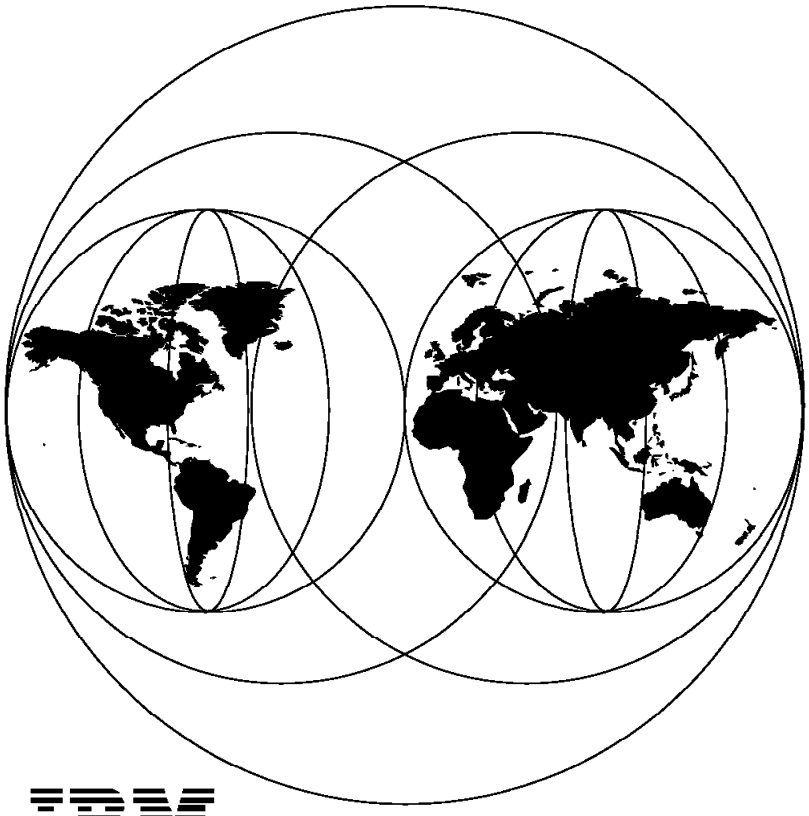
SG24-4929-00

AS/400 Internet Security:
Securing Your AS/400 from HARM in the Internet

June 1997



IBM

International Technical Support Organization
Rochester Center

International Technical Support Organization

SG24-4929-00

AS/400 Internet Security: Securing Your AS/400 from HARM in the Internet

June 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special Notices” on page 245.

First Edition (June 1997)

This edition applies to the IBM TCP/IP Connectivity Utilities for AS/400 (Program 5763-TC1, V3R2M0 or 5716-TC1 V3R7M0); IBM Operating System/400 (Program 5763-SS1 V3R2M0 or 5716-SS1 V3R7M0).

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829

Contents

Preface

Many companies are thinking of connecting their internal corporate networks to the Internet. And for good reasons. There are many rewards associated with both increased visibility and the opportunity to exchange e-mail with the rest of the world or access the wealth of information available on ″the net″.

At the same time, companies are concerned with the security of their systems.

In this book, we take a layered approach to securing your AS/400 system when attaching it to the Internet. We focus on system and application security. We describe the security issues and risks associated with each TCP/IP application and provide examples, recommendations, tips and, techniques that will help the webmaster or system administrator to make an educated decision when implementing those applications in an AS/400 attached to the Internet.

The purpose of this redbook is **not** to cover network security. However, network security is a key component of Internet security and we provide some elements that will help you to evaluate the need for a firewall.

The intended audience for this redbook is technical professionals in the IBM service and consulting community as well as those in the business partner and customer community, especially webmasters and system administrators.

Basic knowledge of each TCP/IP application available on AS/400 is assumed.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Rochester Center.

Marcela Adan is a Senior Technical Support Specialist with the International Technical Support Organization, Rochester Center. Marcela writes extensively and teaches IBM classes worldwide on all areas of system management and AS/400 Internet technologies. Marcela has held several positions as field technical support representative, network administrator, developer and consultant.

Chris Green is a consultant with ASTECH Solutions Inc. in Canada. He is a frequent speaker at AS/400 conferences.

Stephan Imhof is a Senior Systems Engineer in Switzerland. He has 15 years of experience in networking arena. He has worked at IBM for 24 years. His expertise includes most areas of AS/400 networking.

Wayne Markel is a network specialist from the United States. He has worked at IBM for 27 years. His areas of expertise include network design and analysis.

Sverre Selbach is an advisory system engineer in IBM Norway. He has 30 years of experience in data processing and has worked at IBM for 21 years. His areas of expertise include communications, security, and availability. He has written extensively on AS/400 security and auditing and availability recovery.

Silvio Simionatto is an AS/400 specialist in Brazil. He holds a Bachelor's degree from the São Carlos Federal University. He has worked at IBM for 4 years supporting AS/400 customers. His areas of expertise include system management, communications, and Client Access/400.

Brian R. Smith is an Advisory International Technical Support Specialist with the International Technical Support Organization, Rochester Center. He writes extensively and teaches IBM classes worldwide on all areas of AS/400 communications specializing in TCP/IP. Brian spent his first 13 years with IBM Rochester; first with the System/38 and later with the design, development test, and support of the AS/400 system. Currently, Brian is a conference program manager at the International Education Center in La Hulpe, Belgium.

Claus Ziemann is a Technical Support Specialist in Germany. He has worked at IBM for 23 years. His areas of expertise is AS/400 TCP/IP Communications and Internet. He is a frequent presenter in conferences and teaches several workshops on AS/400 Communications and Internet connection.

Thanks to the following people for their invaluable contributions to this project:

Robert Macgregor
International Technical Support Organization, Raleigh Center

Pat Botz
Terry Hennessy
Lynn McEwen
Mark McKelvey
Don Morrison
George Romano
Carol Woodbury
IBM Rochester Laboratory

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 255 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Home Pages at the following URLs:

  For Internet users `http://www.redbooks.ibm.com`
  For IBM Intranet users `http://w3.itso.ibm.com/redbooks`

- Send us a note at the following address:

  `redbook@vnet.ibm.com`

Chapter 1. Internet Security Overview

Many companies are thinking of connecting their internal corporate networks to the Internet. And for good reasons. There are many rewards associated with both increased visibility and the opportunity to run new types of applications.

At the same time, companies are concerned with the security of their systems.

The Internet is a collection of connected networks, but nobody really knows the structure of the Internet. The Internet keeps changing all of the time. There is no centralized network management and no single authority is in charge.

All data crossing the Internet is passed in the clear such as user names, passwords, and e-mail messages. The entire company is exposed to the outside.

The AS/400® integrated security is extremely difficult to circumvent compared to security offerings on other systems. But, this fact should not make AS/400 owners more relaxed when attaching their systems to the Internet. No matter how good the AS/400 security features are, they cannot help if system security is not configured properly and monitored regularly.

In this book, we take a layered approach to securing your AS/400 system when attaching it to the Internet. We focus on system and application security. We strongly recommend not connecting the system to the Internet until you are 100% sure that you have thoroughly reviewed OS/400 security and that the TCP/IP applications you have chosen to use across the Internet are properly and securely configured.

The purpose of this redbook is **not** to cover network security. However, network security is a key component of Internet security and we provide in this chapter some elements that will help you to evaluate the need for a firewall.

This chapter provides a general overview of the security issues and risks when connecting to the Internet and the technologies available to cope with those security challenges.

1.1 AS/400 System as Internet Server

The functions and features to run an AS/400 system as an Internet server are included in the following licensed programs at no additional charge:

- 5763-SS1, OS/400 V3R2 for CISC
- 5716-SS1, OS/400 V3R7 for RISC AS/400 systems
- The TCP/IP Connectivity Utilities/400 (licensed program numbers):
  - 5763-TC1 for CISC models
  - 5715-TC1 for RISC models

The World Wide Web (WWW) HTTP server is the center piece of the Internet functions. The HTTP server delivers HTML documents to the browsers. It allows access to CGI programs as well as to the DB2®/400 database. The Workstation Gateway (WSG) server accessed through the HTTP server transforms a 5250 data stream into HTML dynamically. This enables running AS/400 applications from any Web browser.

The OS/400 HTTP server is implemented based on CERN (Conseil Europeen pour la Recherche Nucleaire or European organization for Nuclear research) specifications.

Web Server/400, Commerce Server/400, and Webulator/400 from I/Net Inc are other Web server products available for the AS/400 system. I/Net's Web Server/400 includes an HTTP server following NCSA (National Center for Super Computing Applications at the University of Illinois at Urbana-Champaign) specifications. Functionally, I/Net's Web server is almost equivalent to the HTTP server included with the Internet Connection for AS/400.

I/NET's Commerce Server/400 includes encryption technologies based upon the SSL standard for the purpose of conducting secure commerce across the Internet.

Webulator/400 is a 5250-to-HTML gateway product that is similar to IBM's 5250 Workstation Gateway (WSG).

The purpose of this document is to discuss security considerations when attaching an AS/400 system to the Internet. It includes the Internet Connection for AS/400 as well as I/NET's Commerce Server/400 and Webulator/400, but does not include I/NET's Web Server/400 since, from a security standpoint, it does not differ considerably from the IBM HTTP server discussed in Chapter 3, "Securing Your First Application - HTTP Server" on page 57.

Let's have a first look at the Internet functions offered by both set of products and its security features. These capabilities are discussed in-depth in the following chapters.

Application	Internet Connection for AS/400	Commerce Server/400	Webulator /400	Anonymous Service	UserID/ Password Required	Logon Exit Program	Activity Logging
Telnet Server	yes	-		-	yes	-	yes 1)
FTP Server	yes	-		yes	yes	yes	yes 2)
SMTP Server	yes	-		-	-	-	
POP3 Server	yes	-		-	yes	-	
LPD	yes	-		--		-	-
HTTP	yes	yes		yes		-	yes
- Authentication	-	yes		-	yes		yes
- SSL Encryption	-	yes		yes	yes	-	yes
5250 to HTML	yes		yes	yes	yes	yes	yes
- SSL Encryption	-		yes	yes	yes	yes	yes

1) See history log for start and termination of interactive job
2) Server request exit program programmed to log activities

Figure 1. Internet Connection for AS/400 and I/NET Security Functions

Commerce Server/400's HTTP server allows host (IP address, domain name) and user (user ID, password) authentication.

Anonymous service means that the server offers a limited set of information to non-identified users and an extended set of information to identified users. This differentiation cannot be made with SMTP and LPD.

1.2 Threats - Is the AS/400 System Secure Enough?

The number of servers connecting to the Internet is growing at a phenomenal rate going from a few thousand systems in 1994 to over 252 000 in June 1996. It is estimated that there are over 50 million Web pages on the Internet. This explosive growth of the Internet makes it an increasingly attractive way of doing business. However, this vast quantity of information also makes it an attractive environment for hackers.

Many systems are attacked. The Computer Security Institute released a survey of corporate security specialists in May 1996 where 42% of them indicated they had knowledge of unauthorized use of their systems in the preceding year. It is to be assumed that the remaining 58% of the companies neither had or detected any attack.

A publized break-in could be very damaging to your company's reputation. Do you want to do business with a bank that has reported a system break-in?

Certainly large and well-known companies are a favorite target for intruders. Being a small company should be of no consolation. Intruders know that small businesses are less likely to be Internet savvy. Tools widely available on the Internet enable new systems to be discovered quickly.

The threat is serious and no company should attach a system to the Internet without understanding the risks involved.

Most of the Internet servers run on UNIX® based systems. It is widely accepted that the AS/400 system has strong security features. Do these capabilities make the AS/400 system immune from attacks from the Internet? Unfortunately, NO. Networks in general and TCP/IP protocols and applications used by the Internet in particular have inherent security issues that are problematic to all kinds of systems.

Technical insufficiency is one source that makes an Internet server vulnerable. However, bear in mind that another source of threats is human errors such as incorrect configuration or careless handling of passwords. These risks are independent from any hardware platform.

1.2.1 Internet Security Exposures

Let's talk about some of the ingenious ways in which hackers attack your system. Figure 2 on page 4 provides an overview of some well-known attacks.

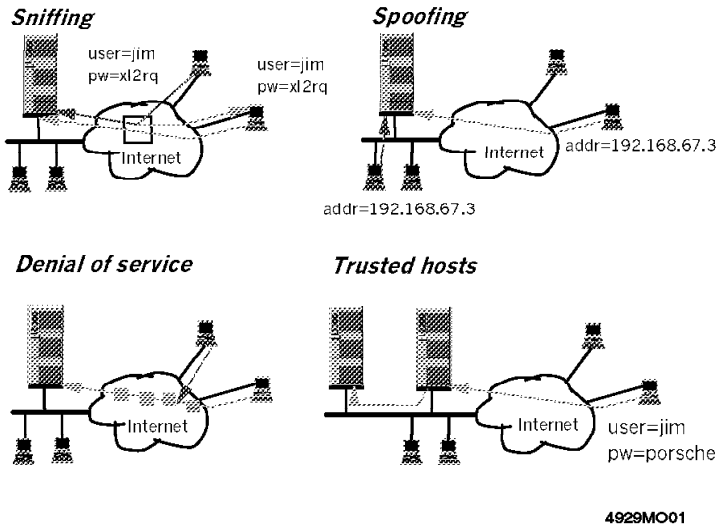


Figure 2. Examples of Internet Security Exposures

1.2.1.1 Sniffing

Many of the most popular Internet applications such as Telnet and FTP send their passwords in the clear. Any router or intermediate system that handles your IP traffic has the opportunity to read and copy these user IDs and passwords. This is called sniffing. There is no need for highly sophisticated trace analyzer tools to accomplish sniffing.

Because the Internet is not centrally controlled and is made up of a number of commercial enterprises, ISPs, government agencies, and educational institutes, there is no way to know the route your data takes and who might be looking at it.

Encryption schemes that encrypt a password using the same key more than once are also unsafe because a hacker can simply reuse the password in its encrypted form.

For the same reason, you do not want to send e-mail with your credit card number or other sensitive data over the Internet.

1.2.1.2 Spoofing

You cannot trust that anyone is who they say they are. One can easily configure a workstation with a different IP address than the one assigned to it. A hacker can send a request that appears to come from your CEO′s workstation. This is called spoofing.

IP subnetting and routing does not easily allow an external hacker to pose as an internal system. However, with the IP source routing function, routing tables can be subverted. The IP source routing field in an IP packet tells the receiver to override the normal route to the originator of a request.

How does someone know the address of your CEO′s workstation? There are many useful tools to discover host names and addresses and the structure of

your network. Tools such as PING, TRACEROUTE, and Domain Name Services enable a hacker to discover your network.

1.2.1.3 Denial of Service

Vandalism has been with us for centuries. Some people like to trash mailboxes or slash other people's car tires; others prefer to crash computer systems. This is also a chance for a malicious competitor to prevent you from advertising your products or providing services over the Internet.

The inability to distinguish good from evil makes it difficult to stop this kind of attack. Is the mail message you are about to receive from an important client or from someone who tries to drive you mad? You cannot tell until you have read the message. Multiply by thousands of false mail messages and you have a denial-of-service attack.

There are many more ways to mount attacks with the objective of preventing your operation by re-directing traffic or bombarding your AS/400 system with junk. Once a hacker succeeds in signing on to your AS/400 system, the damage to your operation can be fatal.

1.2.1.4 Trusted Hosts

Usually a dedicated system is chosen to implement an Internet server. TCP/IP definitions are set on this machine in a way that an external user may only reach the Internet server but not any productive system behind it. The danger is that there is still a way to get from the Internet server into the rest of your company's network since the configuration was not done accurately enough. For example, starting a 5250 pass-through session from a WSG session might still be possible.

1.3 How to Control the Risk?

To avoid any risk, your company might decide not to have the AS/400 system attached to the Internet: *No Internet - No risk!* There are ways of dealing with the Internet without being attached directly:

- Using the services of an ISP (Internet Service Provider). The ISP stores, serves, and maintains your Web pages.
- Using the Internet e-mail gateway of a value-added network provider such as IBM Global Network™ (IGN). Your AS/400 system is connected through an SNA link to IGN and uses SNA protocols to forward and receive e-mail through IGN's Internet e-mail gateway.

Can you avoid any risk by attaching an **isolated AS/400 system** to the Internet? What if someone is able to break in and alter your Web pages? If this is publicly known, the image of your company suffers anyway.

This makes it clear that there is always a risk with being attached to the Internet. However, the benefits for your company being present in the Internet are many. But it is a high-level management decision whether and how to deal with the Internet and to consider the risks. These policies are part of the overall I/T and networking policies and strategies.

1.3.1 Your Company′s Internet Security Policy

There is a fundamental conflict between access and security. With each new connection or new TCP/IP application you provide for your users or customers, you have given a potential intruder one more means to try to enter your system. So before you connect to the Internet, you need to make explicit decisions on what type of access and services are offered.

One of the following statements can be your general Internet security policy:

- No Internet access
- Anonymous-only access from Internet
- Access by user ID and password required

Anonymous access does not require a user ID and password being entered by the user. This does not mean that the user gets unlimited access. The anonymous user gets limited access to a well-defined small portion of your AS/400 system. Anonymous access can be accomplished with FTP, HTTP, and WSG.

Your Internet application may require the identification of the user. This means the user authorization is accomplished with **user ID and password**. You run a high risk when allowing passwords to flow in the clear. Passwords are transmitted more safely when encrypted. This can be done using Commerce Server/400 and Webulator/400 for HTTP and WSG.

Your Internet security policy should be in the context of your larger I/T security policy. It needs to be determined before you look at individual solutions and it has to be decided by higher-level management. It should answer the following questions:

- What are your Internet applications?
- How do the users access your Internet server?
- What are the security objectives to be achieved?
- What are the security measures you are going to implement?
- Is there still a risk? And if yes, what is the probability and the damage involved?

1.4 What Should You Secure?

When you devise your security measures, you should think of a layer approach to security. When you connect your AS/400 system to the Internet, there are many points where security is compromised and, therefore, that you should protect. You should think of this layer approach as a system with multiple locks; if a hacker manages to break one of them, you have others to protect you.

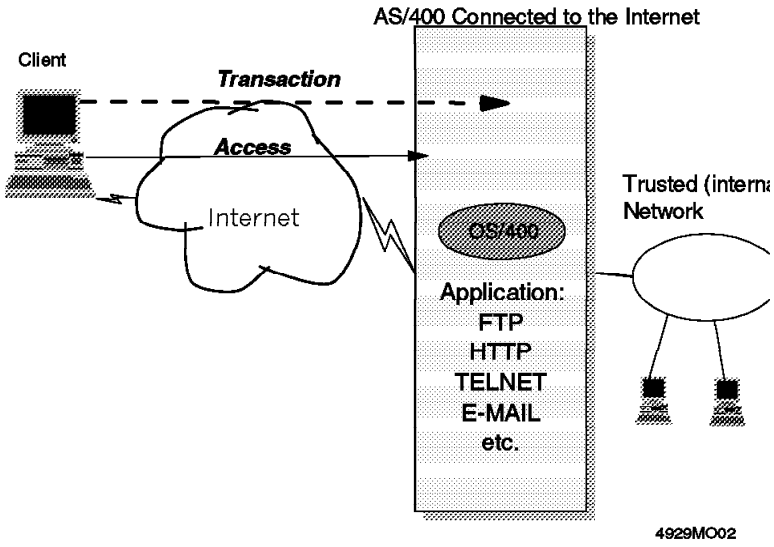


Figure 3. Layer Approach to Security

Figure 3 shows different areas where you should apply security measures:

- **Network security:** Controlling access to your AS/400 system.
- **Application security:** Application-specific security. Do you want to enable a particular application such as FTP or TELNET? Do you want to enable only anonymous users or do you want to require user ID and password?
- **Transaction security:** Ensuring data privacy and partners authentication.
- **System security:** OS/400 offers many features and functions that, used properly, can make your AS/400 system a secure system.

This section introduces the layered approach to Internet security. In the rest of the book, we focus on OS/400 and application security. This redbook does not cover network security.

1.4.1 System Security

The AS/400 system offers a strong set of security tools, but you must take the time to learn about the tools and apply them.

There are various areas of the AS/400 system's security to be considered before attaching your system to the Internet:

- System-wide security values
- User profile and password management
- Resource security
- General TCP/IP definitions

For more information, we suggest reading Chapter 2, "Start Here by Securing OS/400®" on page 37.

1.4.2 Application Security

Each application that you can use on your AS/400 system connected to the Internet such as HTTP, FTP, TELNET, and so on offer different alternatives to limit access and make it safe to use.

A major part of this document discusses some of the most common applications and provides recommendations to make its usage more secure.

1.4.3 Transaction Security

Commercial transactions through the Internet require safe communications. The parties need to be identified and exchanged data has to be protected.

- How can you perform authentication without sending a user ID and password in the clear?
- How can you protect the privacy of your data to ensure that only authorized persons may read it?
- How can you assure that messages have not been altered between the sender and the recipient?

There is a single technology that provides the foundation for solving all of these challenges called cryptography. SSL (Secure Sockets Layer) is an industry-standard providing cryptography. It includes encryption, message integrity verification, and authentication.

Commerce Server/400, together with secure Internet browsers, provides transaction security based SSL. Refer to the *WWW.Security* book for more information on transaction security.

1.4.4 Network Security

Network security controls access to your AS/400 system. Who is allowed to enter your corporation's network to access your Internet server? Probably you do not want to generally limit the access but it is a major issue to protect your internal network and the productive systems within your company's internal network.

Network security can be achieved in various ways:

- Isolating the AS/400 system as an Internet server.
- Multiprotocol router blocking from non-wanted TCP/IP traffic.
- Securing the network gateway (usually called a firewall) to protect the company-internal network.

Internet network security also determines how your own users may access the Internet.

For more information, refer to Section 1.6, "Network Security Scenarios" on page 11. We do not cover network security in-depth in this redbook.

1.5 Internet Security Principles

These general rules and guidelines help you to design and to implement security features and functions with each of your Internet applications.

You can include these principles in your Internet security policy. However, they are more of a general working instruction rather than something to be decided by high-level management such as your Internet security policy.

Areas of principles are:

- Responsibilities
- General implementation rules
- Implementation steps
- Considerations with each application

Information security, unlike network protocols, does not have an agreed-upon set of standards. Within organizations, however, standards and practices evolve in support of that organization′s information security policy. In a few cases, those standards and practices have received widespread support as a basis for other organizations.

The most obvious example of this is the U.S. Department of Defense *″Orange Book″*. It defines information protection standards for the Defense department. For networks, the redbook interprets the *Orange Book′s* criteria. One level of the Orange Book criteria, C2, has become a de facto standard for commercial information security.

A second set of information security principles is evolving under the auspices of the *Information Systems Security Association (ISSA)*. This standard is in the spirit of the *Generally Accepted System Security Principles*.

RFC 1244 - The Site Security Handbook is the product of the Site Security Policy Handbook Working Group, which is an effort of the Internet Engineering Task Force (IETF). It is a guide to setting computer security policies and procedures for sites that have systems on the Internet. Again, it is not a standard. It makes recommendations and gives discussions of relevant areas.

These documents show:

- Internet security is not an isolated subject; it is part of the larger networking and I/T security.
- Experience from large organizations and corporations is documented and it can help you to improve your task.

You should refer to the general available Internet security documentation to help you design your policies and procedures.

1.5.1 Responsibilities

After the security policies and procedures have been decided, it is a good idea to have a single person or team be responsible for the implementation while being periodically audited by objective parties. Can the webmaster be responsible for the security of the entire Internet environment? A webmaster usually maintains the Web server pages and configuration. However, it depends on the webmaster′s skills and resources whether the webmaster is in a position to be responsible for the Internet security as well.

