sign On display, an unattended PC with an active session represents a security exposure. By using PC software, someone can start a server session and access data, again without knowing a user ID and a password. The exposure with 5250 emulation is somewhat greater because it requires less knowledge to start a session and begin accessing data.

You also need to educate your users about the effect of disconnecting their iSeries Access session. Many users assume (logically but incorrectly) that the disconnect option completely stops their connection to the server. In fact, when a user selects the option to disconnect, the server makes the user's session (license) available for another user. However, the client's connection to the server is still open. Another user could walk up to the unprotected PC and get access to server resources without ever entering a user ID and password.

You can suggest two options for your users who need to disconnect their sessions:

- Ensure that their PCs have a lockup function that requires a password. This makes an unattended PC unavailable to anyone who does not know the password.
- To completely disconnect a session, either log off Windows or restart (reboot) the PC. This ends the session to the iSeries.

You also need to educate your users about a potential security exposure when they use iSeries Access for Windows. When a user specifies a UNC (universal naming convention) to identify an iSeries resource, the Win95 or NT client builds a network connection to link to the server. Because the user specifies a UNC, the user does not see this as a mapped Network Drive. Often, the user is not even aware of the existence of the network connection. However, this network connection represents a security exposure on an unattended PC because the server appears in the directory tree on the PC. If the user's session has a powerful user profile, server resources might be exposed on an unattended PC. As with the previous example, the remedy is to ensure both that users understand the exposure and that they use their PC's lockup function.

Protect the server from remote commands and procedures

A knowledgeable PC user with software such as iSeries Access can run commands on the server without going through the Sign On display. The following are several methods that are available for PC users to run server commands. Your client/server software determines the methods that your PC users have available to them.

- A user can open a DDM file and use the remote command function to run a command.
- Some software, such as iSeries Access optimized clients, provides the remote command function through Distributed Program Call (DPC) APIs, without the use of DDM.
- Some software, such as remote SQL and ODBC, provides a remote command function without either DDM or DPC.

For client/server software that uses DDM for remote command support, you can use the DDMACC network attribute to prevent remote commands completely. For client/server software that uses other server support, you can register exit programs for the server. If you want to allow remote commands, you must make sure that your object authority scheme protects your data adequately. Remote command capability is equivalent to giving a user a command line. In addition,

when iSeries receives a remote command through DDM, the system does not enforce the user profiles Limited capability (LMTCPB) setting.

Protect workstations from remote commands and procedures

IBM iSeries Access for Windows provides the capability of receiving remote commands on the PC. You can use the Run Remote Command (RUNRMTCMD) command on the server to run a procedure on an attached PC. The RUNRMTCMD capability is a valuable tool for system administrators and help-desk personnel. However, it also provides the opportunity for damaging PC data, either deliberately or accidentally.

PCs do not have the same object authority functions as iSeries servers. Your best protection against problems with the RUNRMTCMD command is to carefully restrict the system users who have access to the command. IBM iSeries Access for Windows provides the capability to register which users can run remote commands on a specific PC. When the connection is via TCP/IP, you can use the properties control panel on the client to control remote-command access. You can authorize users by user ID or by the remote system name. When the connection is via SNA, some client software provides the capability to set up security for the conversation. With other client software, you simply choose whether or not to set up the incoming-command capability.

For each combination of client software and connection type (such as TCP/IP or SNA), you need to review the potential for incoming-commands to attached PCs. Consult the client documentation by searching for “incoming command” or “RUNRMTCMD”. Be prepared to advise your PC users and network administrators about the correct (secure) way to configure clients to permit or prevent this capability.

Gateway servers

Your system may participate in a network with an intermediate or gateway server between the iSeries system and the PCs. For example, your iSeries system might be attached to a LAN with a PC server that has PCs that are attached to the server. The security issues in this situation depend on the capabilities of the software that is running on the gateway server. Figure 13 shows an example of a gateway-server configuration:

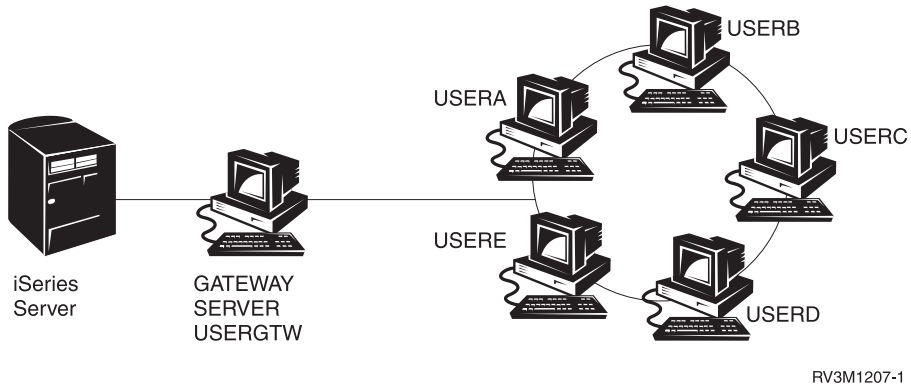


Figure 13. iSeries system with a gateway server

With some software, your iSeries system will not know about any users (such as USERA or USERC) who are downstream from the gateway server. The server will

sign on to the system as a single user (USERGTW). It will use the USERGTW user ID to handle all requests from downstream users. A request from USERA will look to the server like a request from user USERGTW.

If this is the case, you must rely on the gateway server for security enforcement. You must understand and manage the security capabilities of the gateway server. From an iSeries server perspective, every user has the same authority as the user ID that the gateway server uses to start the session. You might think of this as equivalent to running a program that adopts authority and provides a command line.

With other software, the gateway server passes requests from individual users to iSeries servers. The iSeries server knows that USERA is requesting access to a particular object. The gateway is almost transparent to the sytem.

If your system is in a network that has gateway servers, you need to evaluate how much authority to provide to the user IDs that are used by the gateway servers. You also need to understand the following:
- The security mechanisms that the gateway servers enforce.
- How downstream users will appear to your iSeries system.

Wireless LAN communications

Some clients might use the iSeries Wireless LAN to communicate to your system without wires. The iSeries Wireless LAN uses radio-frequency communications technology. As a security administrator, you should be aware of the following security characteristics of iSeries Wireless LAN products:

- These wireless LAN products use spread spectrum technology. This same technology has been used by the government in the past to secure radio transmissions. To someone who attempts to electronically monitor for data transmissions, the transmissions appear to be noise rather than an actual transmission.
- The wireless connection has three security-relevant configuration parameters:
  - Data rate (two possible data rates)
  - Frequency (five possible frequencies)
  - System identifier (8 million possible identifiers)

  These configuration elements combine to provide 80 million possible configurations, which makes a hacker's likelihood of guessing the correct configuration extremely slim.
- Just like with other communications methods, the security of wireless communications is affected by the security of the client device. The system ID information and other configuration parameters are in a file on the client device and should be protected.
- If a wireless device is lost or stolen, normal server security measures, such as sign-on passwords and object security, provide protection when an unauthorized user attempts to use the lost or stolen unit to access your system.
- If a wireless client unit is lost or stolen, you should consider changing the system ID information for all users, access points, and systems. Think of this as changing the locks on your doors if a set of keys is stolen.
- You might want to partition your server into groups of clients that have unique system IDs. This limits the impact if a unit is lost or stolen. This method works only if you can confine a group of users to a specific portion of your installation.

- Unlike wired LAN technology, wireless LAN technology is proprietary. Therefore, no electronic sniffers are publicly available for these wireless LAN products. A sniffer is an electronic device that performs unauthorized monitoring of a transmission.

Chapter 16. Security exit programs

Some iSeries server functions provide an exit so that your system can run a user-created program to perform additional checking and validation. For example, you can set up your system to run an exit program every time that someone attempts to open a DDM (distributed data management) file on your system. You can use the registration function to specify exit programs that run under certain conditions.

Several iSeries publications contain examples of exit programs that perform security functions. Table 24 provides a list of these exit programs and sources for example programs.