1.5.2 General Internet Security Principles

- **Simplicity:** You are probably starting to find that Internet security can be quite complicated. Since Internet security can involve lots of complex configurations, there is the opportunity for introducing errors that can be exploited by a hacker. As a matter of fact, configuration holes are one of the most common means of intrusion. The simpler your configuration, the more likely it is to be correct.
- **Explicit authority:** This is a mind-set. Your defaults should be set up to deny access. Only the specific users you authorize should be able to perform functions. Everything else should be denied.
- **Chokepoints:** Limiting the number of connections or routes data can take allows you to concentrate on your defenses. It makes it easier to control and monitor. This chokepoint may be physical or logical.
- **Secondary defense:** Do not assume your defenses always work. You can make configuration errors or hackers can get past one of your defenses, but if you have another roadblock in place, it either slows them down or stops them completely. Developing a healthy paranoia helps you to do a good job.
- **Do not trust:** Do not trust any information you receive from the Internet such as IP addresses, host names, or passwords. These can be forged.

1.5.3 Steps to Implement Secure Internet Applications

- **Design for Security:** Based on policies decided by observing your company′s general I/T and networking security directions. For later testing, auditing, and extension, document the security measures you decided to implement.
- **Test:** Do not assume that all of the security features you implemented are running properly. Test them! And test them on a regular basis. Anytime you make a change in a configuration, you want to verify that you have not inadvertently opened a security hole.

  Engage a neutral or company-external person to test the security measures of your Internet environment.

  There are utilities available, mostly UNIX-based, to test Internet security. These programs check mainly the network access.
- **Control:** Logging the activities provides information on the usage of your Internet applications. Develop queries to analyze this data and to find possible attacks and misuse.

  PC based utilities are available to analyze and present the result graphically.

  Check for attacks that can be detected and for attacks where appropriate action can be taken immediately. For example, an attempt to use a non-existing user ID should result at least in a message to the QSYSOPR message queue, generation of an SNA alert, or an SNMP trap or transmission of a Pager message.
- **User Education:** You cannot assure security alone. You need to make sure that your users are helping. All of the complex security features in the world are not going to help you if users share their passwords in e-mail messages. Users must be educated on the risks associated with the Internet and be given clear instructions on what they should and should not do.
- **Revision:** Time changes things. Technology is getting more advanced, Internet applications are enhanced, and hackers are getting smarter. Consequently, your security measures need to be revised periodically.

1.5.4 General Considerations for Applications Available Through Internet

Before making an application or service available through the Internet, you should consider the following points:

- **Responsibility**
  - Who is responsible for this application? Who is suppose to implement security for this application? Who should audit this application′s operation?
- **Access Control**
  - Which information is this application allowed to access?
  - How is the information accessed: Read only, update, add, or delete?
  - How is the information access limited?
  - How is the user access to be controlled?
  - Does this application provide an anonymous service or is the user to provide a user ID and password?
  - Does the password flow in the clear or is it encrypted?
- **Risk Assessment**
  - What are the security exposures?
- **Audit and Control**
  - How can your security measures be tested? After you implemented it, after changes, periodically, or by a daily program?
  - How are the activities of your Internet Server logged? How can the collected data be analyzed? Are queries or tools available to analyze the collected data?
  - Are there ways to continuously monitor and report attacks?

1.6 Network Security Scenarios

There are basically three different ways of interconnecting the AS/400 system as an Internet server:

- The isolated Internet server
- The integrated Internet server
- The intranet server

It is agreed that many flavors of these three types exist. However, these three cases allow us to discuss specific issues with each of the different environments.

- The **fully-isolated Internet server** does not represent any danger to the company′s internal network and production systems.
- The **integrated Internet server** is a high risk if you do not carefully plan your security strategy to protect your production systems.
- The **intranet server** addresses a known set of users and the involved risk is comparable to any other network application.

It might be a good practice as well to start off with an isolated Internet server providing Web pages only to gain first experience with minimal risk.

1.6.1 The Isolated Server

Figure 4 shows the AS/400 system as an isolated Internet server. We minimized the security exposures by not connecting the public server to the internal network.

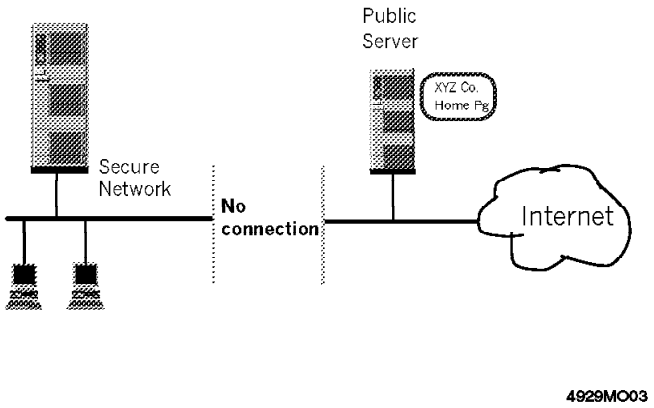


Figure 4. Connecting an Isolated System to the Internet

The AS/400 system is directly connected to the ISP (Internet Service Provider). In most cases, the link to the ISP is accomplished with a multi-protocol router. The AS/400 system can be connected with this router through an isolated LAN segment.

It is essential that there is no permanent link between the Internet server and the productive AS/400 systems (except for maintenance) while the Internet server is active.

A break-in into an isolated AS/400 system as an Internet server does not have an impact on your company′s internal network. However, a hacker still can disturb your Internet server through a denial-of-service type of attack.

If, for whatever reason, a link between the Internet server and any production system is established, it represents a risk to be investigated. Implementing this interconnection, your Internet server immediately becomes an integrated server, which is discussed in the next section.

1.6.2 The Integrated Server

Is it getting too lonely out there? If, after sometime, you start complaining about the disadvantages of having an isolated system attached to the Internet:

- I want my users to be able to exchange mail with Internet users.
- My users want to be able to use a browser to access information from the Internet.
- I need to update the data on my Web server frequently and I want to serve dynamic content and perform business transactions.

It may be time to start thinking about connecting your AS/400 Internet server to your internal network. Figure 5 on page 13 shows how the public server and the internal network can be connected.

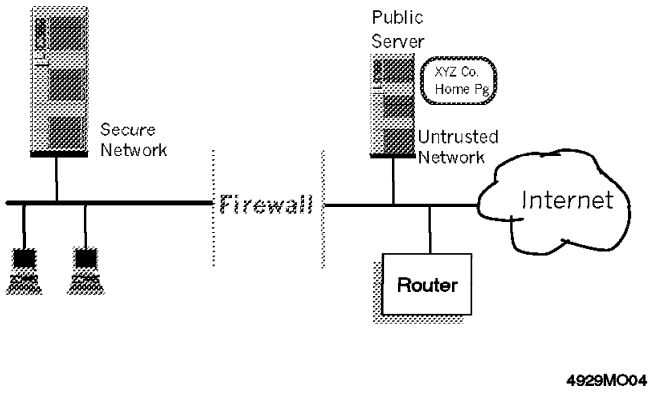


Figure 5. AS/400 System as Integrated Internet Server

If you are finding the isolated system too limiting, it is time to look at your security and access policies. You probably need a more complex security solution that enables you to have your public Web server and enable internal users to have the access they need. Installing a firewall to protect your internal network is probably the best solution since it offers the most flexibility and protection.

We do not recommend that you connect your internal network directly to the Internet because the AS/400 system, just the same as any other system, cannot distinguish the good from evil.

1.6.3 The Intranet Server

You can use the functions provided by IC/400 to service your internal users that access your server through local and remote LANs or SLIP connections. In this case, you are using the same applications but you are not connecting your system to the Internet and, therefore, the security considerations are the same that apply to any privately owned network.

1.7 Internet Firewalls

Firewalls provide a means of protecting your internal corporate network from unauthorized access from the Internet. They are just one of the tools for defense that can be employed.

A firewall is used to help implement your Internet security policy. The firewall provides a barrier between a secure network and unsecured network such as the Internet. The firewall controls access to and from the secure network.

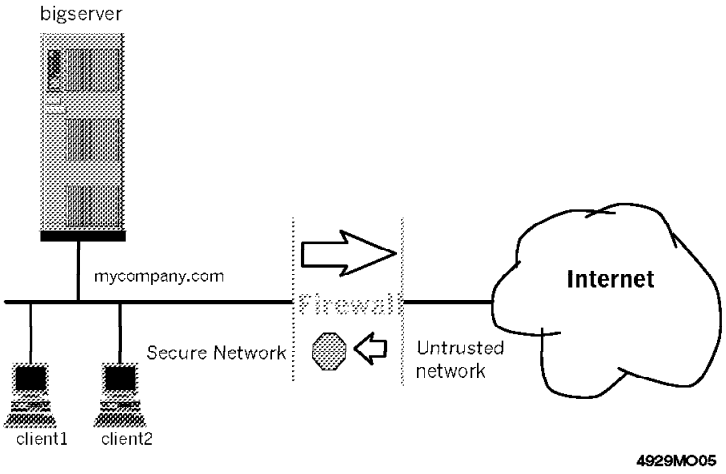


Figure 6. Protecting Your Internal Network with an Internet Firewall

Things a firewall can do:

- Let the internal users access Web servers on the Internet.
- Let the users exchange mail with other users on the Internet.
- Prevent users on the Internet from accessing systems in your corporate network.
- Prevent information about your network (for instance, IP addresses) from being exposed to the users on the Internet.

Things a firewall cannot do:

- A firewall is able to protect from intrusion from the outside. A firewall does not protect you from an inside user sending sensitive information over the Internet.
- A firewall does not provide protection of data that is sent from an internal user to an Internet user.
- Most firewalls are not able to check for viruses.

1.7.1 Why are Firewalls Needed?

There are potential intruders on the Internet. These intruders attempt to exploit the known weaknesses in the IP, TCP, and ICMP protocols and the applications that use them.

Many people believe that since the AS/400 system has strong host security, it can be directly connected to the Internet. Unfortunately, this is not true because the AS/400 system has to contend with the same unsecured TCP/IP protocols as other systems.

It is not just the AS/400 system that you need to protect. Once you connect to the Internet, every system of your internal network is accessible from the Internet.

Firewalls are needed so that a security exposure on any of the systems in your internal network cannot be exploited by users on the Internet.

1.7.2 Firewall Principles

When setting up a firewall, there are a number of principles that you are advised to follow. Some are:

- Make sure that you do not have any other connections to the Internet. The firewall provides a chokepoint, forcing all traffic to and from the Internet to flow through it.
- There should be no direct TCP/IP connections between the applications on the internal systems and the servers on the Internet. A direct connection enables the server to learn information (such as the IP address) about the client system. All communication connections should be broken at the firewall.
- Information about the internal network should be prevented from reaching the Internet. Information on host names and IP addresses is valuable.
- Systems that are intended to be accessed by users on the Internet should be on the outside of the firewall. Once you start letting Internet traffic through the firewall, you open new holes for an intruder.

1.7.3 Firewall Elements

Some people assume that a firewall is a single box with one wire in and one wire out. This is not always the case. A firewall is constructed from one or more software products that run on one or more hosts that may be general purpose systems or routers.

Major technologies implemented with firewalls are:

- Packet filtering to limit traffic.
- Proxy servers or SOCKS servers to break TCP/IP connections.
- Domain name services to hide network information.

Policy plays an important role because the various technologies can be used in many ways. It is important that a company decides on its Internet security policy before it begins the process of building a firewall.

1.7.3.1 IP Packet Filtering

IP packet filtering is a technology inserted at a low level in the IP protocol stack. A packet filter compares the packet against a set of rules that say which packets are permitted (this means which packets have to be forwarded or discarded).

Packet filters are a good way to selectively allow some traffic into a subnetwork to protect from unwanted traffic. A packet filter is completely transparent to the user.

Packet filters check the packet header to determine whether to forward or to discard the packet. Most packet filters allow filtering by:

- Source and destination IP address
- Protocols such as TCP, UDP, or ICMP
- Source and destination ports (ports identify a TCP/IP application such as FTP or Telnet.)
- Whether the packet is destined for or originated from a local application
- Whether the packet is inbound or outbound

Your initial thought might be that this is going to be real easy. But we have to make a distinction between inbound/outbound packets and inbound/outbound connections. Inbound packets resulting from an outbound connection are OK. That means packet filters need to pay attention to the flags in the TCP header (SYN or ACK) that indicate if this is a new connection or a response to an existing connection.

A typical installation has 50 to 100 of these rules. They usually come in sets that allow a particular application to run between a set of IP addresses. And at the end, there is a rule that says to deny all other traffic. This is an implementation of one of the Internet security principles: *That which is not expressly permitted is denied.*

1.7.3.2 Packet Filtering Router

Most popular routers have some sort of packet filtering technology. Although by themselves they are not really a firewall, they may provide enough protection in some circumstances.

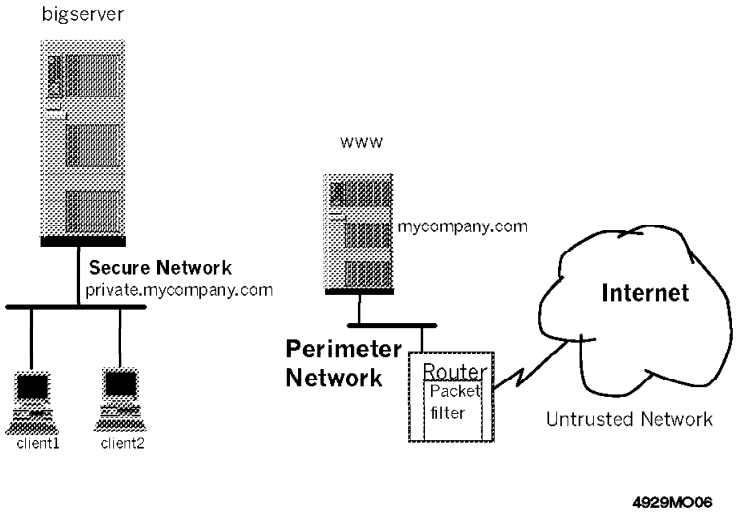


Figure 7. Packet Filtering Router

Let's take the situation where you want to attach the AS/400 system as a Web server to the Internet. This server is a public server, which means you want users on the Internet to be able to easily find it. You want to provide some protection for this server but you cannot isolate it. Using packet filtering support on the router is probably all you need. You can set up your rules to allow HTTP requests in and HTTP requests out but block unwanted traffic such as Telnet and FTP.

Notice the network is broken into two pieces. The internal or secure network has all internal users and production machines. It is kept separate from the "perimeter network", which has your server intended to be accessed from the Internet. We keep these two networks unlinked because a router alone cannot provide enough protection for your internal systems.

This network scenario with an isolated Internet server is a cheap solution since you need a router anyway to connect to the ISP. But this solution has some limitations:

- There is no logging of packets discarded by the router.
- It is hard to keep the isolated system current since it cannot be reached from the internal network.
- Internet applications cannot work with your productive database.

1.7.3.3 Proxy Server

A proxy server is a TCP or UDP application. Its purpose is to receive requests from a client and resend them to a server and to resend responses from the server back to the client.

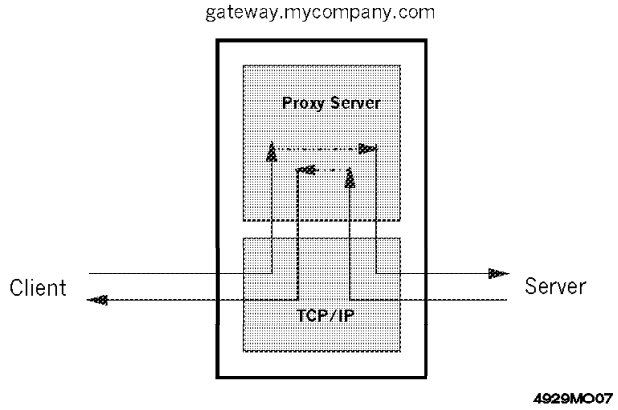


Figure 8. Proxy Server

Proxy servers are unique to the particular protocol that they handle (for instance, an HTTP proxy or a Telnet proxy).

The most important objective of a proxy server is to break the TCP/IP connection. Clients no longer talk directly to servers. The server only sees the IP address of the proxy server, not of the originating client. This is useful to keep the internal network information private.

The clients need to know the address of the proxy server to send the request to the proxy instead of the server it wants to communicate with. This means the client application needs to be proxy-aware, which means specific definitions are required. The servers, on the other hand, are standard. They have no knowledge that a proxy server is being used.

One of the bad things about proxy servers is that they are unique to a particular application. If you obtain a new TCP/IP application, you may have a difficult time finding a proxy server to support it.

Probably the most common example of a proxy server is the **HTTP proxy server**. An HTTP proxy server relays requests from a Web browser to a Web server. The client's browser is configured to send requests for URLs to the proxy server instead of the server.

Not all proxy servers are quite so easy to use. A Telnet proxy server, for example, may require the users to Telnet to the proxy server, to log on, and to Telnet again to the system that they want to communicate with. The IP address of the proxy server is used as the source address, hiding the IP address of the AS/400 system.

Another common proxy is one that relays mail between internal mail servers and other mail servers on the Internet. Because the **mail proxy server** simply forwards mail, sometimes it is called a **mail relay**. The mail proxy server relays all incoming mail to an internal mail server where it can be accessed by the internal users. All outgoing mail is also routed through the mail proxy server.

Mail proxy servers use SMTP. The workstations, when communicating with the internal mail server, communicate through POP.

1.7.3.4 SOCKS Server

Sockets server, SOCKS for short, is another TCP/IP application that resends requests and responses between clients and servers.

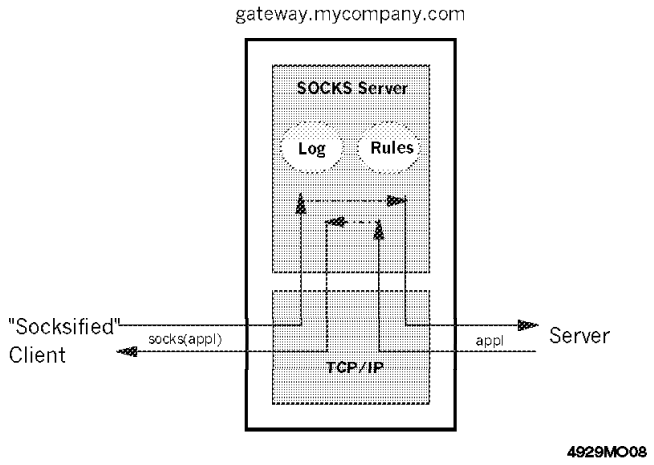


Figure 9. SOCKS Server

The SOCKS server can be thought of as a multi-talent proxy server. Instead of handling one type of application protocol, it handles them all (HTTP, Telnet, FTP, and so on).

The purpose of the SOCKS server is the same as the proxy server; it breaks the TCP/IP connection and hides internal network information.

However, to use a SOCKS server, the client must be written to support the SOCKS protocol. Some applications such as Web browsers support SOCKS. There are also some systems such as OS/2® that support SOCKS in their TCP/IP protocol stack so that all client applications can use a SOCKS server.

The client configuration gives the name of the SOCKS server to use and rules for when it should be used.

To avoid the need to have individual proxy servers such as for HTTP, TELNET, and FTP, there is a move to SOCKS servers.

1.7.3.5 Domain Name Services

Domain Name Services is the application that enables a client to determine the IP address of a given host name. Most of the time, we use host names such as ″www.mycompany.com″ when talking about hosts on the Internet. The Domain Name Server (DNS) translates host names into IP addresses.

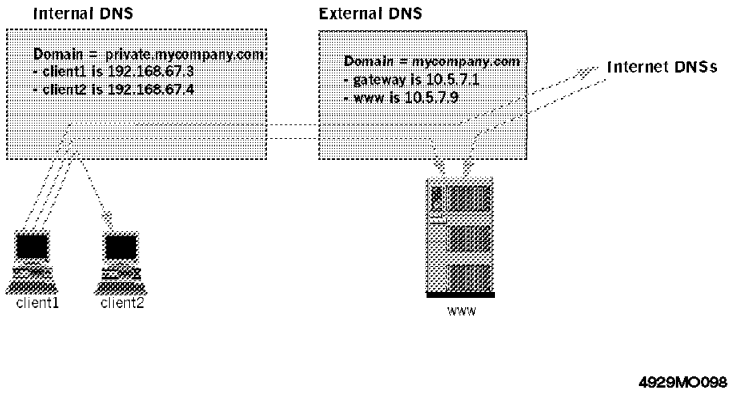


Figure 10. Domain Name Services

When constructing a firewall, we use Domain Name Services so that internal users can locate the IP addresses of all systems, internal and public ones, while users on the Internet can only locate the IP addresses of our Internet servers.

We need two Domain Name Services, one for internal names and one for external names. The internal Domain Name Service is responsible for your internal systems. It forwards name resolution requests to the external Domain Name Service if it does not know the host name. The external Domain Name Service is configured to forward requests to name servers on the Internet if it does not know the host name. This allows internal users to access hosts on the Internet.

Users on the Internet send requests to the external Domain Name Service to locate your Internet server.

Domain Name Service requests only go out. The external Domain Name Service does not forward requests to the internal Domain Name Service.

1.7.4 Firewall Scenarios

1.7.4.1 Dual-Homed Gateway Firewall

The dual-homed gateway firewall is one of the most popular configurations. It is called dual-homed because it uses a host that is connected to two different networks. One connection is to the internal secure network and the other connection is to a perimeter network that has access to the Internet through a router. The perimeter network sometimes is called a demilitarized zone (DMZ).

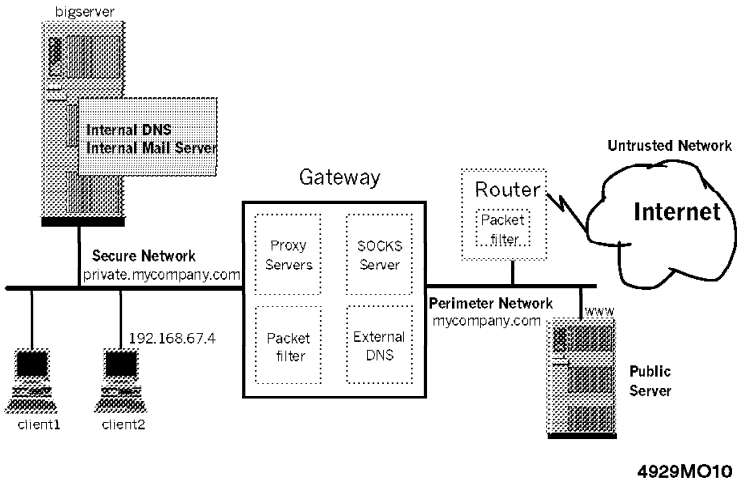


Figure 11. Dual-Homed Gateway Firewall

The firewall host has IP packet forwarding disabled on it. That is, it can send and receive data from local applications but the packets can forwarded only by proxy or SOCKS servers.

The router has packet filtering capability. It is configured to reject undesirable inbound connections. The firewall gateway may also use packet filters to protect itself from unwanted IP traffic. This is a good backup also in case the packet filters on the router are not configured correctly.

By using two domain name servers, the names of the internal hosts are not visible on the Internet. Yet, the internal users have access to all systems, including the Internet server on the perimeter network.

In this environment, the IP addresses used in the internal secure network do not need to be valid IP addresses since they are never seen on the Internet. This allows you to use the private IP addresses ranges.