Table 24. Sources of Sample Exit Programs

Type of exit program	Purpose	Where to find examples
Password validation	The QPWDVLDPGM system value can specify a program name or indicate that validation programs registered for the QIBM_QSY_VLD_PASSWRD exit point be used to check a new password for additional requirements that are not handled by the QPWDxxx system values. The use of this program should be carefully monitored because it receives unencrypted passwords. This program **should not** store passwords in a file or pass them to another program.	• *An Implementation Guide for iSeries Security and Auditing, GG24–4200* • *iSeries Security Reference*, SC41-5302-06
PC Support/400 or Client Access access[1]	You can specify this program name in the Client request access (PCSACC) parameter of the network attributes to control the following functions: • Virtual printer function • File transfer function • Shared folders Type 2 function • Client access message function • Data queues • Remote SQL function	*An Implementation Guide for iSeries Security and Auditing, GG24–4200*
Distributed Data Management (DDM) access	You can specify this program name in the DDM request access (DDMACC) parameter of the network attributes to control the following functions: • Shared folders Type 0 and 1 function • Submit Remote Command function	*An Implementation Guide for iSeries Security and Auditing, GG24–4200*
Remote sign on	You can specify a program in the QRMTSIGN system value to control what users can be automatically signed on from which locations (pass-through.)	*An Implementation Guide for iSeries Security and Auditing, GG24–4200*

© Copyright IBM Corp. 1996, 2002

Table 24. Sources of Sample Exit Programs (continued)

Type of exit program	Purpose	Where to find examples
Open Database Connectivity (ODBC) with iSeries Access[1]	Control the following functions of ODBC: • Whether ODBC is allowed at all. • What functions are allowed for iSeries database files. • What SQL statements are allowed. • What information can be retrieved about database server objects. • What SQL catalog functions are allowed.	None available.
QSYSMSG break handling program	You can create a program to monitor the QSYSMSG message queue and take appropriate action (such as notifying the security administrator) depending on the type of message.	*An Implementation Guide for iSeries Security and Auditing, GG24–4200*
TCP/IP	Several TCP/IP servers (such as FTP, TFTP, TELNET, and REXEC) provide exit points. You can add exit programs to handle log-on and to validate user requests, such as requests to get or put a specific file. You can also use these exits to provide anonymous FTP on your system.	"TCP/IP User Exits in the *iSeries System API Reference* book"
User profile changes	You can create exit programs for the following user profile commands: CHGUSRPRF CRTUSRPRF DLTUSRPRF RSTUSRPRF	• *iSeries Security Reference*, SC41-5302-06 • "TCP/IP User Exits in the *iSeries System API Reference* book"

Notes:

1. Additional information on this topic can be found in the iSeries Information Center. See "Prerequisite and related information" on page xii for more details.

Chapter 17. Security considerations for Internet browsers

Many PC users in your organization have browsers on their workstations. They might connect to the Internet. They might also connect to your server. Following are some security considerations both for the PCs and for your server.

Risk: workstation damage

A Web page that your user visits might have an associated ″program,″ such as a Java applet, an Active-X control, or some other type of plug-in. Although it is rare, this type of ″program″ when run on a PC has the potential to damage the information on the PC. As a security administrator, consider the following for protecting PCs in your organization:

- Understand the security options of the different browsers that your users have. For example, with some browsers, you can control the access that Java applets have outside the browser (the restricted operating environment of Java is called the *sandbox*). This can prevent applets from damaging PC data.

  Note: The sandbox concept and its associated security restrictions do not exist for Active-X and other plug-ins.
- Make recommendations to your users about their browser settings. You probably do not have either the time or the resources to ensure that users follow your recommendations. Therefore, you must educate them about the potential risks of improper settings.
- Consider standardizing on Web browsers that provide the security options that you need.
- Instruct your users to inform you of any suspicious behavior or symptoms that might be associated with particular Web sites.