1.7.4.2 Further Firewall Configurations

The **screened host firewall** is quite similar to the dual-homed gateway with one difference. The separation of the secure network from the perimeter network and the Internet is now logical rather than physical. We fully rely on the packet filtering in the router.

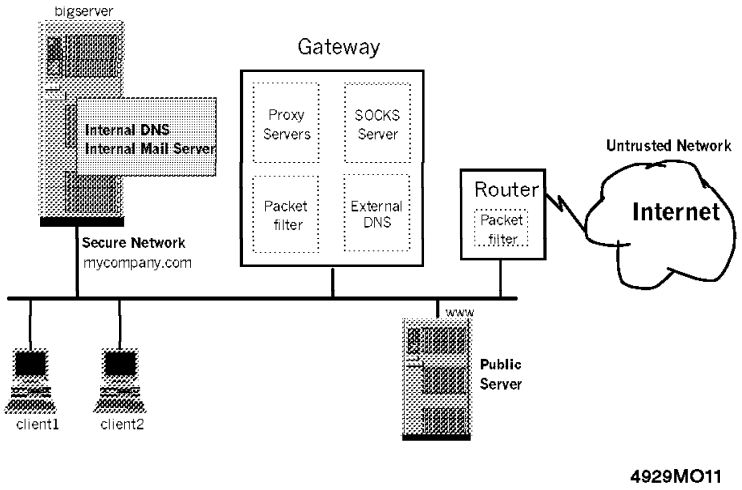


Figure 12. Screened-Host Firewall

The Internet server and the internal systems and users are in the same network. The internal users need to go through the firewall gateway and Internet users may reach your Internet server only.

With this configuration, it is easy to communicate with the internal systems. This makes it possible to update the public server and to get dynamic information from the productive systems.

There are many more Internet connection scenarios. A **bastion** is a simpler but less secure implementation. The Internet server is placed between the secure and the non-secure network. IP forwarding is disabled, which means no traffic can flow through this system. Only users who have a user profile on the bastion can use services in both networks.

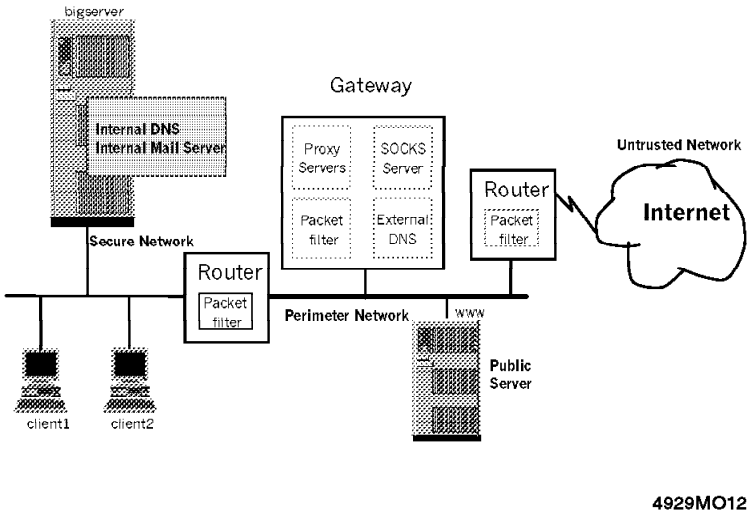


Figure 13. A Bastion Firewall

This scenario does not allow users of the internal network to reach the Internet as long as there is no proxy or SOCKS server available. In addition, good password management has to be enforced not to allow unauthorized signing on.

1.7.5 Firewall Products

The firewall market is growing quickly, fueled by the growth of the Internet itself.

Almost all of the solutions are UNIX-based since the TCP/IP protocol used on the Internet has its origins there.

A firewall can be quite expensive. The price for a firewall, hardware, operating system, and firewall software included runs around tens of thousands dollars. Since security and firewalls are complicated subjects, you might want to engage a consultant to install and configure your firewall.

1.8 Secure Transactions over the World Wide Web

When you place your AS/400 World Wide Web server on the Internet, you are inviting people to come and connect to it; in fact, you would be disappointed if no one visited your Web site. As a public Web server, the AS/400 system can provide company information, product information, customer service, and much more. Your AS/400 public server can provide a storehouse of information that anyone can access using FTP. As we discussed before, even when you might want to make the information on your AS/400 public server generally available and attractive so that more and more people visit your site, you must still protect your system from malicious attacks.

With the advent of electronic commerce, such as online shopping and online bill paying, electronic tax filing, and so on, information privacy becomes necessary. It is not enough to protect your server from malicious attacks; now the *value* of the data you are protecting is different from public data. In addition, intra-company communications over the Internet often include confidential product information that needs to be protected from general access. The

objectives of your security measures depend on the type of data you are exchanging.

Because the Internet was designed to be an open network, it allows any computer attached to it to see the messages passing through. You can think of the Internet as a postal service, sorting and routing messages through various post offices (the systems connected to the network). Unfortunately, anyone and everyone can be a postmaster and there is nothing to prevent a postmaster from reading, copying, and even altering the mail as it passes through. Another drawback is the lack of receiver accountability for items such as registered mail that required signatures or return receipts. As a result, you cannot be sure that the intended recipient received the mail you sent or that the client is receiving the document from you and not someone posing as you.

1.8.1 What Are Your Security Objectives?

Depending on the type of transactions you want to service over the Internet, your security objectives fall into one or more of the following categories:

- Access control
- Privacy or confidentiality
- Integrity
- Accountability
- Authenticity

Access Control: Access control means assurance that the human or machine at the other end of the session is permitted to do what is asked for. This means we want to be able to restrict our server in two ways:

- It should only deliver documents from within certain directories or libraries. This is achieved through server configuration options.
- For certain restricted documents, it should only deliver them to specified users. This requires us to also address authentication because the server must identify the client user in order to decide whether to deliver the document or not.

Privacy or Confidentiality: Privacy or confidentiality means that the messages remain private as they pass through the Internet. The objective is to assure that sensitive information is not visible to an eavesdropper. **Encryption** ensures confidentiality.

Integrity: Integrity means that messages are not altered while being transmitted. It assures that the information that arrives is the same as when it was sent. Any router along the way can insert or delete text or garble the message as it passes by. Without integrity, you have no guarantee that the message you sent matches the message received. *Encryption* and **digital signature** ensure integrity.

Accountability: Accountability assures that any transaction that takes place can subsequently be proved to have taken place. Both the sender and the receiver agree that the exchange took place (also known as *non-repudiation*). *Digital signature* ensures accountability.

Authenticity: Authenticity means that you know who you are talking to and that you can trust the person. Without authenticity, you have no way to be sure that anyone is who they say they are.

Authentication ensures that the resource (human or machine) at the other end of the session really is what it claims to be.

These objectives are closely related to the type of information that is being transferred. The first example that always comes to mind is transactions that involve credit card numbers. However, there are many possible uses for WWW security enhancements. For example, imagine that a college wants to replace its correspondence courses by classes entirely based on World Wide Web. This venture involves sending many different types of documents with a variety of security objectives. Here are some examples:

- We want to advertise our WWW program to the public with no restrictions.
- We want to ensure that the course materials are only available to registered students so we apply *access control* to them.
- When the students take their online exams, we need to be sure that the papers really come from the student and we also want to protect them in transit to prevent cheating. This exchange needs both *privacy* and *authentication*.
- Finally, the student receives the diploma from the dean of the university and goes out into the job market armed with this prestigious document. The student needs to be able to prove that it was really signed by the dean and that it was really received. This exchange requires *authentication* and *accountability*.

1.9 Web Servers Security Facilities

The application level communication protocol used by the World Wide Web is Hypertext Transfer Protocol (HTTP). HTTP includes a simple user ID and password based authentication scheme known as *basic authentication*. The implementation of basic authentication is server specific, but in general, they all use it for two purposes:

- As a mechanism to identify which user is accessing the server.
- To limit users to accessing only specific pages.

Basic authentication is an attempt to address two of our security objectives (access control and authentication). However, it does not address confidentiality and data integrity.

To provide more advanced security features, a protocol that can provide encryption, digital signatures, and authentication is needed.

1.9.1 What is HTTP Basic Authentication?

Basic authentication is based on user IDs and passwords. You have to configure the server to identify which parts of the document tree can be accessed and which ones are protected. The zones of protection are called *realms*. Each realm is associated with a set of user IDs and passwords that are allowed access. Realms can contain any kind of server objects such as CGI programs as well as HTML pages. When a client requests a URL, the server checks to see if it requires user authentication. If it does, the server rejects the request and the browser pops up a dialog box on the user′s display asking for a user ID and password. The browser resends the request but with the addition of the user ID and password information provided by the user.

There is one obvious loophole in HTTP basic authentication: The user ID and password are included in the packet header, which means that they can be captured by anyone with a network sniffer or trace tool at any place in the session.

How serious is this exposure? Within the corporate network, it may not be a big problem. In the Internet, it is a different story. Here you have to assume that someone, somewhere is tracing everything you send. Clearly, HTTP basic authentication should not be used to protect critical resources.

You can make basic authentication secure by providing an encrypted connection for it to operate in.

1.9.2 What is Encryption?

Encryption in its simple form scrambles a message so that it cannot be read until it is unscrambled later by the receiver. The sender uses an algorithmic pattern or **key** to scramble or *encrypt* the message. The receiver has the *decryption* key. Encryption ensures confidentiality in transmissions sent over the Internet.

The are basically three techniques:

- Symmetric-key encryption
- Asymmetric or public-key encryption
- Hashing functions

1.9.2.1 Symmetric-Key Encryption

Symmetric-Key encryption (also sometimes called *bulk* encryption) is what most people think of as a secret code. The essence of symmetric keys is that both parties must know a shared secret (some kind of pattern called secret key). This pattern is used by the sender to encrypt the message and by the receiver to decrypt the message. The result is a scrambled message that can only be interpreted reversing the encryption process using the same secret key.

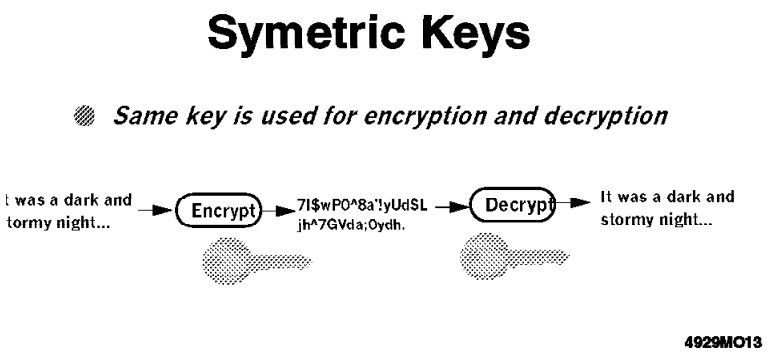


Figure 14. Symmetric-Key Encryption

Using modern computers, symmetric-key encryption is fast and secure. Its effectiveness is governed by two main factors:

- The size of the key. All symmetric-key algorithms can be cracked but the difficulty of doing so rises exponentially as the key size increases. With

modern computers, there is no problem in encrypting with keys that are large enough to be impossible to economically crack.

- The security with which the key is disseminated and stored. Since both partners in a symmetric-key system must know the secret key, there must be a safe way for it to be transmitted from one to another. It is, therefore, vital to protect the key when you are sharing it with the people you want to communicate with as well as when it is stored on either of the partner′s systems.

  The most commonly used symmetric-key encryption methods are:

  – Data Encryption Standard (DES). This was defined as a standard by the U.S. Government in 1977 and was originally developed by IBM. The DES standard operates in 64-bit blocks, using a 56-bit encryption key.

  – RC2 and RC4 from RSA Data Security Inc. The RCx ciphers are symmetric-key algorithms that are designed to provide an alternative to DES. They have the advantages of executing faster than DES and also permitting the use of a range of key sizes.

  – International Data Encryption Algorithm (IDEA). IDEA is another symmetric block-cipher similar to DES. IDEA also encrypts in 64-bit blocks but is has a larger 128-bit key.

1.9.2.2 Asymmetric or Public-Key Encryption

It is quite easy to understand how a symmetric-key algorithm works, at least to an intuitive level. Public or asymmetric-key systems are more difficult to envision, although they are not necessarily more complex, at least mathematically speaking. Instead of having one shared key, a public-key system has a **key pair**.

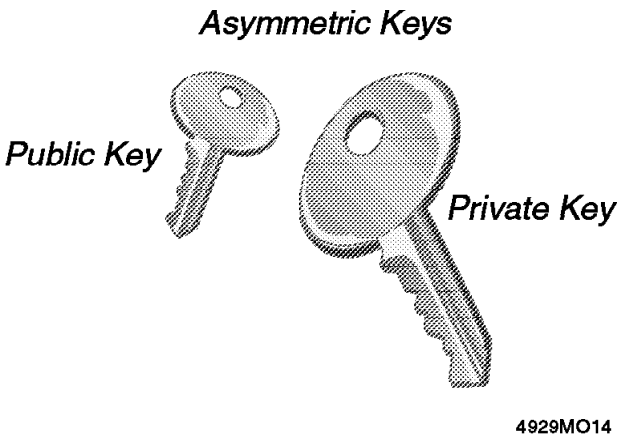


Figure 15. Asymmetric-Key Encryption

The key pair is made up of a **public key** and a **private key**. As the name suggests, the private key is a secret known only by the owner. The public key is made generally available. As a sender, you can broadcast the public key to whomever you want to communicate securely. You hold on to the private key and protect it. The ingenious part is this: anything encrypted using one half of the key can only be decrypted using the other half. As a result, only you can decrypt a message that has been encrypted with your public key because only

you have the private key. Reflexively, only someone who has your public key can decrypt a message that you encrypt using your private key.

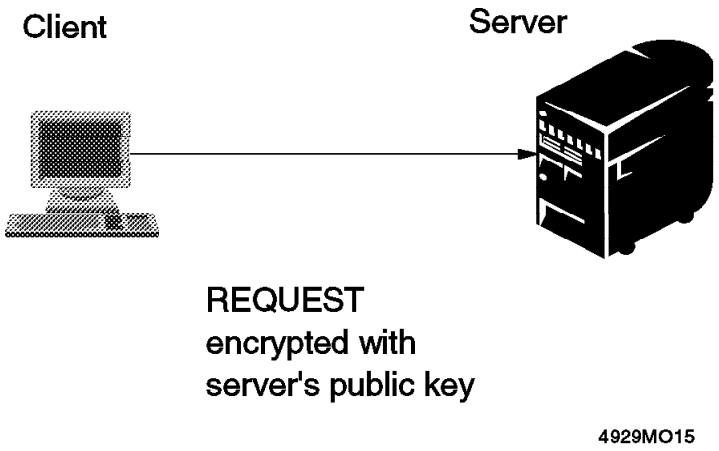


Figure 16. Public-Key Cryptography: Encrypting the Client Request with your Public Key

The flow shown in Figure 16 is used to give *data privacy* since the encrypted data can only be interpreted by the target system (the owner of the private key).

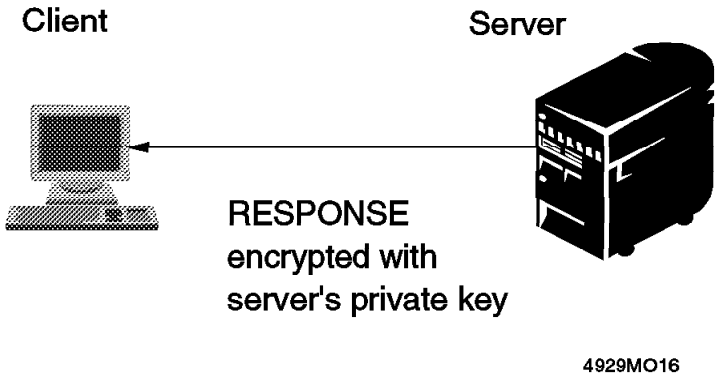


Figure 17. Public-Key Cryptography: Encrypting the Server Response with your Private Key

The flow shown in Figure 17 does *not* guarantee data privacy since anyone that knows the public key can decrypt your server's response encrypted with your private key. What it *does* give you is a method to *authenticate* the sender because only the owner of the private key can encrypt the data.

Public or asymmetric-key cryptography algorithms tend to be much less efficient than symmetric-key systems in terms of the computing power they consume. On the other hand, they do not suffer from key distribution problems. Public-key systems are often used in combination with symmetric-key systems. Public-key systems are used for distribution of symmetric keys and authentication purposes, leaving the bulk encryption job to the symmetric-key cipher.

A public-key cryptography system commonly used is the RSA algorithm patented by RSA Data Security Inc.

1.9.3 What are Secure Hash Functions or Message Digests?

As we discussed before, public-key and symmetric-key cryptography techniques can provide data privacy and sender authentication. We still need to find a way to provide integrity and accountability as stated in our security objectives in Section 1.8.1, “What Are Your Security Objectives?” on page 23. The technique usually used to implement integrity is *hashing* or *message digest*.

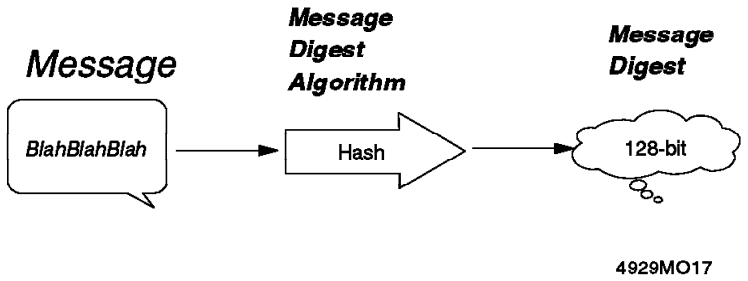


Figure 18. Message Digest

The following principal attributes are a secure hashing function:

1. It is a one-way process. It is impossible (or at least difficult) to reconstruct the original data from the hashed result.
2. The hashed result is not predictable. Given one set of data, it is extremely difficult to find another set of data with the same hashed result.

How can we use these functions to our advantage? The sender of the message (the server) creates a 128-bit message digest based on the message you want to send.

The receiver executes the same hash function with your message, which should result in the same message digest you sent. This tells the receiver that the message has not been altered in transit. Thus, we have achieved the *integrity* objective.

1.9.4 What is Digital Signature?

Digital signatures are used to achieve another security objective (accountability). Often the source of the message is at least as important as its content. Digital signatures can be used to identify the source of a message. You can compare a digital signature to a finger print that travels with your message or to the signature on your check that tells the bank that the check has really been written by you. Digital signature uses a message digest. You sign the message by encrypting the message digest with your private key. Your message is sent along with the encrypted message digest. Because you have signed your message with your private key, the receiver knows it comes from you.

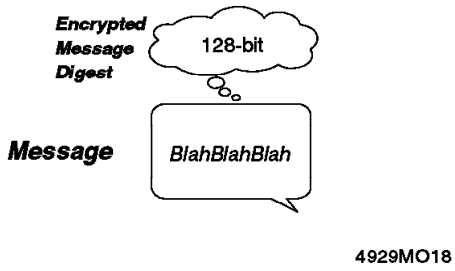


Figure 19. Signed Message - Digital Signatures

To achieve *non-reputability*, you want to guarantee both partners accountability.

If you are going to both sign and encrypt the message, follow this process:

1. Sign the message by encrypting the message digest with your private key.
2. Encrypt the message by encrypting the message and the encrypted message digest with the intended receiver's public key.
3. The receiver decrypts the message and the encrypted message digest with a private key.
4. The receiver decrypts the message digest with your public key.
5. The receiver re-creates the message digest from your message.
6. The receiver compares the message digest that you sent with the one re-created.

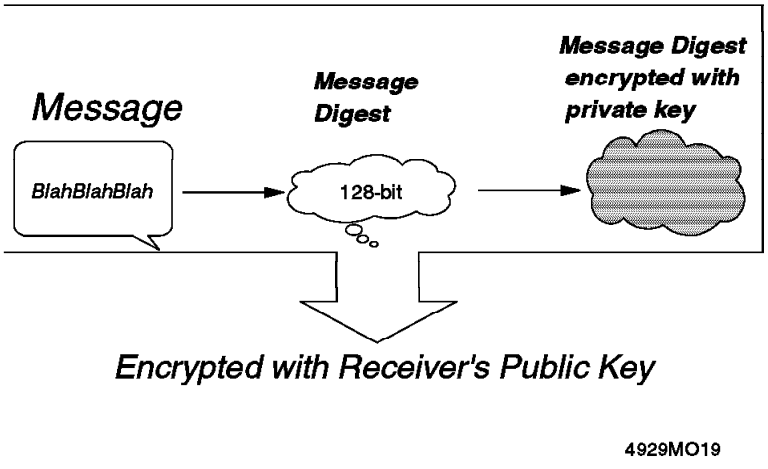


Figure 20. Encryption with Receiver's Public Key

Because you have signed the message with your private key, it is safe for the receiver to assume that the message came from you. Because the message digest was successfully compared, integrity is also assured. As you can see, digital signature can ensure data integrity and accountability for an Internet transmission.

1.9.5 What is Authentication?

Authentication is the process used to verify identity so you can make sure that others are who they say they are. There are two ways in which the server uses authentication:

- Digital signature
- Digital certificates

As covered in Section 1.9.4, “What is Digital Signature?” on page 28, digital signatures ensure accountability. But how can we make sure that the person (or system) sending the message is a *”decent citizen”*? You look at the sender’s **digital certificate**. If you think of a digital signature as some kind of picture ID, the digital certificate is a passport or driver’s license. In other words, it is a picture ID issued by a recognized authority that you trust. You know that the document has been issued by a trustworthy authority (the police or the government) because of some unique characteristics that make it difficult to forge the passport or the driver license. Likewise, digital certificates include some unique information that identifies the trusted authority issuing the certificate.

You base your trust for the authenticity of the sender on whether you trust the third party that issued the *digital certificate* or not. The third party that issues digital certificates is called a **certification authority**. At the time of writing, the most popular commercially accepted Certification Authority is Verisign Inc.

There are different types of certificates. X.509 certificates are made up of:

- The public key of the person (or server) being certified
- The name and address of the person being certified, also known as **Distinguished Name**
- The digital signature of the Certification Authority
- The issue date
- The expiration date
- Issuer distinguished name
- Serial Number

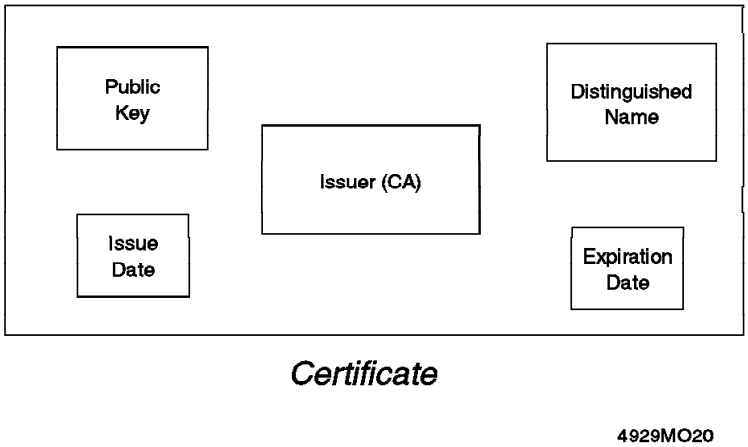


Figure 21. Certificate

The distinguished name is the name and address of the person or organization requesting the certificate. You enter your distinguished name as part of requesting a certificate. The digitally-signed certificate includes not only your own distinguished name but the distinguished name of the Certification Authority.

Certification authorities make their certificates readily available by placing them on public web sites and having them preloaded in some browsers.

When you designate the public key and certificate from a certification authority to be a trusted root key means that your server (or browser) trusts anyone who has a certificate from that Certification Authority. You may have many trusted roots as part of your server or browser. In fact, most servers and secured browsers include several default trusted root keys, and you can add others as needed.

To communicate securely, the receiver in the transmission must trust the Certification Authority that issued the certificate the sender is using. This is true whether the receiver is a Web browser or server. As a result, anytime a sender signs a message, the receiver must have the corresponding Certification Authority′s certificate and public key designated as a trusted root key.

1.9.6 What is Secured Sockets Layer (SSL)?

The SSL protocol was originally created by Netscape Inc. and RSA Data Security, but now it is implemented in World Wide Web browsers and servers from many vendors. SSL makes use of a number of cryptographic techniques such as public key and symmetric key encryption, digital signatures, and public key certificates.

SSL has two main objectives:

1. To ensure *confidentiality* by encrypting the data that a client and server send.
2. To provide *authentication* of the session partners using RSA public key methods. The session partner must use a private key to encrypt the data. Most current implementations only require the server to be authenticated in this way, although the protocol does allow for client authentication.

HTTPS (**not** to be confused with S-HTTP) is a unique protocol that combines SSL and HTTP. You need to specify **https://** as an anchor in HTML documents that link to SSL-protected documents. A client user can also open a URL by specifying **https://** to request SSL-protected documents.

Because HTTPS (HTTP+SSL) and HTTP are different protocols and usually use different ports (443 and 80, respectively), you can run both secure and non-secure HTTP servers at the same time. As a result you can choose to provide information to all users using no security and specific information only to browsers who make secure requests. This is how a retail company on the Internet can allow users to look through the merchandise without security, but then fill out order forms and send their credit card numbers using security. Although SSL is normally used to provide secure encapsulation of HTTP, it can be applied to any TCP/IP application. A number of implementations for other protocols have been developed.

There are two parts to SSL:

- The *handshake*, in which the session partners introduce themselves and negotiate session characteristics.
- The *record protocol*, in which the session data is exchanged in an encrypted form.

1.9.6.1 The SSL Handshake

Figure 22 shows a simplified version of the SSL handshake.

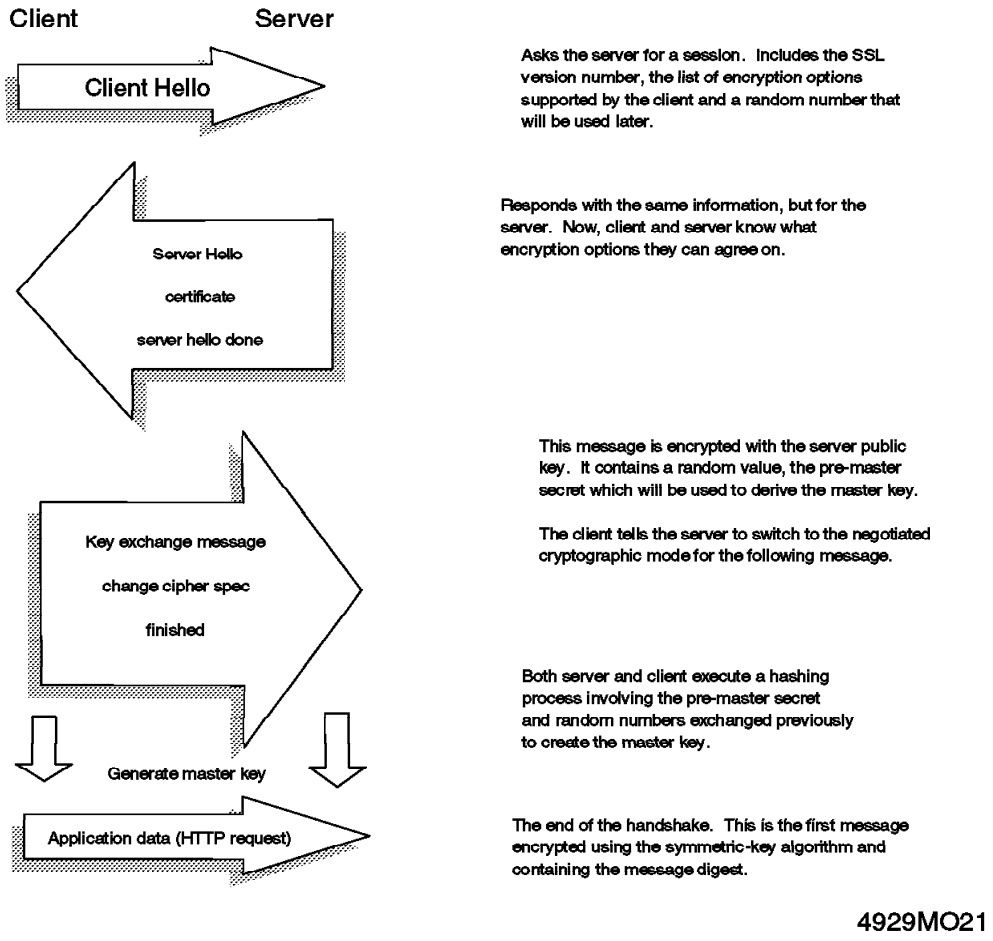


Figure 22. SSL Handshake Process

The two *hello* messages are used to exchange information about the capabilities of the client and server. This includes a list of *ciphersuites* (combinations of cryptographic algorithms and key sizes that the client and server accept for the session). Also, the server provides a public key certificate. This is the method by which SSL checks identity and authenticity of the session partner. In this example, we only show the steps for server authentication but if client authentication is required, there is another message exchange using the client public key. Finally, the session partners separately generate an encryption key, the *master key*, from which they derive the keys to use in the encrypted session that follows.

You can see from this example that there is significant additional overhead in starting up an SSL session compared with a normal HTTP connection. The

protocol avoids some of this overhead by allowing the client and server to retain session key information and to resume that session without negotiating and authenticating a second time.

1.9.6.2 Using SSL in Practice

The negotiation and authentication process of the SSL handshake is rather complex but fortunately, it is transparent to the user. In fact, all that a user has to do to enter an SSL connection is to alter the URL prefix from http: to https:. This acts as a trigger to the browser software to start the SSL handshake.

A browser that does not have support for HTTP over SSL naturally is not able to request URLs using HTTPS. The non-SSL browsers do not allow submission of forms that need to be submitted securely.

Once the SSL connection has been established, the browser gives the user a visual indication. In the case of Netscape Navigator, this is a key symbol at the lower left of the window (see Figure 23).

Figure 23. SSL Session Indicator in Netscape

From the point of view of the webmaster, SSL is also quite simple. First, it needs to generate a key pair for the server and obtain a certificate for it. Normally, this involves providing documentation to a certifying authority and paying an annual fee, although it is also possible to generate your own certificates for testing and intranet use.

Once the server certificate has been installed, the webmaster can create HTML links with an https: prefix to cause SSL to be invoked. For example:

```
<A HREF=https://my_server/secret.doc>Go into SSL</A>
```

Note: HTTP with SSL (HTTPS) is often 2-3 times slower than plain HTTP.

1.9.6.3 SSL and Certification Authorities

You can see from Figure 22 on page 32 that authentication in SSL depends on the client being able to trust the server's public key certificate. A certificate links the description of the owner of a key pair to the public part of the key. The validity of a certificate is guaranteed by the fact that it is signed by some trusted third party, the *certifying authority* (CA). But how does a certifying authority become trusted? In the case of an SSL-capable browser, the certificates of trusted authorities are kept in a key database, sometimes called a key-ring file. The list of top-level authorities is pre-installed when you get the browser. Figure 24 shows part of the list of CA certificates provided by Netscape Navigator.

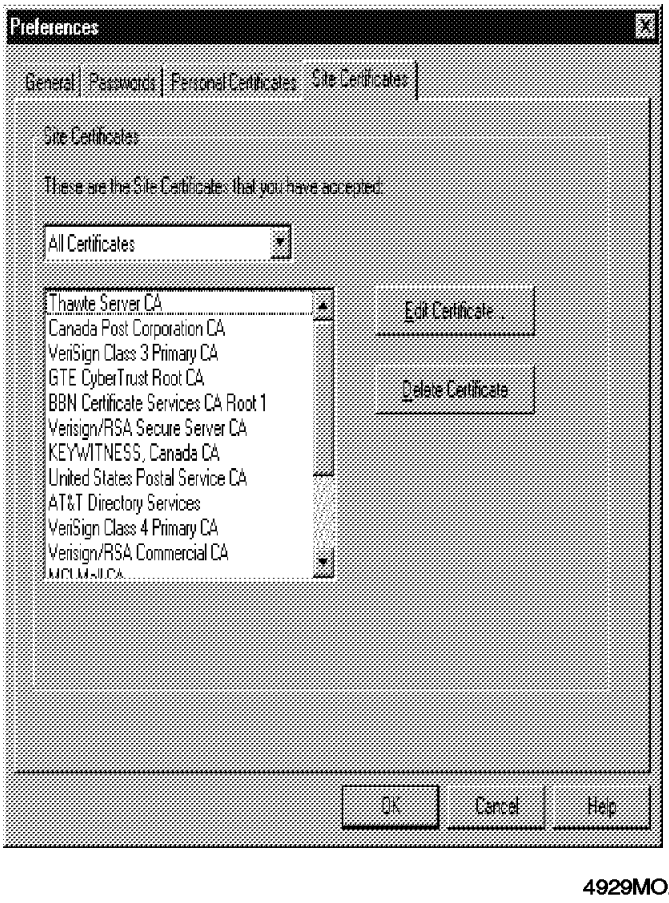


Figure 24. Certifying Authorities in Netscape Navigator

To conduct commercial business on the Internet, you use a CA such as VeriSign, Inc., who is widely known by clients and servers.

For a private Web network within your own company, university, or group, you can be your own CA or use some test certificate issued by, for example, Entrust Technology (visit their site at `www.entrust.com`). Entrust gives away free demo certificates with the objective of providing Web service providers and end users with an easy and inexpensive way to set up an SSL-secured Web session.

If your server is using a demo or test certificate or one generated by yourself, a secure browser tells you that you are connecting to a secure server whose certificate is not from a known CA. This approach is useful for secure transactions in an Intranet environment.

Chapter 2. Start Here by Securing OS/400®

This chapter contains information about the steps that should be done to secure OS/400. Completing these steps ensures that if you make a mistake later in an application, OS/400 is still there to protect you.

You need to check exposures such as the security level that you are working with, your passwords rules, the security system values you defined, the user profiles, the authority to your system objects and to the AS/400 integrated file system, and your TCP/IP applications and ports.

Copied Information

For your convenience, we have copied information from different manuals that were valid when this redbook was written. You must not use this copied information when you start to improve the security implementation on your AS/400 system. You must use the manuals that are valid for your installed release of OS/400. Otherwise, your work may not reach the expected level of quality.

2.1 Mandatory Manuals

The *Tips and Tools for Securing Your AS/400* manual helps you understand and use the integrated security tools (formerly known as Security Toolkit). In addition, it has many valuable tips about security. You must be familiar with the *AS/400 Security - Reference* manual. The manual, *SECUREWAY: AS/400 and the Internet*, provides specific information about security considerations that must be taken into account when connecting an AS/400 system to the Internet. The redbook, *An Implementation Guide for AS/400 Security and Auditing*, may also be helpful.

Use these sources of information when you establish your main line of defense.

2.2 Change in Security Thinking

Connecting to the Internet adds a new dimension to security thinking. You need to ″think in layers″. It is like dressing for the cold: always assume that your first, second... lines of defensewill be broken. Understand that connecting your system to the Internet is to expose it to millions of users: even if only a very small percentage of them have malicious intentions it is still a large number.

Never assume that because you wouldn′t know how to do it, nobody will be able to break your first line of defense. There are many smart people out in the net with plenty of tools and times in their hands to cause damage.

One advantage that AS/400 users have over other platforms is the robust set of security features and functions of OS/400. If everything else fails, OS/400 has the potential to be there for your to prevent serious damages. But, we say, *potential*: you must take advantage of OS/400 security capabilities by tightly configuring security on a system that will be connected to the Internet (even if it is through a firewall).

2.2.1 Security Implemented on Your AS/400 System as Stand-Alone System

You should not connect your AS/400 to the Internet, not even using a sophisticated firewall, until your main line of defense is established. You may not like this, but security has a cost. It does not come free, never has, never will.

2.2.2 Proceed With Care

Keep it simple.

Complexity and security are difficult to combine. The more complex your implementation of security is, the greater the chance that you have overlooked something and left a hole in your defense.

Simple maintenance requires simple implementation.

2.3 System Values

When we talk about system values that control your system, we can break them into four groups: security level, system values that control passwords, general security system values, and system values that control auditing.

2.3.1 Security Levels

You can change the security level (QSECURITY) using the Change System Value (CHGSYSVAL) command. You must consult the *AS/400 Security - Reference* manual before you do so! A change to this system value takes effect on the next IPL.

The system offers five levels of security:

Level 10: Should not be used on any production system.

Level 20: Should not be used when connecting to the Internet.

Level 30: The system requires a user ID and password for signing on. Users must have authority to use objects. This is called **resource security**.

Level 30 is not good enough when connecting to the Internet.

Level 40: The system requires a user ID and password for signing on. In addition to resource security, the system provides **integrity protection** functions. The integrity protection functions 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 AS/400 system with V3R7 or a 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. Security level 50 adds enhanced integrity protection such as:

- Validation of parameters for interfaces to the operating system.
- Restriction of message-handling between system state programs and user state programs.

Security level 50 is intended for AS/400 systems with high security requirements.

Tip

To display the current security level on your system, use the DSPSECA command shown in Figure 25 on page 39. The difference between DSPSECA and DSPSYSVAL QSECURITY is that DSPSYSVAL QSECURITY shows the level to be used after the next IPL; DSPSECA shows the current level.

```
                        Display Security Attributes

User ID number . . . . . . . . . . . . . . :   693
Group ID number  . . . . . . . . . . . . . :   121
Security level . . . . . . . . . . . . . . :   50
```

Figure 25. Display IPLed Value of QSECURITY System Value

2.3.2 Password Rules

Good passwords and passwords well protected are important. The help desk at a U.S. university found that 80% of reported problems were due to poor passwords or lack of password protection.

Set a policy that states that passwords must not be trivial and must not be shared. Set system values to help you with enforcement.

Copied Table

Table 1 shows recommended system value settings.

It is copied from the *Tips and Tools for Securing Your AS/400* manual. Use the manual valid for the version of your operating system when you start your work.

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 criteria:

- A list of the criteria for passwords.
- Examples of passwords that are and are not valid.
- Suggestions for how to think of a good password.

Table 1 (Page 1 of 2). 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)
QPWDMINLEN	The minimum number of characters in a password.	6
QPWDMAXLEN	The maximum number of characters in a password.	8

Table 1 (Page 2 of 2). System Values for Passwords

System Value Name	Description	Recommended Value
QPWDRQDDIF	How long a user must wait before using the same password again.	5 or less (expiration intervals)[1]
QPWDLMTCHR	What characters may not be used in passwords.	AEIOU#$@
QPWDLMTAJC	Whether the system prevents adjacent characters that are the same.	1 (yes)
QPWDLMTREP	Whether the system prevents the same character from appearing more than once in the password.	2 (not allowed consecutively)
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)
QPWDVLDPGM	What exit program is called to validate a newly assigned password.	*NONE

Note:

[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 *Security - Reference* book provides more information about how these system values work together.

Use the Security Toolkit menu to print the system security values shown in Figure 26 on page 41.

```
                                        System Security Attributes                                      Page     1
5716SS1 V3R7M0  961108                                                                    SystemA   12/10/96  11:45:38
 System Value
 Name                 Current value           Recommended value
    QALWOBJRST        *ALL                    *NONE
    QALWUSRDMN        *ALL                    QTEMP
    QATNPGM           QEZMAIN   QSYS          *NONE
    QAUDENDACN        *NOTIFY                 *NOTIFY
    QAUDFRCLVL        *SYS                    *SYS
    QAUDCTL           *NONE                   *AUDLVL   *OBJAUD
                                              *NOQTEMP
    QAUDLVL           *NONE                   *AUTFAIL  *CREATE
                                              *DELETE   *SECURITY
                                              *SAVRST
    QAUTOCFG          1                       0
    QAUTORMT          1                       0
    QAUTOVRT          9999                    0
    QCMNRCYLMT        0 0                     0 0
    QCRTAUT           *EXCLUDE                Control at library level.
    QCRTOBJAUD        *NONE                   Control at library level.
    QDEVRCYACN        *MSG                    *DSCMSG
    QDSCJOBITV        240                     120
    QDSPSGNINF        1                       1
    QINACTITV         *NONE                   60
    QINACTMSGQ        *ENDJOB                 *ENDJOB
    QLMTDEVSSN        0                       1
    QLMTSECOFR        0                       1
    QMAXSGNACN        3                       3
    QMAXSIGN          5                       3
    QPWDEXPITV        31                      60
    QPWDLMTAJC        0                       1
    QPWDLMTCHR        *NONE                   AEIOU@$#
    QPWDLMTREP        0                       1
    QPWDMAXLEN        10                      8
    QPWDMINLEN        1                       6
    QPWDPOSDIF        0                       1
    QPWDRQDDGT        0                       1
    QPWDRQDDIF        0                       1
    QPWDVLDPGM        *NONE                   *NONE
    QRETSVRSEC        0                       0
    QRMTIPL           0                       0
    QRMTSIGN          *FRCSIGNON              *FRCSIGNON
    QRMTSRVATR        0                       0
    QSECURITY         50                      50
    QSRVDMP           *DMPUSRJOB              *NONE

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

Figure 26. Print System Security Attributes (PRTSYSSECA) Report - Example

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).

One-way encryption is used to store the password on the system. No method is available to decode it. If a password is forgotten, the security officer and any other user with special authority *ALLOBJ and *SECADM can use the Change User Profile (CHGUSRPRF) command to assign a temporary password and set that password to expired, which requires the user to assign a new password at the next sign-on.

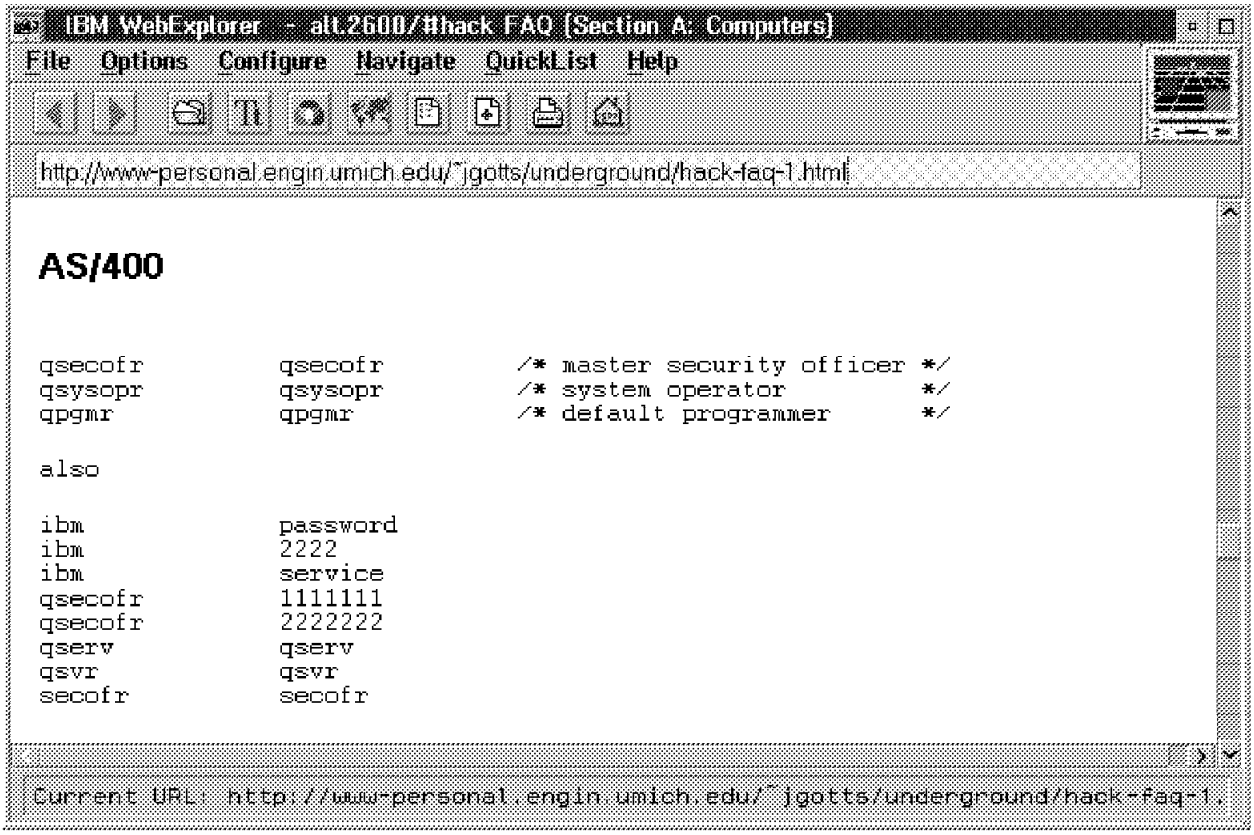


Figure 27. AS/400 Default Passwords Found in alt.2600/#hack FAQ!

Figure 27 shows just how serious one should take Internet security. This home page shows common user IDs and passwords for AS/400 systems and other IBM and OEM system user IDs and passwords. Many hackers attack a system using a dictionary of passwords; this clearly shows the AS/400 system is thought of as a target.

TIP

The Analyze Default Passwords (ANZDFTPWD) command allows you to print a report of all the user profiles on the system that have a default password and to take an action against the profiles. A profile has a default password when the user profile name matches the profile's password.

The format of the report depends on what action is taken against the profiles. When no action is taken, each entry contains the user profile name, the user profile's status (STATUS), whether the profile's password is expired (PWDEXP), and the text description associated with the profile (TEXT). When an action is taken against the profiles, each entry also contains the user profile's STATUS and PWDEXP values after the profile has been changed. Figure 28 on page 43 shows an example of user profiles that were disabled by the ANZDFTPWD command.

You can get the same report using the Security Toolkit menu.

```
                                  User profiles with default passwords.                                  Page     1
5716SS1 V3R7M0  961108                                                                  SYSTEMA   11/25/96  13:43:00
 Action taken against profiles  . . . . . . :   *DISABLED
 User
 Profile        STATUS         PWDEXP     Text
 USER01         *DISABLED       *NO       User number one
 USER02         *DISABLED       *NO       User number two
 TEST2          *DISABLED       *NO       Test user
 TEST1          *DISABLED       *NO       Test user
                                 * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 28. Analyze Default Passwords (ANZDFTPWD) Report - Example

The Analyze Default Passwords (ANZDFTPWD) command is an important tool to investigate if the IBM-supplied profiles do not have default passwords. These passwords are published, and they are the first choice of anyone who is trying to break into your system. Use the Change User Profile (CHGUSRPRF) command to change the password to the recommended value. Table 2 shows these recommended values.

Copied Table

Table 2 is copied from the *Tips and Tools for Securing Your AS/400* manual. Use the manual valid for the version of your operating system when you start your work.

Table 2. Passwords for IBM-Supplied Profiles

User ID	Password	Recommended Value
QSECOFR	QSECOFR[1]	A nontrivial value known only to the security administrators. **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]
QSRV	QSRV	*NONE[2]
QSRVBAS	QSRVBAS	*NONE[2]

Note:

[1] Beginning with V3R2 and V3R7, the system arrives with the *Set password to expired* value for QSECOFR set to *YES. The first time that you sign on to a new V3R2 or V3R7 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.