Risk: access to iSeries directories through mapped drives

Assume that a PC is connected to your server with an IBM iSeries Access for Windows session. The session set up mapped drives to link to the iSeries integrated file system. For example, the PC's **G** drive might map to the integrated file system of the SYSTEM1 server in the network.

Now assume that the same PC user has a browser and can access the Internet. The user requests a Web page that runs a mischievous ″program″ such as a Java applet or Active-X control. Conceivably, the program could attempt to erase everything on the PC's G drive.

You have several protections against damage to mapped drives:

- Your most important protection is resource security on your server. The Java applet or Active-X control looks to the server like the user who established the PC session. You need to carefully manage what PC users are authorized to do on your server.
- Advise your PC users to set their browsers to prevent attempts to access mapped drives. This works for Java applets but not for Active-X controls, which do not have the sandbox concept.
- Educate your users about the dangers of being connected to your server and the Internet in the same session. Also, make sure your PC users (with Windows 95

© Copyright IBM Corp. 1996, 2002

clients, for example) understand that drives remain mapped even when the iSeries Access session appears to be ended.

Risk: trusted signed applets

Your users might have followed your advice and set up their browsers to prevent applets from writing to any PC drives. However, your PC users need to be aware that a *signed applet* can override the setting for their browser.

A signed applet has an associated digital signature to establish its authenticity. When a user accesses a Web page that has a signed applet, the user sees a message. The message indicates the applet's signature (who signed it and when it was signed). When your user accepts the applet, the user grants the applet an override to the security settings for the browser. The signed applet can write to the PC's local drives, even though the default setting for the browser prevents it. The signed applet can also write to mapped drives on your server because they appear to the PC to be local drives.

For your own Java applets that come from your server, you might need to use signed applets. However, you should instruct your users in general not to accept signed applets from unknown sources.

Chapter 18. Related information

Manuals

- *APPC Programming*, SC41-5443-00describes the advanced program-to-program communications (APPC) support for the iSeries system. This book guides in developing application programs that use APPC and defining the communications environment for APPC communications. It includes application program considerations, configuration requirements and commands, problem management for APPC, and general networking considerations. See the iSeries: Information Center Supplemental CD-ROM.
- *AS/400 Internet Security: Protecting Your AS/400 from HARM on the Internet*, SG24-4929 discusses the security issues and risk associated with connecting your iSeries to the Internet. It provides examples, recommendations, tips and techniques for TCP/IP applications.
- *Backup and Recovery*, SC41-5304-06 provides information about planning a backup and recovery strategy, saving information from your system, and recovering your system. See the iSeries: Information Center Supplemental CD-ROM. Additional information on these topics can also be found in the iSeries Information Center. See "Prerequisite and related information" on page xii for more details.
- *CL Programming*, SC41-5721-05, provides detailed descriptions for coding the data description specifications (DDS) for files that can be described externally. These files are physical, logical, display, print, and intersystem communication function (ICF) files. See the iSeries: Information Center Supplemental CD-ROM
- The CL topic in the Information Center (See "Prerequisite and related information" on page xii for more details.) provides a description of all the iSeries control language (CL) and its OS/400 commands. The OS/400 commands are used to request functions of the Operating System/400 (5738-SS1) licensed program. All the non-OS/400 CL commands--those associated with the other licensed programs, including all the various languages and utilities--are described in other books that support those licensed programs.
- *Implementing iSeries Security, 3rd Edition* by Wayne Madden and Carol Woodbury. Loveland, Colorado: 29th Street Press, a division of Duke Communications International, 1998. Provides guidance and practical suggestions for planning, setting up, and managing iSeries security.

  ISBN Order Number:

  1-882419-78-2
- *HTTP Server for iSeries Webmaster's Guide*, GC41-5434-08, provides the system administration with information for configuring and managing the Internet Connection Server and the Internet Connection Secure Server. For more information regarding HTTP server, see the following URL:

  `http://www.ibm.com/eserver/iseries/software/http/docs/doc.htm`
- *iSeries Security Reference*, SC41-5302-06, provides complete information about security system values, user profiles, resource security, and security auditing. This manual does not describe security for specific licensed programs, languages, and utilities. See the iSeries: Information Center Supplemental CD-ROM.

© Copyright IBM Corp. 1996, 2002

- The ″Basic system operations″ topic in the Information Center provides information on some of the key concepts and tasks required for iSeries basic operations. See “Prerequisite and related information” on page xii for more details.
- The Information Center describes how to use and configure TCP/IP and the several TCP/IP applications, such as FTP, SMTP, and TELNET. See “Prerequisite and related information” on page xii for more details. .
- *TCP/IP File Server Support for OS/400 Installation and User's Guide*, SC41-0125, provides introductory information, installation instructions, and setup procedures for the File Server Support licensed program offering. It explains the functions available with the product and includes examples and hints for using it with other systems.
- *Trusted Computer Systems Evaluation Criteria* DoD 5200.28.STD, describes the criteria for levels of trust for computer systems. The TCSEC is a publication of the United States government. Copies may be obtained from:

  Office of Standards and Products
  National Computer Security Center
  Fort Meade, Maryland 20755-6000 USA
  Attention: Chief, Computer Security Standards
- The Information Center contains several topics regarding System Management and Work Management on the iSeries. Some of these topics include performance data collection, system values management, and storage management. For details on accessing the Information Center, see “Prerequisite and related information” on page xii. Work Management, SC41-5306-03, provides information about how to create and change a work management environment. See the iSeries: Information Center Supplemental Manuals CD-ROM.

In addition to these Information Center topics and Supplemental Manuals, you can use the following resources for assistance:

- **IBM SecureWay**
  IBM SecureWay provides a common brand for IBM's broad portfolio of security offerings; hardware, software, consulting and services to help customers secure their information technology. Whether addressing an individual need or creating a total enterprise solution, IBM SecureWay offerings provide the expertise required to plan, design, implement and operate secure solutions for businesses. For more information about IBM SecureWay offerings, visit the IBM SecureWay home page:

  `http://www.ibm.com/secureway`
- **Service offerings**
  Installing new hardware or software can ultimately enhance your efficiency and business operations. But it also poses the threat of business disruption and downtime, and it can tax your valuable internal resources. IBM Global Services provides services that relate to iSeries security. The following web site allows you to search complete listings of services for your iSeries:

  `http://www.as.ibm.com/asus`

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

© Copyright IBM Corp. 1996, 2002

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator
3605 Highway 52 N
Rochester, MN 55901-7829
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy,

modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

ADSM/400
Advanced 36
Advanced Peer-to-Peer Networking
AIX
AnyNet
Application System/400
APPN
AS/400
AS/400e
Client Access
CT
DATABASE 2
Distributed Relational Database Architecture
DRDA
Global Network
IBM
iSeries
iSeries Access for Windows
iSeries Navigator
iSeries server
Net.Data
OfficeVision
Operating System/400
OS/2
OS/400
PowerPC AS
QMF
SAA
SecureWay
SmoothStart
System/36
System/38
S/390
Ultimedia
400

Lotus® and Lotus Notes™ are registered trademarks of Lotus Development Corporation. Domino™ and Notes are trademarks of Lotus Development Corporation.

C-bus™ is a trademark of Corollary, Inc.

Microsoft, Windows, Windows NT, and the Windows 95 logo are registered trademarks of Microsoft Corporation.

Java and HotJava™ are trademarks of Sun Microsystems, Inc.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

PC Direct™ is a registered trademark of Ziff Communications Company and is used by IBM Corporation under license.

Other company, product, and service names may be trademarks or service marks of others.

Index

Special Characters

Numerics

A

© Copyright IBM Corp. 1996, 2002

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Readers' Comments — We'd Like to Hear from You

iSeries
Tips and Tools
for Securing Your iSeries
Version 5

Publication No. SC41-5300-06

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

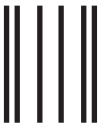
Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SC41-5300-06

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM CORPORATION
ATTN DEPT 542 IDCLERK
3605 Highway 52 N
ROCHESTER MN 55901-7829

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.

SC41-5300-06