We also recommend that you verify that the passwords of some special user profiles used by some TCP/IP server applications are *NONE. Table 3 shows an example of these user profiles:

Table 3 (Page 1 of 2). User Profiles Used by TCP/IP Server Applications

User Profile	TCP/IP Application
QTMHHTTP and QTMHHTP1	HTTP Server
QTMPLPD	LPD Server
WWWUSER[1]	I/NET web server

Table 3 (Page 2 of 2). User Profiles Used by TCP/IP Server Applications	
User Profile	**TCP/IP Application**
QTMTWSG	Workstation Gateway server
ANONYMOUS[1]	FTP Server
QTMPLPD	LPD Server
QTCP	Other TCP/IP server applications
Note: [1] The name of these user profiles may be different if they were defined with other names during its configuration.	

You also must check if the passwords for Dedicated Service Tools are not the defaults. Table 4 shows the shipped values for these special user IDs. They must be changed. The *Backup and Recovery - Advanced* book provides information about changing the Dedicated Service Tools passwords.

Copied Table

Table 4 is copied from the *Tips and Tools for Securing Your AS/400* manual. Use the manual valid for the version of your operating system when you start your work.

Table 4. 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	A nontrivial value known only to the security administrator.[2]
Security capability	QSECOFR	QSECOFR	A nontrivial value known only to the security administrator.[2]
Note: [1] A user ID is only required for V3R6 and V3R7. [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.			

2.3.3 General Security Values

Copied Table

Table 5 on page 45 is copied from the *Tips and Tools for Securing Your AS/400* manual. Use the manual valid for the version of your operating system when you start your work.

Table 5 on page 45 shows general system values that can be set to make it more difficult for an unauthorized person to sign on to your system.

Table 5. 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 automatically creates if no device is available for use.	0
QDEVRCYACN	What the system does when a device reconnects after an error.	*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.	*DSCJOB
QLMTDEVSSN	Whether the system prevents a user from signing on at more than one workstation at the same time.	1 (Yes)
QLMTSECOFR	Whether users with *ALLOBJ or *SERVICE special authority can sign on only at specific workstations.	1 (Yes)[1]
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)

Note:

[1] If you set the system value to 1 (Yes), you need to explicitly authorize users with *ALLOBJ or *SERVICE special authority to devices. The simplest way to do this is to give such users *CHANGE authority to specific devices.

Changing Sign-On Error Messages: Hackers want to know when they are making progress towards 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 6 on page 46 shows the recommended text.

Copied table

Table 6 on page 46 is copied from the *Tips and Tools for Securing Your AS/400* manual. Use the manual valid for the version of your operating system when you start your work.

Table 6. 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.

TIP

You can change the message text using the Change Message Description (CHGMSGD) command. Example:

```
CHGMSGD MSGID(CPF1107) MSGF(QCPFMSGF) MSG('Sign-on information is not
        correct.') SECLVL(*NONE)
CHGMSGD MSGID(CPF1120) MSGF(QCPFMSGF) MSG('Sign-on information is not
        correct.') SECLVL(*NONE)
```

Note: If a system upgrade is done, the text of the messages you changed are set to the default values. You can create a CL program to change these message descriptions after every system upgrade using the CHGMSGD command.

2.3.4 System Values for Auditing

The AS/400 system has an audit journal, QAUDJRN. Events related to security such as changes in system values, ownership, and so on, can be logged to this journal.

Use the Security Toolkit to start logging and printing the reports from the journal. See *Tips and Tools for Securing Your AS/400* and *AS/400 Security - Reference* for more information.

2.4 User Profiles Security

When your system values are in order, you should concentrate on getting control over the user profiles. On most systems, this is a neglected area. We recommend that you use the reports that are provided by the Security Toolkit to find out where you stand. When you know your current status, you can always find the road to your target. The *Tips and Tools for Securing Your AS/400* manual, the *AS/400 Security - Reference* manual, and the redbook, *An Implementation Guide for AS/400 Security and Auditing*, provide you with valuable information and guidance.

Here is a list of concerns:

- Public authority to user profiles should be *EXCLUDE.
- Do not hand out special authority. Only give special authority to those who need it, and before you do, question this need.
- Give as many users as you possibly can limited capabilities.
- Verify that a user is a member of the correct group or groups.
- Verify that the groups have correct members.

- Check if a user has private authorities to objects.

2.4.1 Scheduling 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, you may want to disable user profiles with *ALLOBJ special authority (including the QSECOFR user profile) during off-hours.

The *Tips and Tools for Securing Your AS/400* manual explains the different options provided by the Security Toolkit to control user profiles.

2.5 Resource Security

Resource security defines which users are allowed to use objects on the system and what operations they are allowed to perform on those objects. You define who can use an object in several ways:

- **Public Authority:** The public consists of anyone who is authorized to sign on to your system. Public authority is defined for every object on the system. Public authority to an object is used if no other authority is found for the object.
- **Private Authority:** You can define specific authority to use (or not use) an object. You can grant authority to an individual user profile or to a group profile.

You can give authority to individual users, groups of users, and the public:

- **Group Authority:** Group profiles may be given authority to use objects on the system. A member of the group gets the group′s authority unless an authority is specifically defined for that user. Group authority is also a form of private authority.
- **Object Ownership:** Every object on the system has an owner. The owner has *ALL authority to the object by default. However, the owner′s authority to the object can be changed or removed. The owner′s authority to the object is not considered private authority.

  If you want to secure your resources, you must have control over which user profile owns what objects. As an owner, you can do what you want with an object.

  If you have group profiles who own objects (such as files or libraries), you may be in trouble. Every member of the group inherits the group profile′s authorities. Objects owned by a group profile cannot be secured from the members of the group. It has been this way since day one of the AS/400 system but it is widely ignored.
- **Primary Group Authority:** You can specify a primary group for an object and the authority the primary group has to the object. Primary group authority is stored with the object and may provide better performance than private authority granted to a group profile. Primary group authority is not considered private authority.

Authority means the type of access allowed to an object. Different operations require different types of authority. Authority to an object is divided into two categories:

- **Object authority** defines which operations can be performed on the object as a whole (object authority types: *OBJOPR, *OBJMGT, *OBJEXIST, *OBJREF, and *AUTLMGT).
- **Data Authority** defines which operations can be performed on the contents of the object (data authority types: *READ, *ADD, *UPDT, *DLT, and *EXECUTE).

Copied Table

Table 7, Table 8 on page 49, and Table 9 on page 49 are copied from the *AS/400 Security - Reference* manual. Use the manual valid for the version of your operating system when you start your work.

Table 7 describes the types of authority available and lists some examples of how the authorities are used.

Table 7 (Page 1 of 2). Description of Authority Types

Authority	Name	Functions Allowed
*OBJOPR	Object Operational	Look at the description of an object. Use the object as determined by the user's data authorities.
*OBJMGT	Object Management	Specify the security for the object. Move or rename the object. All functions defined are for *OBJALTER and *OBJREF.
*OBJEXIST	Object Existence	Delete the object. Free storage of the object. Perform save and restore operations for the object[1]. Transfer ownership of the object.
*OBJALTER	Object Alter	Add, clear, initialize, and reorganize members of the database files. Alter and add attributes of database files. Add and remove triggers. Change the attributes of SQL packages.
*OBJREF	Object Reference	Specify a database file as the parent in a referential constraint. For example, you want to define a rule that a customer record must exist in the CUSMAS file before an order for the customer can be added to the CUSORD file. You need *OBJREF authority to the CUSORD file. You need *OBJREF authority to the CUSMAS file to define this rule.
*AUTLMGT	Authorization List Management	Add and remove users and their authorities from the authorization list.
*READ	Read	Displays the contents of the object such as viewing records to a file.
*ADD	Add	Add entries to an object such as adding messages to a message queue or adding records to a file.
*UPDT	Update	Change the entries in an object such as changing records to a file.
*DLT	Delete	Remove entries from an object such as removing messages from a message queue or deleting records from a file.

Table 7 (Page 2 of 2). Description of Authority Types		
Authority	**Name**	**Functions Allowed**
*EXECUTE	Execute	Run a program, service program, or SQL package. Locate an object in a library or a directory.
Note: [1] If a user has save system (*SAVSYS) special authority, object existence authority is not required to perform save and restore operations on the object.		

2.5.1 Commonly Used Authorities

Table 8 shows the system-defined authorities available using the object authority commands and displays.

Table 8. System-Defined Authority				
Authority	***ALL**	***CHANGE**	***USE**	***EXCLUDE**
*OBJOPR	X	X	X	
*OBJMGT	X			
*OBJEXIST	X			
*OBJALTER	X			
*OBJREF	X			
*READ	X	X	X	
*ADD	X	X		
*UPD	X	X		
*DLT	X	X		
*EXECUTE	X	X	X	

Table 9 shows additional system-defined authorities that are available using the WRKAUT and CHGAUT commands.

Table 9. System-Defined Authority							
Authority	***RWX**	***RW**	***RX**	***R**	***WX**	***W**	***X**
*OBJOPR	X	X	X	X	X	X	X
*OBJMGT							
*OBJEXIST							
*OBJALTER							
*OBJREF							
*READ	X	X	X	X			
*ADD	X	X			X	X	
*UPDT	X	X			X	X	
*DLT	X	X			X	X	
*EXECUTE	X		X		X		X

2.5.1.1 Library Security

Most objects on the system reside in libraries. To access an object, you need authority both to the object itself and the library in which the object resides. For most operations, including deleting an object, *USE authority to the object library is sufficient (in addition to the authority required for the object itself). Creating a new object requires *ADD authority to the object library.

Special attention is required when a library is added to a user's library list. When it happens, the authority the user has to the library is stored with the library list information. The user's authority to the library remains for the entire job, even if the user's authority to the library is revoked while the job is running. This represents a potential security exposure.

2.5.1.2 Directory Security

When accessing an object in a directory, you must have authority to all of the directories in the path containing the object. You must also have the authority to the object to perform the operation you requested.

You may want to use directory security in the same way that you use library security. Limit access to directories and use public authority to the objects within the directory. Please read Section 2.6, "Integrated File System Security" on page 51 for more information.

2.5.1.3 Authorization List Security

You can group objects with similar security requirements using an authorization list. An authorization list conceptually contains a list of users and the authority that the users have to the objects secured by the list. Each user can have a different authority to the objects secured by the list.

You can also use an authorization list to define public authority for the objects on the list. If the public authority for an object is set to *AUTL, the object gets its public authority from its authorization list. You cannot use an authorization list to secure a user profile or another authorization list. Only one authorization list can be specified for an object.

2.5.1.4 Source Code and Compilers

Many security administrators put up great efforts to secure the payroll applications and leave the source physical files unprotected. From a security point of view, it should have been the other way around. Many do not even know which libraries contain source code (a paradise for an intruder). With access to the source code, the intruder can put you out of business.

If the user profile used by the intruder is authorized to the compiler and can create programs in a specific library, the intruder can put a logical bomb (a Trojan horse) in your system.

If the intruder can only access the source code, the intruder can still change the code, and if it is not detected, the next compilation is fatal.

2.6 Integrated File System Security

The **Integrated File System** is a part of OS/400 that supports input/output data streams and storage management similar to personal computer and UNIX operating systems while providing an integrating structure over all information stored in the AS/400 system.

A **file system** provides the support that allows users and applications to access specific segments of storage that are organized as logical units. These logical units are files, libraries, and objects. The file systems are ″root″, QOpenSys, QSYS.LIB, QDLS, QLANSrv, QOPT, QFileSrv.400, UDFS, NFS, and QNetware.

A **directory** is a special object that is used to locate objects by name. Each directory contains a list of objects that are attached to it. That list may include other directories. The integrated file system provides a hierarchical directory structure that allows users and application programs to access all objects in the AS/400 system. You might think of this directory structure as an inverse tree where the roots are at the top and the branches below. These directory branches represent directories in the directory hierarchy called sub-directories. Attached to the various directory and sub-directory branches are objects such as files. An object is located by specifying a path through the directories to the sub-directory to which the object is attached.

TIP

Securing your directory structure is similar to library security. You can limit the access to directories and its objects. You can do the following steps:

- Use a directory to store all files for a particular group of applications.
- Make public authority for all of the objects in the directory sufficient for the application needs (*CHANGE or *ALL).
- Restrict public authority to the directory itself.
- Give selected groups or individuals authority to the directory (*USE or *ADD if the applications require it).

When accessing an object in a directory, you must have authority to all the directories in the path containing the object. You must also have the necessary authority to the object to perform the operation you requested.

2.6.1 PC Virus

Using the integrated file system or the shared folders structure from a PC network, you are more exposed to common PC problems such as viruses.

A virus is a program that can change other programs to include a copy of itself. The virus program usually performs operations that can take up system resources or destroy data. When your users connect to the Internet, they might unintentionally download a program with a virus. They might store the infected program in a shared folder or in the integrated file system on your AS/400 system. That virus might be copied accidentally to other PCs in your network. You can follow some security solutions:

- On your AS/400 system, use object authority to control where PC users can create new objects. If your PC users use shared folders, use the authority to DLOs (document library objects) to limit them to create new documents in

specific folders. If your PC users use the integrated file system, use the authority to directories to control where they can place new objects.

- Ensure that most users do not have authority to create objects in the root directory. Change the public authority of the root directory from *RWX to *RX.
- Regularly run virus scan programs against the directories or folders where your PC users place new objects.
- Install virus-scan software on all PCs and require PC users to run it regularly. Consider including the virus scan program in every PC's startup routine.
- Consider staging the movement of new objects from private PC drives to a shared environment. Move them to a temporary drive (shared folder or directory) first. Have a system administrator move them to a shared environment after running a virus scan program.
- Educate your users both about viruses and about the risks of downloading programs from untrusted sources.

2.7 Basic TCP/IP Security

TCP/IP and the Internet are designed for openness and interoperability. The Internet clients and Internet servers from many different providers can communicate and exchange information successfully. You need to control your TCP/IP connection and applications to protect your system resources.

The *Tips and Tools for Securing Your AS/400* manual explains how to control the use of TCP/IP applications. You should only start the TCP/IP applications you need. You should also refer to the manual *SECUREWAY: AS/400 and the Internet* to plan the connection of your AS/400 system to the Internet.

2.7.1 TCP/IP Ports

Ports are used by TCP and UDP protocols to identify a unique origin or destination of communication with an application. Ports are integer values from one to 65 535. There are two unique sets of ports. One set is for TCP processing and the other is for UDP processing. They are completely independent sets of ports and have no relationship to one another.

Commonly used protocols and applications such as FTP and SMTP have assigned port numbers. These assigned port numbers are called **well-known ports**. TCP and UDP port numbers one to 1023 are reserved for the well-known ports and should not be used by user application programs. If the user specifies one of these ports, it can affect the operation of those applications.

The following list of well-known ports is not exhaustive and lists only assigned-to services that are widely implemented or of general interest. The list applies to both the TCP and UDP sets of ports.

Copied Table

Table 10 on page 53 is copied from the *TCP/IP Configuration and Reference* manual. Use the manual valid for the version of your operating system when you start your work.

Table 10. Well-Known Ports

Decimal	Keyword	Description
5	RJE	Remote Job Entry
7	ECHO	Echo
11	USERS	Active Users
13	DAYTIME	Daytime
20	FTP-DATA	File Transfer (Data)
21	FTP	File Transfer (Control)
23	TELNET	Remote Terminal Protocol
25	SMTP	Simple Mail Transfer Protocol
37	TIME	Time
42	NAMESERV	Host Name Server
43	NICNAME	Who is
53	DOMAIN	Domain Name Server
67	BOOTPS	Bootstrap Protocol Server
68	BOOTPC	Bootstrap Protocol Client
69	TFTP	Trivial File Transfer Protocol
70	GOPHER	Used to browse Internet resources
79	FINGER	Finger
80	WWW	World Wide Web HTTP
101	HOSTNAME	NIC Host Name Server
102	ISO-TSAP	ISO TSAP
103	X400	X400
104	X400SND	X400 SND
105	CSNET-NS	CSNET Mailbox Name Server
109	POP2	Post Office Protocol 2
110	POP3	Post Office Protocol 3
111	RPC	Sun™ RPC Portmap
137	NETBIOS-NS	NetBIOS Name Service
138	NETBIOS-DG	NetBIOS Datagram Service
139	NETBIOS-SS	NetBIOS Session Service
161	SNMP	Simple Network Management Protocol
512(TCP)[1]	EXEC	Remote Command Execution (REXEC)
515	LPD	Remote Printing
555	WGMAIL	IBM WorkGroup Mail Server
5061	WSG	Work Station Gateway
5110	AS-POP3	Client Access™ using IP
32110	AS-POP3-X	Client Access using IPX

Note:

[1] This is for TCP only; UDP port 512 is reserved for a different application.

To prevent someone from associating a user application such as a socket application with a port the system normally uses for a server application, do the following steps:

1. Type GO CFGTCP to display the Configure TCP/IP menu.
2. Select option 4 (Work with TCP/IP port restrictions).
3. On the Work with TCP/IP Port Restrictions display, specify option 1 (ADD).
4. For the lower port range, specify the decimal number of the application you want to protect. You can see the decimal number in Table 10 on page 53.
5. For the upper port range, you can specify *ONLY, or the maximum range for more than one application. Example: if you want to restrict all the POP ports, you must define 109 as the lower port range and 110 as the upper port range. You can see the decimal number in Table 10 on page 53.
6. For the protocol, specify *TCP.
7. 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 for a specific user, you automatically exclude all other users.
8. Repeat the preceding steps for the *UDP protocol.

Figure 29 shows an example of how to restrict the use of the HTTP port (port 80) and the TELNET port (port 23) for the TCP protocol. Only the QTCP, QTMHHTP1, and QTMHHTTP user profiles can access the HTTP port and only the QTCP user profile can access the TELNET port.

```
                      Work with TCP/IP Port Restrictions
                                                            System:   SYSTEMA
Type options, press Enter.
  1=Add   4=Remove

        --Port Range---                 User
Opt     Lower     Upper     Protocol    Profile
 _       _____     *ONLY     ________    ________
 _         80      *ONLY       *TCP      QTCP
 _         80      *ONLY       *TCP      QTMHHTP1
 _         80      *ONLY       *TCP      QTMHHTTP
 _         23      *ONLY       *TCP      QTCP




F3=Exit   F5=Refresh   F6=Print list   F12=Cancel   F17=Top   F18=Bottom
```

Figure 29. Restricting Port 23 (TELNET) and Port 80 (HTTP) - Example

Note: The port restriction takes effect the next time that you start TCP/IP. If TCP/IP is active when you set the port restriction, you should end TCP/IP and start it again.

You can find more information about the TCP/IP ports in the *TCP/IP Configuration and Reference* manual.

2.7.2 Disabling Client Applications (Sockets)

The Start Host Server (STRHOSTSVR) command is used to start the optimized host server daemons and the server mapper daemon. These server daemons are used by applications such as Client Access/400 when using TCP/IP. There is one server daemon for each of the host server types. In addition, there is one server mapper daemon for all host servers that provides support for client applications to obtain a particular host server daemon's port number. This port number is used by the client application to connect to the host server's daemon. The daemon accepts the incoming connection request and routes it to the server job for further processing.

You do not need to start all (*ALL) the servers, but just those you really want to enable client applications to be connected with. The host servers are *CENTRAL, *DATABASE, *DTAQ, *DRDA, *FILE, *NETPRT, *RMTCMD, *SIGNON, and *SVRMAP. If you are not using Client Access/400 with TCP/IP, do not start these servers.

2.8 Ten Rules of Security

Follow these 10 rules of security when securing applications and when evaluating applications offered by someone else:

1. Functions should require the same authority as similar functions in OS/400.
2. Do not require too much authority to perform the function.
3. If you create a profile for use with your application, create it without a password and with no special authority.
4. Create the application's objects with appropriate *PUBLIC authority:
   - Create all objects with *PUBLIC(*USE), except:
     - Create user *USRPRFs, *JOBDs, *SBSDs, and sensitive files with *PUBLIC(*EXCLUDE).
     - Create *DEVDs and *MSGQs with *PUBLIC(*CHANGE).
5. If a user profile is specified in a job description, create the job description as *PUBLIC(*EXCLUDE).
6. If the application programs adopt, be careful not to provide users access to a command line. (Watch for pop-up windows, function keys, menu options, and so on.)
   - Do not provide access to a command line through a program that adopts.
7. If the application program needs to adopt a powerful profile, call the program, do the necessary functions, and return immediately. *Make sure* the call to this program is a library-qualified call.
8. If the application adds libraries to the library list, remove them when the application has completed.
9. **Do not** store passwords in the clear.
10. If the language allows it, make library-qualified calls and object accesses.

2.9 Summary

When you start out to establish the main line of defense that is mandatory before connecting to the Internet, you should actively use the *Tips and Tools for Securing Your AS/400* and the *AS/400 Security - Reference* manuals together with the redbook, *An Implementation Guide for AS/400 Security and Auditing*.

Spend some time getting to know the functions in the Security Toolkit. It is time well spent.

What a Mess!

Keep it simple. A complicated and sophisticated implementation of security is the safe road to a mess. Avoid that road.

We recommend that you structure your work and perform the different tasks required to improve the security in the following sequence:

- Start using the QAUDJRN. Any change you make is logged in this journal, and more important, so are changes made by others. You do not get anywhere if changes made by you are changed by someone else without your knowledge. Use the Security Toolkit.
- If you are not already there, set the QSECURITY system value to 40 or higher. Please see the *AS/400 Security - Reference* manual before you make any changes. Changing the security level may require some preparations. Stay out of trouble.
- Improve the system values. The changes made to system values affects everyone on the system.

  Show Respect

  Before you make any changes, analyze the impact the change has on your users. Inform them about the change, the reason for the change, and the impact it will have.

- Get control of the user profiles. The authority checking is based on the authority given to a user profile. A user profile may get the authority needed to access an object through private authority, group authority, public authority, or adopted authority.
- Get control of ownership. This may be the most complicated task. Ownership is important, since you cannot stop an owner. You may, of course, revoke the owner's authority to owned objects, but if you choose this road, you probably end up in a messy situation.

  Note: Protect Yourself

  You should always consult your application's provider before you make any changes to object authority or ownership. Applications are not always as straight forward as you may think.

- Implement resource security using authorization lists, public authority, and so on.

Get started and good luck!

Chapter 3. Securing Your First Application - HTTP Server

This chapter explains what steps you should take to secure your IBM Internet Connection/400 HTTP server, hereafter referred to as the IC/400 HTTP server.

The primary purpose of the HTTP server is to provide access for visitors to a Web site on an AS/400 system.

When running a Web server on a multi-user system such an AS/400 system, the data that you serve through the HTTP server should be protected properly on the system the same as any other critical data.

The HTTP (Hypertext Transfer Protocol) server provides World Wide Web browsers (clients) with access to AS/400 multimedia objects such as HTML (Hypertext Markup Language) documents. You can also set up the HTTP server to allow browser clients to request functions that run programs on your AS/400 system. The IC/400 HTTP server uses the Common Gateway Interface (CGI) specification to call these programs.

The first step in designing a secure Web site is to define the AS/400 information that is accessible through the Web server. You must define which users have access to the Web server. Since the V3R7 implementation of the HTTP server does not provide for user authentication, it is considered an anonymous server. Access to the server and, thus, the Web documents is controlled by the network design.

3.1.1 Controlling Access to AS/400 Objects: HTTP Server Configuration

The primary purpose of the HTTP server is to provide access for visitors to a Web site on your AS/400 system.

You might think of someone who visits your Web site as someone who views an advertisement in a trade journal. The visitor is not aware of the characteristic of your Web site, nor do you want to put any barrier (such as a sign-on display) between a potential visitor and your Web site.

HTTP server requests cannot put, update, or delete data on your system. A client cannot do anything with the HTTP server until the server administrator (webmaster) defines directives for the server. To define directives, you must use the Work with HTTP Configuration (WRKHTTPCFG) command. This command requires *IOSYSCFG special authority.

In the following discussion, we assume that you are already familiar with the configuration of the HTTP server on the AS/400 system. For more information about how to configure your AS/400 system as a Web server, see the redbook *Cool Title About the AS/400 and Internet* and the *TCP/IP Configuration and Reference* manual.

Web servers on all platforms use directives to allow access to information on the host. The advantage of the AS/400 platform is the powerful security features of OS/400. All HTTP requests must satisfy two conditions:

- The object that is being requested must be allowed by the directives in the HTTP server configuration.

- The QTMHHTTP user profile must have authority to the object being requested.

The HTTP server fails, by default, all incoming requests *unless* the URL (as translated by any preceding *Map* directives) matches a *Pass, Redirect,* or *Exec* directive that has been explicitly coded by the HTTP server administrator.

- A match with a *Pass* directive statement enables the HTTP server to serve a document.
- A match with a *Redirect* directive statement causes the HTTP server to return a *Redirect Response* to the client application. No AS/400 data is accessed.
- A match with an *Exec* directive statement enables the HTTP server to execute a CGI program on behalf of the client.

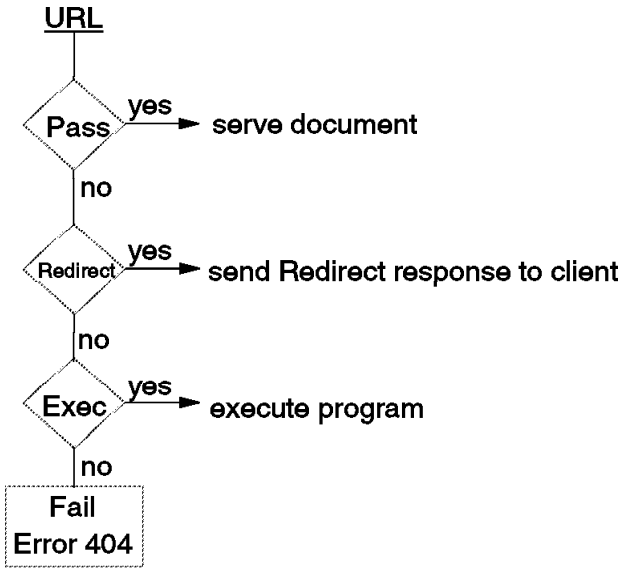


Figure 30. Configuration Directives that Control Access to HTTP Server Data

Note: The HTTP directives are checked in the order in which they appear in the file.

The HTTP server configuration file, as shipped, does *not* contain any *Pass, Redirect,* or *Exec* statements. When you add a directive to the HTTP server, make the template value for the path as specific as possible. This reduces the chance that someone can browse through your system and discover files. For instance, if documents that you want to serve reside in the /WWWDOC/HTML/PUBLIC directory, do not set a pass statement to a higher point in the directory structure than is necessary. A directive such as

```
Pass info/*  /WWWDOC/*
```

possibly allows access to more documents than you intended. Instead, design the pass directive so that it allows access at the lowest possible point in the directory structure. For example:

```
Pass info/*  /WWWDOC/HTML/PUBLIC/*
```

Use Map or Pass directives to mask the file names on your AS/400 Web server. Why let the potential intruder know the directory structure. For example, the client (browser) might issue a URL that looks similar to this:

```
http://hostname/web/products
```

Use the WRKHTTPCFG command to add a Pass directive to the HTTP configuration that looks similar to this:

```
Pass /web/* /QSYS.LIB/WEBDATA.LIB/WWWDATA.FILE/*
```

The requester has no idea that the product's data is in the WWWDATA file in the WEBDATA library. This method protects (hides) your AS/400 file names and library names from potential hackers. It also gives you the flexibility to change the location of your AS/400 data without changing the URL.

A FAIL statement can be used to block access to an object. Keep in mind that a FAIL statement in an HTTP configuration is case sensitive while most of the AS/400 file systems are case insensitive. The FAIL statement is actually not too useful. The default action (if the URL does not match any Pass or Exec statements) of the HTTP server is to fail. So, code your Pass and Exec statements carefully to allow only those operations in the directories as needed. Do not count on the FAIL directive primarily. For example, read the following three lines in an HTTP configuration:

```
Pass /QSYS.LIB/*
Fail /QSYS.LIB/PAYROLL.LIB/*
Fail /qsys.lib/payroll.lib/*
```

All a hacker has to do is issue the following URL, alternating in upper and lower case, and they are in the payroll library.

```
http://systemname/QSyS.lIb/PaYrOlL.lIb/*
```

So you see, this example works as long as a Pass statement inadvertently allows it. It is a better practice to carefully allow access to AS/400 objects on a selective basis rather than allowing access to all objects and blocking only selected objects. The most important thing to remember when creating mapping rules is that they are processed sequentially. If you create a rule and find that it is not working as expected, check that your request does not match some other directives earlier in the configuration file. The processing sequence for mapping directives is as follows:

1. The request string is compared against the template in the mapping directives. Comparison begins at the top of the configuration file and moves toward the bottom.
2. If a request string matches a Map template exactly, the resulting string replaces the original request string. The resulting string is used for all successive directives including map, pass, and exec.
3. If a request string matches a Map template with a wildcard, the part of the request that matches the wildcard is inserted in place of the wildcard in the result string. If the result string has no wildcard, it is used as it is. The result string is used as the request string for successive mapping directives.
4. If a request string matches Pass, Fail, Redirect, or Exec templates, the request is processed according to that directive. The request is not checked against any other mapping directive.

If the DirAccess directive is set to ON, any request for a directory that fails to discover a file returns a directory listing. It is recommended that you use a

directive to either disallow directory listings (DirAccess off) or limit them to specific sub-directories.

3.1.2 Serving HTML Pages (Read-Only Server)

The V3R7 implementation of the HTTP server supports only the GET, POST, and HEAD request methods. The significance of this from a security aspect is that the flow of information is strictly a one-way street. There is no capability for a user to write, delete, or alter in any way, information on the AS/400 system. This is a safe and reliable solution that provides the benefit of using Internet technologies to share information with no exposure to unauthorized alteration from a Web browser. Physical or internal security measures must be circumvented to compromise the integrity of your information.

3.1.3 Using Net.Data to Develop Your Web Applications

Net.Data is an application that runs on the IC/400 HTTP server. It allows you to easily create dynamic documents. For detailed information on Net.Data, visit the IBM Web site at:

`http://www.as400.ibm.com/netdata`

The significance of Net.Data from a security point of view is that we now introduce the ability to create and alter information on the AS/400 system from a Web browser. Although Net.Data does not provide any type of security measures directly, you can keep your assets secure with the existing measures you should already use to protect your system and data. Access control is achieved through network design and the application itself.

The benefit of using Net.Data is the ability to quickly develop Web applications with minimal experience or programming skills. The security risks as compared to just serving HTML pages are greater because of the added capability, but with thoughtful planning and control, a secure solution can be implemented.

3.1.4 Using CGI Programs to Develop Your Web Applications

Custom CGI solutions are more powerful and flexible than Net.Data but add the exposure of improperly functioning programs. The security administrator must ensure that the CGI program concurs with the security policy and does not allow unauthorized access to data.

3.2 Tips and Techniques

This section provides some tips and techniques on how to configure and use the IC/400 HTTP server for AS/400 functions in V3R7 and V3R2 to minimize security exposures.

3.2.1 Objects Related to HTTP Server Security

The AS/400 objects that affect the security of the HTTP server are:

- QTMHHTTP -- user profile for server
- QTMHHTP1 -- user profile for CGI programs
- QUSRSYS/QATMHTTPC -- HTTP configuration file
- QUSRSYS/QATMHTTP -- HTTP server attributes file
- HTML documents and IFS directories, folders, or files where they reside
- CGI-BIN programs and libraries where they reside
- Net.Data macros

- Object authority for AS/400 objects to be accessed by the server

3.2.1.1 QTMHHTTP User Profile

The Web server runs under the authority of the QTMHHTTP user profile. The QTMHHTTP user profile or *PUBLIC must have *USE authority to all AS/400 library system objects that you intend to serve. The QTMHHTTP user profile must have *RX authority to all QDLS and integrated file system (IFS) objects that you intend to serve. If the documents are served from the QDLS file system (folders), the profile must also have a directory entry added using the ADDDIRE command.

Tip

The HTTP server can access objects that have the proper level of *PUBLIC authority. In this case, access to those objects from the Internet are controlled only by the server directives and do not have the added level of protection provided by object authority.

If you want to give some level of *PUBLIC authority to an object to be used by local users, but you do not want the object to be accessed through the HTTP server, set QTMHHTP authority to *EXCLUDE for that object.

The QTMHHTTP profile should:

- Have a password value of *NONE.
- Have user class = *USER.
- Not be member of a group profile.
- Not be used as a group profile.
- Not have special authorities.

3.2.1.2 QTMHHTP1 User Profile

The server runs CGI programs and Net.data under the QTMHHTP1 user profile. This user profile should only have access to the programs and data files that it needs to create the dynamic HTML or to process forms. The QTMHHTP1 profile should:

- Have a password value of *NONE.
- Have user class = *USER.
- Not be member of a group profile.
- Not be used as a group profile.
- Not have special authorities.

See Figure 31 on page 62 for an illustration of the relationship between profiles and objects in the IFS.

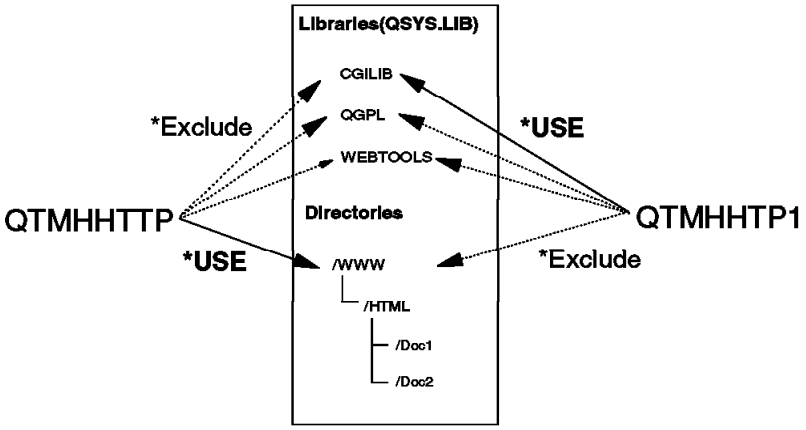


Figure 31. Recommended Object Authority for QTMHTTP and QTMHHTP1

3.2.1.3 IC/400 HTTP Server Configuration File

In addition to OS/400 object authority, the IC/400 HTTP server configuration file is used to control what information can be accessed by the server. Directives in the HTTP server configuration file are used to define which URLs are to be accepted and where the corresponding data is located. See Figure 32 on page 63 for an illustration of the relationship between a URL request, directives, and IFS objects.

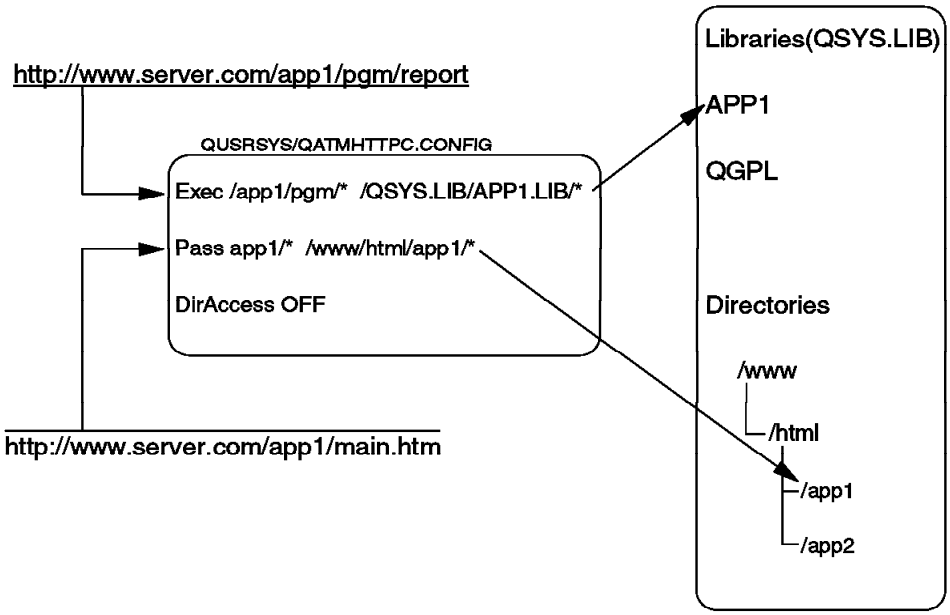


Figure 32. HTTP Server Directives

The HTTP server configuration file is QUSRSYS/QATMHTTPC.CONFIG. The configuration file that is shipped by IBM does not allow access to any data or any program on your system.

Note

The HTTP server does **not** do anything that the server administrator (webmaster) has not explicitly configured it to do.

Use the Map, Pass, and Exec directives to control access to your data and programs.

The HTTP server configuration file is important to control the Web site. Make sure that only the webmaster can change the HTTP server directory. Use the EDTOBJAUT OBJ(QUSRSYS/QATMHTTPC) OBJTYPE(*FILE) command shown in Figure 33 on page 64 to verify that only the webmaster can update the HTTP server configuration.

Do remember that this does not stop any user with *ALLOBJ and *IOSYSCFG special authority from updating the HTTP server configuration.

```
                          Edit Object Authority

Object . . . . . :   QATMHTTPC       Owner  . . . . . :   QSYS
  Library  . . . . :     QUSRSYS       Primary group  . . :   *NONE
Object type  . . . :   *FILE

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . .   *NONE

                         Object    ---------------Data---------------
User        Group        Authority  Read  Add  Update  Delete  Execute
QSYS                     *USE        X                           X
WEBMASTER                *CHANGE     X     X     X       X       X
*PUBLIC                  *EXCLUDE

                                                                   Bottom
F3=Exit   F5=Refresh    F6=Add new users   F10=Grant with reference object
F11=Nondisplay detail   F12=Cancel         F17=Top   F18=Bottom
```

Figure 33. HTTP Server Configuration File Authority

3.2.1.4 HTTP Server Attributes File

The HTTP server attributes are changed using the CHGHTTPA command.

```
                      Change HTTP Attributes (CHGHTTPA)

Type choices, press Enter.

Autostart  . . . . . .   *YES          *YES, *NO
Number of server jobs:
  Minimum  . . . . . .   3            1-200, *S
  Maximum  . . . . . .   5            1-200, *S
Coded character set identifier    00819        1-65533,
Server mapping tables:
  Outgoing EBCDIC/ASCII table  .   *CCSID       Name, *SA
    Library  . . . . . .                       Name, *LI

  Incoming ASCII/EBCDIC table  .   *CCSID       Name, *SA
    Library  . . . . . .                       Name, *LI
```

Figure 34. HTTP Server Attribute File

For more information, see the *TCP/IP Configuration and Reference* manual.

3.2.1.5 Web Administrator User Profile

Although not an integral part of the HTTP server, an important role in the creation and operation of a Web site is the person commonly referred to as the webmaster. The user profile for this individual must be carefully constructed to allow management of the Web site without compromising I/T security policy in other areas. For instance, the webmaster is probably not the same person who creates user profiles and sets object authority for sensitive data.

The server administrator's user profile is called webmaster in this example. This is valuable information for an intruder. We recommend that you name your server administrator user profile differently.

3.2.2 Securing HTML Documents

The HTML documents represent your company's image on the Web. HTML documents are stored in folders, in IFS directories, or in source physical file members. You must secure them; you cannot afford unauthorized changes to these documents. The biggest exposure probably comes from sabotage within the company. Limit (or not allow) Client Access/400™ access to the server by local users. Tightly control the security of the libraries, folders, and IFS directories where the HTML documents reside.

Group all of the HTML documents to be served by the server under the same directory tree and do not mix them with other documents you do not want to serve.

Avoid starting FTP server on the same system where the HTTP server is running.

You should always validate HTML form input in your programs. HTML forms can be saved locally, changed, and sent back to the server containing unexpected data. If your form includes a list of items to select from, or hidden fields, the user can easily change the pre-configured values. Even Java™ Script field auditing can be disabled at the browser!

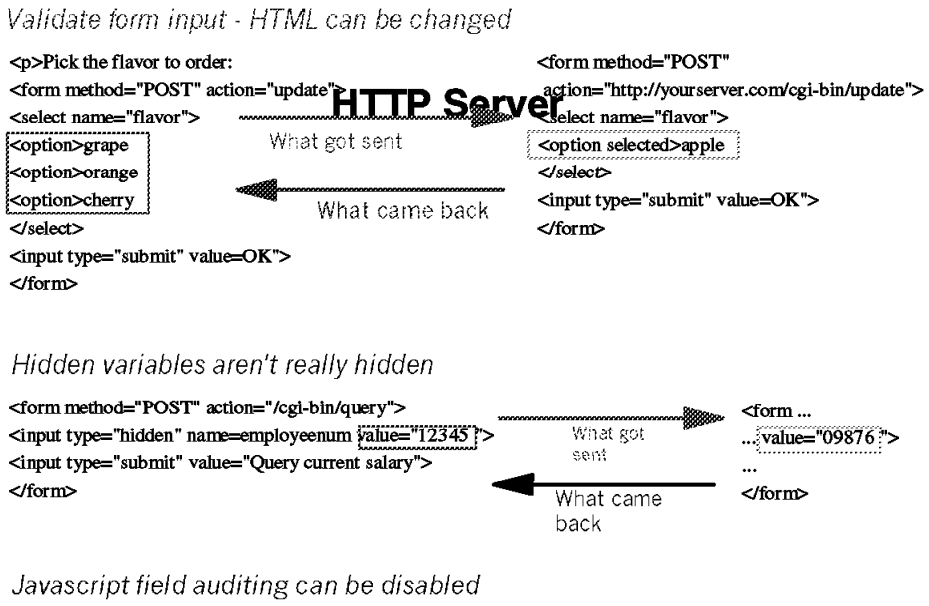


Figure 35. Always Validate HTML Input

3.2.3 Common Gateway Interface (CGI) Security Considerations

To execute a CGI program on the AS/400 system, you must use the EXEC directive in your HTTP configuration file. The EXEC directive allows your CGI program to run if the request string matches the URL template. CGI programs should be written with the same care and attention given to Internet servers themselves, because, in fact, they are miniature servers. All CGI programs run under the user profile QTMHHTP1.

As webmaster, you should monitor the authorizations of the QTMHHTP1 user profile and the functions of the programs. You should never import CGI programs from some unchecked source just because they fit your current needs. Make sure you understand them completely and all of their security implications before using them. You also need to know all of the details of programs called **from** a CGI program.

Some important points to keep in mind:

- The user profile QTMHHTP1 should **only** have access to programs that it needs to run and data files that it needs to access.
- If you have objects with some degree of *PUBLIC authority, you can limit access to these objects from the Internet by setting QTMHHTP1 authority to *EXCLUDE.
- All CGI programs should be stored in a separate Library (CGILIB, for example) and separated from documents. See Figure 31 on page 62 for an example of server directories and libraries structure and server profiles authorities.
- Set both object and data authority to control who can put new objects in the library (CGILIB) and who can run programs in this library.

Due to the ability of most browsers to display HTML source, extra care must be taken when developing CGI programs. A user may make a copy of the HTML form and modify it to send data that your CGI program does not expect. If your CGI program does not exactly validate form input, it may not operate correctly or it may have security exposures. Programs that process forms that contain list boxes should not assume that form input contains one of the valid options. The original form can be copied and modified to send an invalid option.

Another area of concern is hidden variables. If a user displays the HTML source, they are no longer hidden. For example, HTML source can be modified to query information for a different employee number than was intended by the creator of the form.

Java Script gives form developers new capabilities for field editing. Java Script is valuable because it provides immediate feedback to the user when an input value is not correct. However, Java Script can be disabled at the browser. Therefore, the field values must be revalidated by the CGI program to ensure accuracy.

When you use CGI programs on your Web server, you must completely understand the functions of the CGI or Net.Data program and the functions of programs called from a CGI or Net.Data program. It might be a big challenge to check such a CGI program or an entire CGI application, but this is important to protect your Web server and your data.

There are three major areas you should be concerned with when running CGI or Net.Data applications:

- Calls to external functions or programs:

  Scan the programs for external program calls and look for weaknesses in called programs.

- Database files accessed by your applications:

  Find out which database files are accessed from the programs and how the program handles the data.

- Input and output data:

  The CGI programs use the Post or Get method to get a data stream from the Web server. You must analyze how the program parses this data stream and builds the input parameter, and how the program handles unexpected data streams.

3.2.3.1 Under Which User Profile Does Your CGI Program Run?

The server runs the CGI program under QTMHHTP1 user profile. The QTMHHTP1 profile can only access objects with *PUBLIC authority set to *USE or higher. If you do not want to give *PUBLIC the necessary authority to some CGI programs or data because you are concerned that local users might get access to them and use them in a way that you did not intend, you must provide the QTMHHTP1 user profile with the authority to access the objects.

If, on the other hand, you want to prevent some sensitive data from being accessed by CGI programs through the IC/400 HTTP server, you can exclude the QTMHHTP1 user profile from calling certain CGI programs or accessing certain data by specifying that QTMHHTP1 has *EXCLUDE authority.

Note

As usual, we are applying a layer approach to protecting your data. If you do not want some AS/400 libraries and objects to be accessed through the Internet, you first configure the HTTP server with the appropriate directives as discussed in Section 3.1.1, “Controlling Access to AS/400 Objects: HTTP Server Configuration” on page 57. Then you add a second layer of protection, setting up object authority appropriately as discussed in this section.

Figure 36 on page 68 shows that the CGI-app1 program created with the User profile (USRPRF) parameter set to *USER (specifies the user profile under which the program runs) runs under the QTMHHTP1 user profile. QTMHHTP1 or *PUBLIC needs *USE authority to the CGI program and also read and write to the PUBLIC data file. We want to protect the sensitive data in the PAYROLL file from being accessed through the Web server, so we specifically excluded QTMHHTP1 from it. If the CGI program attempts to read data from PAYROLL, the Read operation will fail.

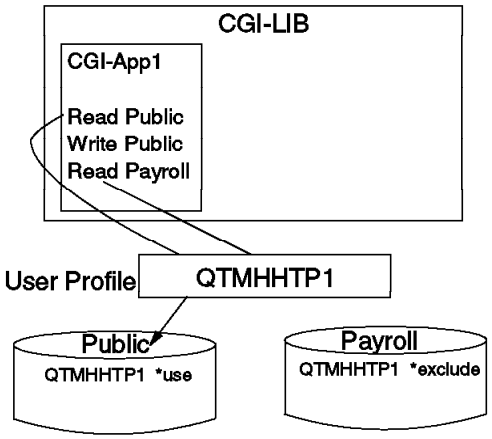


Figure 36. Preventing QTMHHTP1 from Accessing the PAYROLL File

Now, let's assume that you want to allow access to the PAYROLL file through the IC/400 HTTP server but **only** through the CGI-App2 program and **no** other one. To allow the server to run CGI-App2 and access PAYROLL, QTMHHTP1 must have the proper authority to both program and data. Everything is fine if you have "well-behaved" programs **and** programmers that only access the data the way you intended. For example, we do **not** want any other program but CGI-App2 to access PAYROLL. But now we have the exposure that, because QTMHHTP1 has authority to the file, a malicious programmer, an accident, or something else can misuse the PAYROLL data. Figure 37 shows that even when we intended for **only** CGI-App2 to access PAYROLL, we have opened a hole that enables a "not-so-well-behaved-program" such as CGI-App1 to read PAYROLL data.

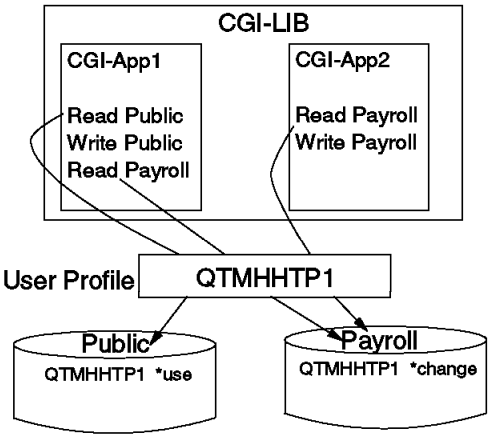


Figure 37. Exposures of Running All CGI Programs under QTMHHTP1 User Profile

Figure 38 on page 69 shows CGI-App2 running under BOSS user profile (USRPRF parameter set to *OWNER in CRTPGM or CHGPGM commands). BOSS has the necessary authority to access PAYROLL except *PUBLIC and, therefore, QTMHHTP1 is *EXCLUDE. This approach allows for more granularity and control. Now PAYROLL data can be accessed through the IC/400 HTTP server, but **only** through CGI-App2 because that program adopts its owner's authority.

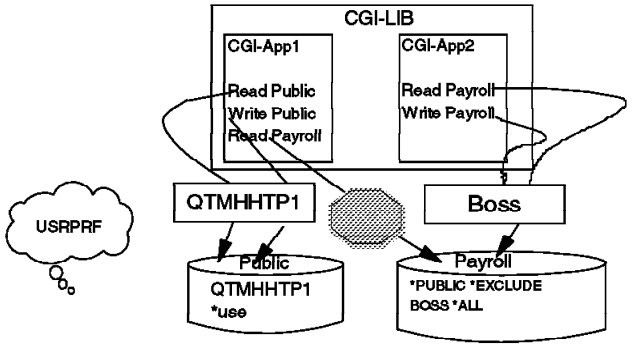


Figure 38. Limiting Access of CGI programs to Specific User Profile

3.2.4 Net.Data Security Considerations

Net.Data replaces DB2WWW Connection Version 1. Net.Data is composed of a program, the **Web macro processor**, and one or more dynamic libraries called **language environments**. The executable input to Net.Data is the **Web macro**. Net.Data is a cgi-bin program (DB2WWW), many of the same considerations apply.

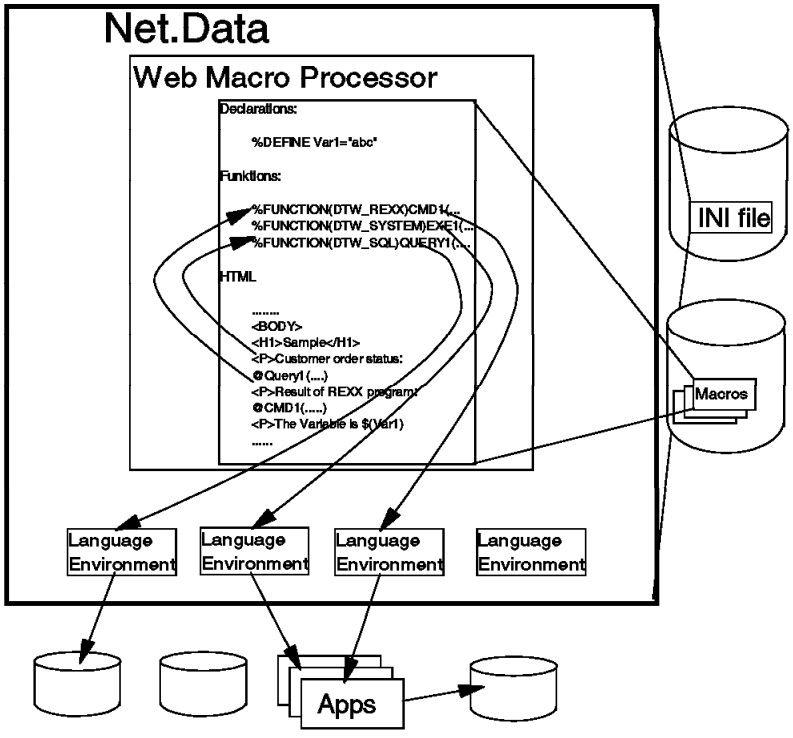


Figure 39. HTTP Server Directives

Although Net.Data does not provide any type of security measures directly, you can keep your assets secure with the existing measures you should already be

using to protect your system and data. In fact, the risk is not the Net.Data environment, but the functionality of the CGI program itself.

IBM recommends that the HTTP administrator move both QTMHIMAG *PGM and DB2WWW *PGM (the Net.Data CGI program) from the QTCP library to their own CGI library (for example, CGILIB).

The IBM manual (*OS/400 TCP/IP Configuration and Reference*) mentions that IBM recommends that the HTTP administrator move both the QTMHIMAG.PGM and DB2WWW.PGM to your own CGI library to avoid the security problem of allowing all programs in QTCP.LIB to potentially be executed (or called) by the HTTP server. This is prudent advice because no one is really sure what damage a vandal can cause by calling FTP through the HTTP server.

The problem is that after you have moved QTMHIMAG.PGM and DB2WWW.PGM to your own CGI library, any PTF updates to your software are not automatically made because it is the original program in QTCP.LIB that has been updated. After PTFs are applied, copy the programs once more.

Because Net.Data uses the CGI interface, the same security measures as described in Section 3.2.3, “Common Gateway Interface (CGI) Security Considerations” on page 66 should be considered.

To execute a Net.Data macro, you can use the Map and Exec directive in the HTTP configuration file to allow access to the macro and allow execution of the Net.Data program. The Map directive hides the file system structure by using an alias for the real library and file name. The Exec directive allows the CGI program to be run. Here is an example of how the directives might be constructed:

```
Map  /cgi-bin/db2www/* /QSYS.LIB/CGILIB.LIB/DB2WWW.PGM/*
Exec /QSYS.LIB/CGILIB.LIB/*
```

Important!

Net.Data uses macros that are executed by the Macro processor, similar to CGI programs. To ensure that you comply with the security policy, you have to understand the functions of the macros and the functions of the programs called from the macros.

Net.Data is designed to allow new language and database interfaces to be added in a ″pluggable″ fashion. These language environments are accessed as service programs. The name of the service program is configured in the Net.Data initialization file (the INI file) and associated with a language environment name. This INI file must be located in the CGI library where the DB2WWW program object resides. Only the webmaster should be authorized to change the Net.Data INI file. For further information, see the Internet Page:

`http://www.as400.ibm.com/netdata`

3.3 Implementation Examples

We illustrate how to implement two different solutions. The first solution is a read-only server that can be used for advertising or information publishing. The second solution uses a CGI program to set up a reader comment application.

3.3.1 Example 1: Serving HTML Pages (HTTP Read-Only Server)

This section shows how to set up the IC/400 HTTP server as a read-only server. This type of configuration is typically used to make information generally available over the Internet. It makes no provision for users with Web browsers to input information on your Web site. A typical application is advertising a product or providing investor information about your company. The following browser window is an example.

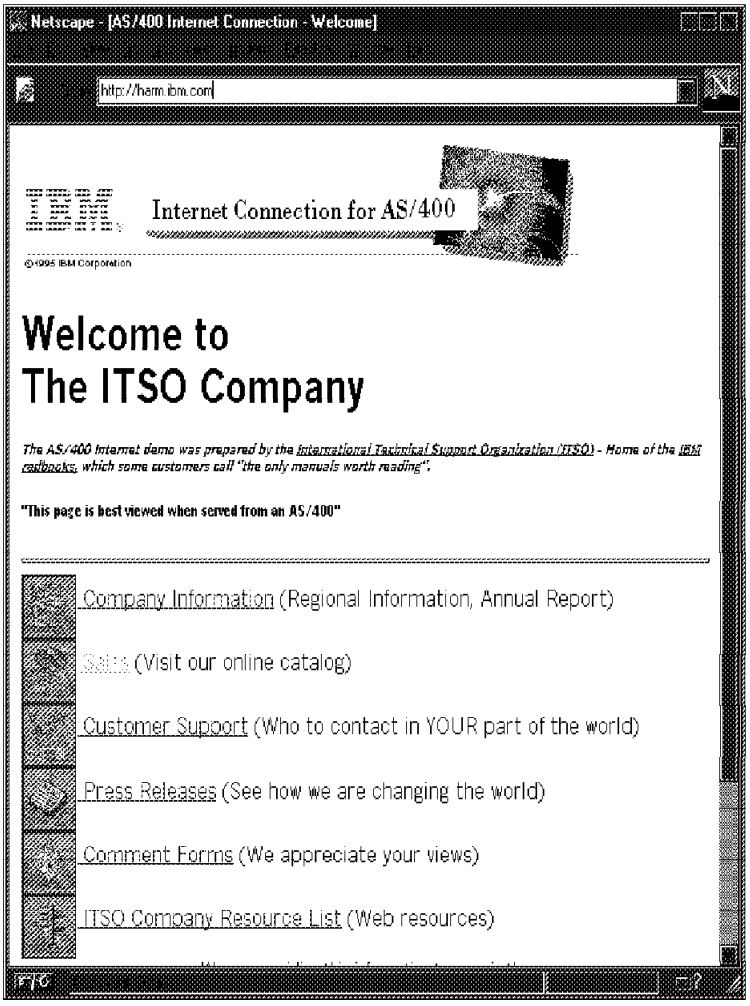


Figure 40. Welcome Page

Use the AS/400 security functions as described in Chapter 2, “Start Here by Securing OS/400®” on page 37 to meet your security policy by considering the following points:

- System security level
- Password rules

- Other security system values
- User profiles
- Resource security
- TCP/IP security

Assuming you have secured your AS/400 system, the special considerations that apply to this HTTP server example are:

- QTMHHTTP (Web server user profile)
- Web server configuration file (QUSRSYS/QATMHTTPC.CONFIG)
- Web document object authority
- Webmaster profile (optional Web server administrator)

3.3.1.1 Web Server User Profile

The Web server profile QTMHHTTP must have *USE authority to the library objects and *RX authority to the stream files that it serves as Web documents. This authority can be either private or public. In this example (read-only HTTP server), we are giving *PUBLIC Read/Execute (*RX authority) and no private authorities to QTMHHTTP.

The QTMHHTTP user profile should have PASSWORD *NONE to prevent a hacker from trying to use this profile to sign on to the system. See Section 3.4, “Logging and Auditing” on page 81 for further information on ensuring that this profile is not altered.

3.3.1.2 Web Server Configuration File

Use the Work with HTTP Configuration (WRKHTTPCFG) command to display and change the Web server configuration entries. Special authority *IOSYSCFG is required to use the WRKHTTPCFG command to add, change, or delete.

In this Read-Only example, no CGI program can be called because no *Exec* directive is specified, and only the Get method is Enabled. Figure 41 shows a portion of the HTTP configuration file.

```
                        Work with HTTP Configuration
                                                            System:   RC
Type options, press Enter.
  1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

      Sequence
Opt    Number    Entry

       00010     # * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       00020     # HTTP CONFIGURATION
       00030     # * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       00040     AccessLog ACCESSFILE
       00050     ErrorLog ERRORFILE
       00060     LogFormat common
       00070     HostName  my.host.name
       00080     AlwaysWelcome        On
       00090     Welcome welcome.html
       00100     Enable                          GET
       00110     Disable                         POST
       00120     Disable                         HEAD
                                                                    More...
F3=Exit   F5=Refresh   F6=Print List   F12=Cancel   F17=Top   F18=Bottom
F19=Edit Sequence
```

Figure 41. Enable Only Get Method (Disable Post and Head)

To prevent the server from returning a directory list if a request from a Web browser fails, the DirAccess directive must be set to *Off* as shown in Figure 42 on page 73.

```
                          Work with HTTP Configuration
                                                                 System:   RC
Type options, press Enter.
  1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

      Sequence
Opt    Number    Entry

       00130     DirAccess  Off
       00140     DirReadme Off
       00150     DirShowDescription Off
       00160     DirShowMaxDescrLength  20
       00170     DirShowOwner Off
       00180     DirShowDate Off
       00190     DirShowSize Off
       00200     DirShowBytes Off
       00210     #
       00220     #*****************************************************#
       00230     # This is the ITSO Company application (read only)
       00240     #*****************************************************#
                                                                        More...
F3=Exit   F5=Refresh   F6=Print List   F12=Cancel   F17=Top   F18=Bottom
F19=Edit Sequence
```

Figure 42. Set DirAccess to Off

The Pass directives in Figure 43 enable the Web server to serve only documents from the directory /WWWHARM/ITSOIC.400. The Pass statement

```
Pass/ /WWWHARM/ITSOIC.400/*
```

is used to serve the home page *welcome.html* from the /WWWHARM/ITSOIC.400 directory.

```
                          Work with HTTP Configuration
                                                                 System:   RC
Type options, press Enter.
  1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

      Sequence
Opt    Number    Entry

       00250     Map  /itso/* /WWWHARM/ITSOIC.400/*
       00260     Map  /ITSO/* /WWWHARM/ITSOIC.400/*
       00270     Pass /WWWHARM/ITSOIC.400/*
       00280     Pass / /WWWHARM/ITSOIC.400/*
       00290     #*****************************************************#
```

Figure 43. Map and Pass Directives

The following directives are what allow and cause the home page to be sent to the browser:

```
Enable          Get
AlwaysWelcome   On
Welcome         welcome.html
DirAccess       Off
Pass            / /WWWHARM/ITSOIC.400/*
```

The browser uses the URL:

```
http://hostname
```

(with or without trailing '/') and receives the page as shown in Figure 40 on page 71.

The following complete listing shows the configuration values for this example.

```
 HTTP Configuration Entries
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * #
# HTTP CONFIGURATION                                                       #
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * #
AccessLog ACCESSFILE
ErrorLog ERRORFILE
LogFormat DDS
HostName  my.host.name
AlwaysWelcome        On
Welcome welcome.html
Enable                          GET
Disable                         POST
Disable                         HEAD
DirAccess  Off
DirReadme Off
DirShowDescription Off
DirShowMaxDescrLength  20
DirShowOwner Off
DirShowDate Off
DirShowSize Off
DirShowBytes Off
#
#*****************************************************#
# This is the ITSO Company application (read only)
#*****************************************************#
#HTTP Configuration Entries
Map  /itso/* /WWWHARM/ITSOIC.400/*
Map  /ITSO/* /WWWHARM/ITSOIC.400/*
Pass /WWWHARM/ITSOIC.400/*
Pass / /WWWHARM/ITSOIC.400/*
* * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 44. HTTP Configuration File - Read-Only Server

3.3.1.3 Web Document Object Authority

A decision must be made on how to set object authority for HTML documents to be served by the HTTP server. As mentioned earlier, QTMHHTTP needs Read/Execute access to those documents. A good balance of performance and ease of maintenance can be achieved by creating directories with the following default authorities shown in Figure 45.

```
                          Create Directory (CRTDIR)

Type choices, press Enter.

Directory  . . . . . .  harm

Public authority for data  . . .  *RX          Name, *INDIR, *RWX, *RW
Public authority for object  . .  *NONE        *INDIR, *NONE, *ALL...
              + for more values
Auditing value for objects . . .  *SYSVAL      *SYSVAL, *NONE, *USRPRF
```

Figure 45. Object Authority for the HTML Documents Directories

This automatically sets authority for new objects added to the directory. Assuming the user WEBMASTER creates a new stream file into the /HARM/WEBDOCS/PUBLIC directory, the object authority for the document is set as shown in Figure 46 on page 75.

```
Object . . . . . :   /harm/webdocs/public/pubhome.htm
Owner  . . . . . :   WEBMASTER
Primary group  . . . . . :   *NONE
Authorization list . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                  Data     --Object Authorities--
   Opt  User       Authority  Exist  Mgt  Alter  Ref

     *PUBLIC     *RX
     WEBMASTER   *RWX            X     X     X     X
```

Figure 46. Object Authority for Documents Accessed by Read-Only Server

This configuration allows the user profile WEBMASTER to maintain HTML documents using Client Access/400. At the same time, the Web documents are secured from change by anyone except WEBMASTER and those with *ALLOBJ authority.

The only exposure to this scheme is the fact that all AS/400 workstation users who have direct local access to the IFS can learn the owner of the object. Knowing the owner is an advantage when trying to break security.

This configuration allows all AS/400 workstation users, and also Client Access/400 users, to open and copy documents since *PUBLIC authority is *USE. If you do not feel comfortable with this configuration, another scheme is to set *PUBLIC *EXCLUDE, the server profile QTMHHTTP to *RX, and the WEBMASTER to *RWX.

3.3.1.4 Webmaster Profile

It is recommended that only one user (or few users) should have the authority to change the HTTP server configuration file. This user (webmaster) needs the special authority **IOSYSCFG*.

3.3.2 Example 2: HTTP Read/Write Server

The next step in your Internet business strategy may require more flexibility and functionality. You might need to access information from your AS/400 database or you might want to update your AS/400 database with input from the clients. You can do this by using CGI programs or Net.Data.

This section shows how to extend the Read-Only Web server to a Read/Write Web server. This example is actually an extension of the previous example. The security related objects in this example are:

- QTMHHTTP - user profile used for serving HTML documents
- QTMHHTP1 - user profile used for calling CGI programs
- QATMHTTPC - server configuration file
- CGI programs
- Objects used by the CGI programs
- Web administrator user profile

3.3.2.1 Example 2 Overview

From the Welcome page in the previous example, there is a link to the comment page shown in Figure 47. This provides a way for users to input personal information and comments to the Web server. The users type in their information in the input fields and click on ″Submit Form″. The submit starts the RPG Program (IMVR501) on the AS/400 system. This program writes the information in the FEEDB file and sends information back to the browser.

The browser windows for this example are shown in Figure 47 and Figure 48 on page 77.

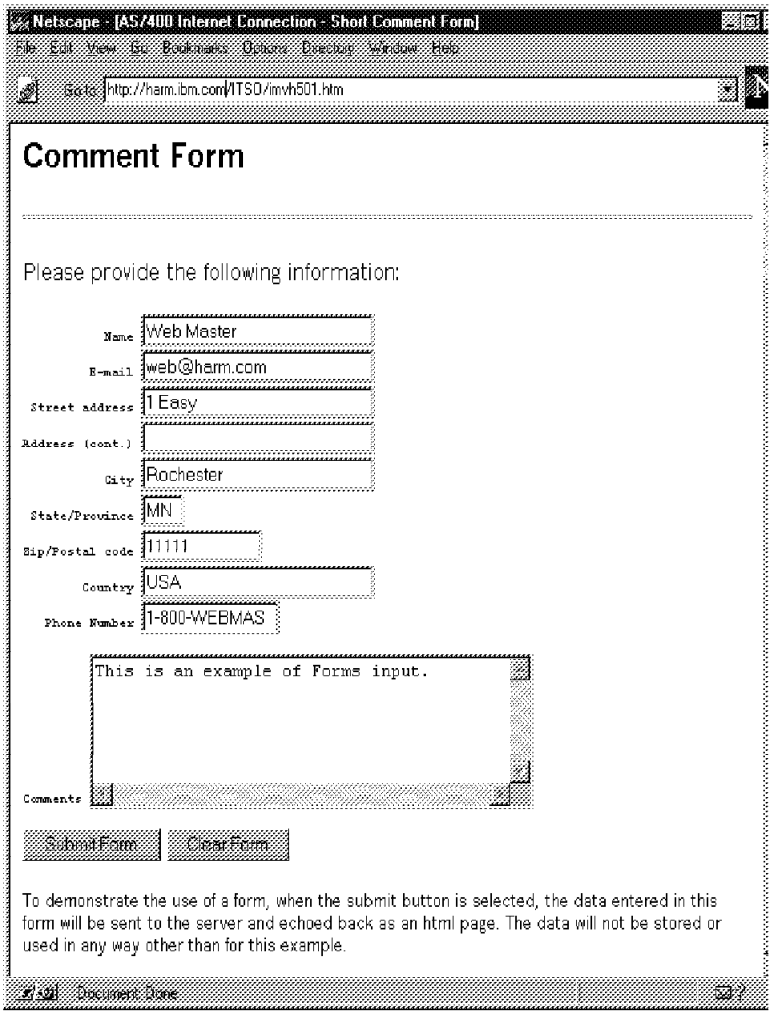


Figure 47. Feedback Comment Input Form

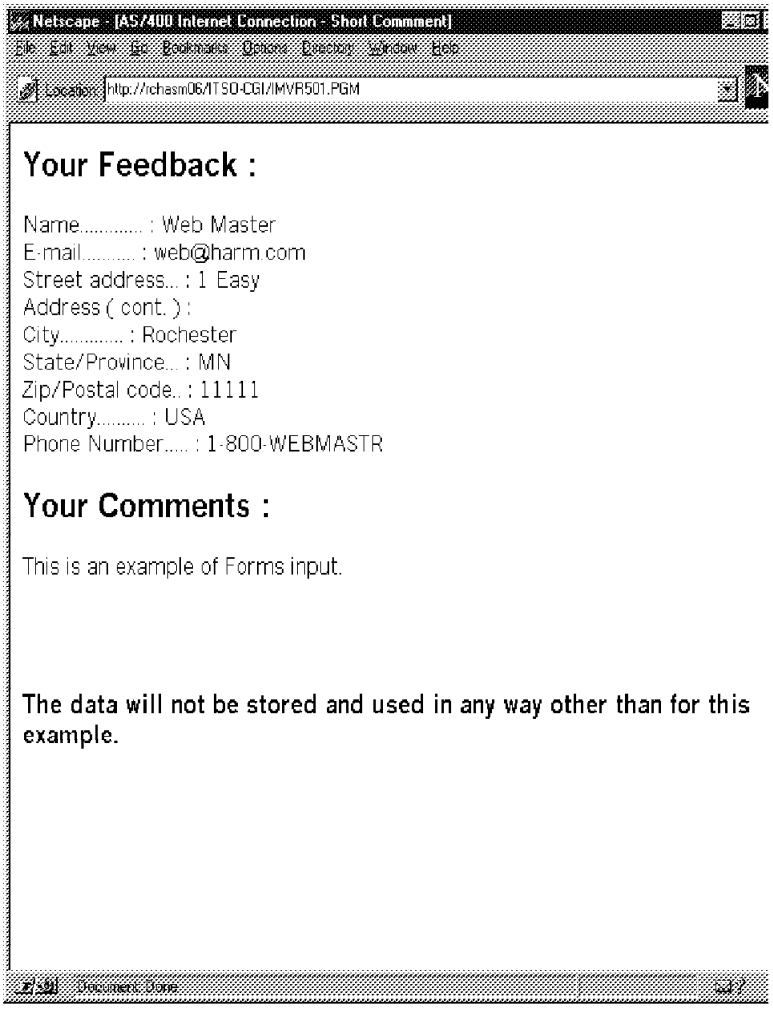


Figure 48. Feedback Confirmation Page

Since this is a follow-on to the previous example, we cover only the points that are unique to this sample CGI application. Please refer to the discussion in Section 3.3.1, “Example 1: Serving HTML Pages (HTTP Read-Only Server)” on page 71 as to how the server user profile QTMHHTTP and the administrator authorities should be configured. As always, try to achieve a good balance between security, availability, and useability. *And keep it simple*.

3.3.2.2 QTMHHTP1 User Profile

When the HTTP server receives a request from a browser to call a CGI program, it switches from running under the QTMHHTTP profile to the QTMHHTP1 profile. This user profile is properly configured when you first install the IC/400 HTTP server utilities.

3.3.2.3 Securing CGI Program Objects

In our example, we followed the recommendations in the *TCP/IP Configuration and Reference*. We created a separate CGI library and set authorities as shown in Figure 49 on page 78. QTMHHTP1 is granted authority to our CGI library, the CGI program, and the file that the CGI program uses.

```
                          Change Authority (CHGAUT)

Type choices, press Enter.

Object . . . . . . . . . . . . .   /QSYS.LIB/ITSOIC400/

User . . . . . . . . . . . . . .   QTMHHTP1      Name, *PUBLIC, *NTWIRF
               + for more values
New data authorities . . . . . .   *RX           *SAME, *NONE, *RWX, *RX
New object authorities . . . . .   *SAME         *SAME, *NONE, *ALL...
               + for more values
Authorization list . . . . . . .                Name, *NONE
```

Figure 49. CGI Library Authority

```
                            Edit Object Authority

Object . . . . . . . :   IMVR501         Owner  . . . . . . . :WEBMASTER
  Library  . . . . . :     ITSOIC400     Primary group  . . . :   *NONE
Object type  . . . . :   *PGM

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . . . . . . . .   *NONE

                          Object
User        Group       Authority
WEBMASTER               *ALL
QTMHHTP1                *USE
*PUBLIC                 *EXCLUDE
```

*Figure 50. QTMHHTP1 *USE Authority to CGI Programs*

3.3.2.4 Securing Programs and Objects Used by CGI Programs

The simplest way to set authority for objects that are used by the CGI program is to give QTMHHTP1 the necessary level of authority to those programs and objects. In our example, the CGI programs need to update a file called FEEDB. Therefore, we grant the proper authority as shown in Figure 51.

```
                            Edit Object Authority

Object . . . . . . . :   FEEDB           Owner  . . . . . . . :   DON
  Library  . . . . . :     ITSOIC400     Primary group  . . . :   *NONE
Object type  . . . . :   *FILE

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . . . . . . . .   *NONE

                          Object
User        Group       Authority
QTMHHTP1                *CHANGE
*PUBLIC                 *EXCLUDE
```

*Figure 51. Grant QTMHHTP1 *CHANGE Authority to Data File*

Figure 51 shows that the owner of the file is DON. Do not allow any IBM-supplied user profile to own any NON-IBM supplied objects on the system. If QPGMR, for example, were the owner of this file, and it is also used as a

group profile, any member of the group has QPGMR′s authority to the file FEEDB.

This same scheme should be used for any program or data object for which the CGI program needs access. If you do not want all CGI programs to have equal access to other objects, a different scheme is needed. Refer to Section 3.2.3.1, “Under Which User Profile Does Your CGI Program Run?” on page 67 for a general discussion about what user profile you want your CGI program to run under.

When a CGI program is launched by the IC/400 HTTP server, the server switches from running under the QTMHHTTP user profile to the QTMHHTP1 profile. The CGI program probably needs access to other programs and data files. Access to those objects is controlled differently depending on the USRPRF parameter that is set when the CGI program is created. Setting the program USRPRF parameter to *USER requires QTMHHTP1 to have authority to the called program or data file that it requires. Setting the USRPRF parameter to *OWNER requires the owner of the CGI program to have authority to the called program or data file.

See the flowchart (Figure 52 on page 80) for an illustration of the logic. In this case, the CGI program that is running under the authority of QTMHHTP1 needs to open and write to a file named FILE1. Notice that at the second decision block, the USRPRF parameter determines which user profile authority is checked. In this example, if the USRPRF is *USER, authority to FILE1 is checked to see if QTMHHTP1 can write to the file. Specifically, does QTMHHTP1 have *CHANGE level authority? If the USRPRF is *OWNER, DON (the owner of the program) must have *CHANGE authority. If neither of these conditions are met, the operation will fail. If either of these conditions are true, the CGI program proceeds to open FILE1 and completes the write operation.

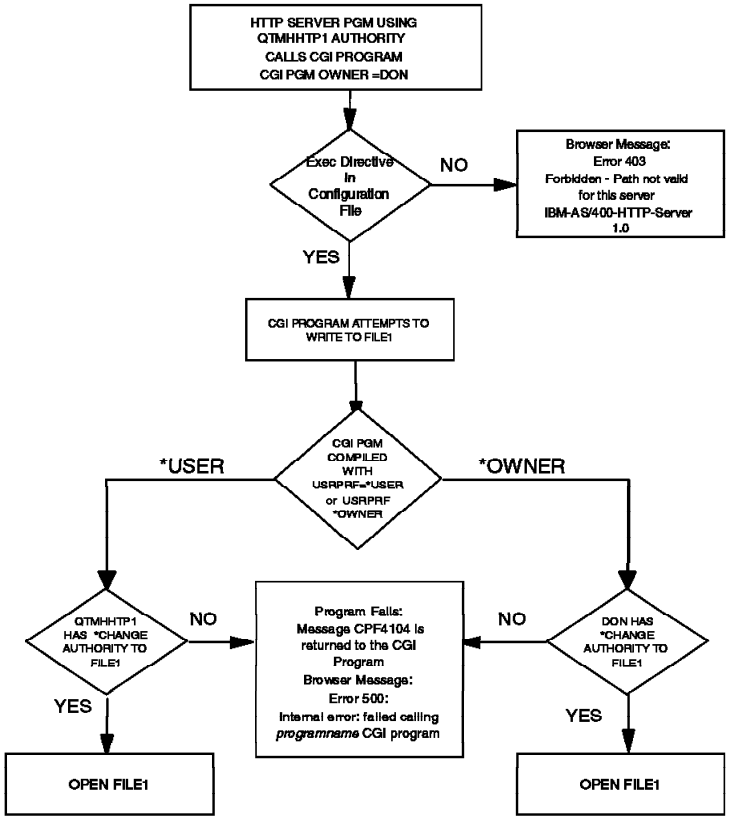


Figure 52. CGI Program Authority Checking

After the EXEC directive is checked, the web server makes sure QTMHHTP1 is authorized to the program object.

3.3.2.5 HTTP Directives to Control the Scope of Access

In addition to the Pass and Fail directives discussed in the previous example, the post method needs to be enabled and Map and Exec directives must be set in the HTTP configuration file to allow the CGI program to execute (see Figure 53).

```
                        Work with HTTP Configuration
                                                              System:   RC
Type options, press Enter.
  1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

      Sequence
Opt    Number    Entry
       00120     Enable                          POST
       00010     # * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       00020     # These two lines Map requests to ITSO-CGI to
                 # the ITSOIC400 library and allow program execution
       00030     # * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       00360     Map  /ITSO-CGI/* /QSYS.LIB/ITSOIC400.LIB/*
       00370     Exec  /QSYS.LIB/ITSOIC400.LIB/*
```

Figure 53. Additional Directives in HTTP Configuration File for Read/Write Server

3.4 Logging and Auditing

For security to be affective, the security controls must be monitored regularly. For information on how to audit your AS/400 system, refer to *AS/400 Security - Reference*. In this section, we focus on verifying that the recommendations included in this chapter to secure your HTTP server are being followed.

Use the AS/400 Security Audit functions to look for unplanned changes. To enable Security Audit Journal, use the following commands:

1. Create journal receiver:

   ```
   CRTJRNRCV JRNRCV(QGPL/AUDRCV0001) THRESHOLD(5000) TEXT('Security Journal
   Receiver')
   ```

2. Create audit journal QAUDJRN:

   ```
   CRTJRN JRN(QSYS/QAUDJRN) JRNRCV(QGPL/AUDRCV0001) MNGRCV(*SYSTEM)
   TEXT('Security Audit Journal') AUT(*EXCLUDE)
   ```

3. Set the audit level (QAUDLVL) system value using WRKSYSVAL or the SECTOOLS menu option 10 (*Change security auditing*).

4. Start auditing by changing the QAUDCTL system value to a value other than *NONE. Use the WRKSYSVAL command or SECTOOLS menu option 10 (*Change security auditing*).

Note: In this example, the journal receivers are placed in QGPL. We recommend that you put them in your own library and secure that library.

Figure 54 shows the security auditing values used during our tests.

```
                        Current Security Auditing Values

Security Auditing Journal Values

  Security journal QAUDJRN exists . . . . . :   YES

  Journal receiver attached to QAUDJRN  . . :   AUDRCV0001
    Library . . . . . . . . . . . . . . . . :     QGPL

Security Auditing System Values

  Current QAUDCTL system value  . . . . . . :   *AUDLVL   *OBJAUD   *NOQTEMP

  Current QAUDLVL system value  . . . . . . :   *AUTFAIL  *OBJMGT   *SECURITY
                                                *CREATE   *DELETE
```

Figure 54. Security Auditing Values

3.4.1 Audit Server User Profiles

1. Verify that QTMHHTTP and QTMHHTTP1 have Password *NONE:

   ```
   PRTUSRPRF TYPE(*PWDINFO) SELECT(*USRCLS)
   ```

```
                                        User Profile Information                                        Page     1
5716SS1 V3R7M0  961108                                                                      SYSNAM    12/13/96  13:11:32
 Report type  . . . . . . . . . . :   *PWDINFO
 Select by  . . . . . . . . . . . :   *USRCLS
 User class . . . . . . . . . . . :   *ALL
 QPWDEXPITV system value  . . . . :      31
 User                          Not Valid          No        Previous     Password     Expiration     Password
 Profile        Status         Sign-ons         Password    Sign-on      Changed       Interval      Expired
 ANONYMOUS      *ENABLED                0         X          / /        12/11/96      *SYSVAL        *NO
 LABUSER        *ENABLED                0                   07/10/96     07/10/96      *SYSVAL        *NO
 MRK            *ENABLED                0                   12/09/96     12/02/96      *SYSVAL        *NO
 M36DFT         *ENABLED                0                   04/11/96     04/11/96      *SYSVAL        *NO
 PAYROLLUSR     *ENABLED                0                     / /        12/06/96      *SYSVAL        *NO
 WEBMASTER      *ENABLED                0                   11/23/96     11/16/95      *SYSVAL        *NO
 QTCP           *ENABLED                0         X           / /        10/31/95      *SYSVAL        *NO
 QTMHHTP1       *ENABLED                0         X           / /        10/23/96      *NOMAX         *NO
 QTMHHTTP       *ENABLED                0         X           / /        10/23/96      *NOMAX         *NO
 QTMPLPD        *ENABLED                0         X           / /        10/31/95      *NOMAX         *NO
 WEBUSER        *ENABLED                0                   12/03/96     12/03/96      *SYSVAL        *NO
 WWWUSER        *ENABLED                0                   12/09/96     12/10/96      *SYSVAL        *NO
                                      * * * * *  E N D  O F  L I S T I N G  * * * * *
```

*Figure 55. PRTSYSSECA Report - Users with Password *NONE*

2. Verify that *only* the intended user profiles have *ALLOBJ or *IOSYSCFG special authorities:

   ```
   PRTUSRPRF TYPE(*AUTINFO) SELECT(*SPCAUT) SPCAUT(*ALLOBJ *IOSYSCFG) +
   USRCLS(*ALL)
   ```

```
                                        User Profile Information                                        Page     1
5716SS1 V3R7M0  961108                                                                      SYSNAM    12/13/96  13:34:37
 Report type  . . . . . . . . . . :   *AUTINFO
 Select by  . . . . . . . . . . . :   *SPCAUT
 Special authorities  . . . . . :   *ALLOBJ     *IOSYSCFG
                                 ------------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
 BOSS        *NONE               X     X     X     X          X     X    *SECOFR    *USRPRF        *NONE      *PRIVATE   *NO
 QLPAUTO     *NONE         X           X     X     X     X               *SYSOPR    *USRPRF        *NONE      *PRIVATE   *NO
 QLPINSTALL  *NONE         X           X     X     X     X               *SYSOPR    *USRPRF        *NONE      *PRIVATE   *NO
 QSECOFR     *NONE         X     X     X     X     X     X     X     X   *SECOFR    *USRPRF        *NONE      *PRIVATE   *NO
 QSYS        *NONE         X     X     X     X     X     X     X     X   *SECOFR    *USRPRF        *NONE      *PRIVATE   *NO
 WEBMASTER   *NONE                     X                                 *USER      *USRPRF        *NONE      *PRIVATE   *NO
                                        * * * * *  E N D  O F  L I S T I N G  * * * * *
```

*Figure 56. PRTSYSSECA Report - Users with Special Authority *IOSYSCFG and *ALLOBJ*

3. For the directories and subdirectories your HTTP server serves HTML pages from (/WWWHARM and /WWWHARM/ITSOIC.400/ in our example), verify that object authority is set up according to your policy. In our example:

   - *PUBLIC has Read and eXecute authority.
   - Webmaster has *ALL.

   ```
   DSPAUT OBJ('/WWWHARM')
   ```

```
                                          Display Authority                                            Page     1
5716SS1 V3R7M0  961108                                                                      SYSNAM    12/13/96  14:50:22
 Object . . . . . . . . . . . :   /WWWHARM
 Owner  . . . . . . . . . . . :   WEBMASTER
 Primary group  . . . . . . . :   *NONE
 Authorization list . . . . . :   *NONE
                 Data       -------------------Data Authorities---------------------     ------Object Authorities-------
 User            Authority   Objopr     Read     Add     Update     Delete     Execute     Exist     Mgt     Alter     Ref
 *PUBLIC         *RX
 WEBMASTER       *RWX          X         X        X        X          X          X
                                      * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 57. DSPAUT Report - Authority to /WWWHARM Directory

```
DSPAUT OBJ('/WWWHARM/ITSOIC.400')
```

```
                                        Display Authority                                                Page     1
5716SS1 V3R7M0  961108                                                                  RCHASMO6  12/13/96  14:50:48
 Object . . . . . . . . . . . . :   /WWWHARM/ITSOIC.400
 Owner  . . . . . . . . . . . . :   WEBMASTER
 Primary group  . . . . . . . . :   *NONE
 Authorization list . . . . . . :   *NONE
                Data       --------------------Data Authorities---------------------     ------Object Authorities-------
 User           Authority     Objopr     Read     Add     Update     Delete     Execute     Exist     Mgt     Alter     Ref
 *PUBLIC        *RX             X         X                                      X
 WEBMASTER      *RWX            X         X        X        X          X          X
                                       * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 58. DSPAUT Report - Authority to /WWWHARM/ITSOIC.400 Directory

Note: By setting the Web server root directory to (/WWWHARM in our examples) the desired authority, all of the subdirectories and stream files under it should inherit this authority. You must audit changes to the Web server directories, subdirectories, and files.

4. Verify that the QTMHHTTP and QTMHHTP1 user profiles' public authority is *EXCLUDE, and that they are not used as group profiles. If used as group profiles, the group members are shown in Figure 59:

   ```
   DSPOBJAUT OBJ(QTMHHTTP) OBJTYPE(*USRPRF)
   ```

```
                                    Display Object Authority                                             Page     1
5716SS1 V3R7M0  961108                                                                  RCHASMO6  12/13/96  17:09:34
 Object . . . . . . . :   QTMHHTTP                              Owner  . . . . . . . :   QSYS
   Library  . . . . . :     QSYS                                Primary group  . . . :   *NONE
 Object type  . . . . :   *USRPRF

                           Object      -------------Object---------------    --------------------Data--------------------
 User          Group       Authority    Opr    Mgt    Exist    Alter    Ref    Read    Add    Update    Delete    Execute
 QSYS                      *ALL          X      X       X        X       X      X       X       X         X          X
 QTMHHTTP                  USER DEF      X      X                               X       X       X         X          X
 *PUBLIC                   *EXCLUDE
                                    * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 59. DSPOBJAUT Report - Public Authority to QTMHHTTP User Profile

```
DSPOBJAUT OBJ(QTMHHTP1) OBJTYPE(*USRPRF)
```

```
                                    Display Object Authority                                             Page     1
5716SS1 V3R7M0  961108                                                                  RCHASMO6  12/15/96  13:49:25
 Object . . . . . . . :   QTMHHTP1                              Owner  . . . . . . . :   QSYS
   Library  . . . . . :     QSYS                                Primary group  . . . :   *NONE
 Object type  . . . . :   *USRPRF

                           Object      -------------Object---------------    --------------------Data--------------------
 User          Group       Authority    Opr    Mgt    Exist    Alter    Ref    Read    Add    Update    Delete    Execute
 QSYS                      *ALL          X      X       X        X       X      X       X       X         X          X
 QTMHHTP1                  USER DEF      X      X                               X       X       X         X          X
 *PUBLIC                   *EXCLUDE
                                    * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 60. DSPOBJAUT Report - Public Authority to QTMHHTP1 User Profile

5. Verify that the QTMHHTTP user profile does not have access to other important libraries such as PAYROLL:

   ```
   DSPUSRPRF URSPRF(QTMHHTTP) TYPE(*OBJAUT)
   ```

```
                                        Display Authorized Objects                                          Page     1
5716SS1 V3R7M0  961108                                                                          RCHASM06  12/14/96  13:51:05
 User Profile . . . . . . . . . . . . . . . :   QTMHHTTP
                                  -----------Object-------------   --------------Data--------------             List
 Object       Library      Type       Opr   Mgt   Exist   Alter   Ref   Read   Add   Upd   Dlt   Execute   Exclude   Mgt
 PAYROLL      QSYS         *LIB                                                                                  X
 QSC3040940   QSYS         *LIB        X     X      X       X      X     X      X     X     X       X
 QSC3040941   QSYS         *LIB        X     X      X       X      X     X      X     X     X       X
 QTMHHTTP     QSYS         *USRPRF     X     X                           X      X     X     X       X
 QTMHHTTP     QTCP         *JOBD       X                                 X                          X
 QTMHHTTP     QTCP         *CLS        X                                 X                          X
*CPI220D - 92 objects were not included in this list.
                                    * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 61. DSPUSRPRF Report - PAYROLL Library Authority

6. Verify the HTTP user profile's authority to the CGI library (ITSOIC400 in our example).

   You need to check if the following user profiles have the correct authority to the CGI library:

   - QTMHHTP1 - *USE
   - WEBMASTER - *ALL
   - PUBLIC - *EXCLUDE

   ```
   DSPOBJAUT OBJ(ITSOIC400) OBJTYPE(*LIB)
   ```

```
                                        Display Object Authority                                            Page     1
5716SS1 V3R7M0  961108                                                                          RCHASM06  12/14/96  15:21:31
 Object . . . . . . . :   ITSOIC400                          Owner . . . . . . . :   QPGMR
   Library  . . . . . :     QSYS                             Primary group . . . :   *NONE
 Object type  . . . . :   *LIB
 Object secured by authorization list  . . . . . . . . . . . . :   *NONE
                          Object      -------------Object---------------    -------------------Data-------------------
 User          Group      Authority    Opr    Mgt    Exist    Alter    Ref    Read    Add    Update    Delete    Execute
 QTMHHTP1                 *USE          X                                     X                                    X
 WEBMASTER                *ALL          X      X       X        X       X     X       X       X         X          X
 *PUBLIC                  *EXCLUDE
                                    * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 62. DSPOBJAUT Report - CGI Library Authorities

3.4.2 Audit HTTP Configuration Files

You can use the following command to see when the last change was made:

```
DSPOBJD OBJ(QUSRSYS/QATMHTTPC) OBJTYPE(*FILE) DETAIL(*FULL)
```

```
5716SS1 V3R7M0  961108            Display Object Description  - Full                    12/14/96 16:37:36       Page    1
Object . . . . . :   QATMHTTPC       Attribute  . . . . . :    PF
  Library  . . . . :     QUSRSYS      Owner  . . . . . . . :    QSYS
Type . . . . . :   *FILE            Primary Group  . . . :    *NONE
User-defined information:
  Attribute  . . . . . . . . . . . :
  Text . . . . . . . . . . . . . . :   HTTP Configuration physical file
Creation information:
  Creation date/time . . . . . . . :   10/23/96  22:55:18
  Created by user  . . . . . . . . :   QLPINSTALL
  System created on  . . . . . . . :   SYSTEM06
  Object domain  . . . . . . . . . :   *SYSTEM
Change/Usage information:
  Change date/time . . . . . . . . :   12/11/96  10:26:08
  Usage data collected . . . . . . :   YES
  Date last used . . . . . . . . . :   12/14/96
  Days used count  . . . . . . . . :      17
  Date use count reset . . . . . . :
  Allow change by program  . . . . :   NO
Auditing information:
  Object auditing value  . . . . . :   *NONE
Storage information:
  Size . . . . . . . . . . . . . . :            167936
  Offline size . . . . . . . . . . :            0
  Freed  . . . . . . . . . . . . . :   NO
  Compressed . . . . . . . . . . . :   INELIGIBLE
  Auxiliary storage pool . . . . . :    1
    Object overflowed  . . . . . . :     NO
Save/Restore information:
  Save date/time . . . . . . . . . :
  Restore date/time  . . . . . . . :
  Save command . . . . . . . . . . :
  Device type  . . . . . . . . . . :

                * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 63. DSPOBJD Report - Checking Date of Last Change

1. Change the object auditing value to *CHANGE:

   ```
   CHGOBJAUD OBJ(QUSRSYS/QATMHTTPC) OBJTYPE(*FILE) OBJAUD(*CHANGE)
   ```

2. Duplicate the model database outfile (QASYZCJE) to auditing library:

   ```
   CRTDUPOBJ OBJ(QASYZCJE) FROMLIB(QSYS) OBJTYPE(*FILE) TOLIB(AUDITLIB)
   ```

3. Periodically, (once a day) display journal entries:

   ```
   DSPJRN JRN(QAUDJRN) ENTTYP(ZC) OUTPUT(*OUTFILE) OUTFILFMT(*TYPE2) +
   OUTFILE(AUDITLIB/QASYZCJE)
   ```

4. Write a QUERY (or use SQL/400®) to monitor for changes to the HTTP server configuration file:

```
5716QU1  V3R7M0  961108            IBM Query/400             SYSTEM06 12/16/96
  Query . . . . . . . . . . . . . . . . HTTPCHG
    Library . . . . . . . . . . . . . . AUDITLIB
  Query text  . . . . . . . . . . . . . Changes to QUSRSYS/QATMHTTPC

Selected files
  ID     File          Library       Member        Record Format
  T01    QASYZCJE      AUDITLIB      QASYZCJE      QASYZCJE

Select record tests
  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')
           ZCONAM           EQ       'QATMHTTPC'

Ordering of selected fields
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  ZCDATE                                       Date of entry
  ZCTIME                                       Time of entry
  ZCONAM                                       Name of object
  ZCOLIB                                       Library name
  ZCUSPF                                       User profile name

                                                   Changes to QUSRSYS/QATMHTTPC


12/16/96  17:25:10                        PAGE    1
DATE       TIME   OBJECT      LIBRARY     USER
                  NAME        NAME        PROFILE
121596  174,225   QATMHTTPC   QUSRSYS     WEBMASTER
121596  174,305   QATMHTTPC   QUSRSYS     WEBMASTER
121696   32,510   QATMHTTPC   QUSRSYS     BADGUY
121696   82,510   QATMHTTPC   QUSRSYS     WEBMASTER
* * *  E N D  O F  R E P O R T  * * *
```

Figure 64. Changes to HTTP Server Configuration File (QUSRSYS/QATMHTTPC)

3.4.3 Auditing Objects in Your CGI Library (ITSOIC400)

After approving the CGI programs that are loaded on your system, you have to make sure no one makes unauthorized changes to them.

Changes to objects in the CGI library should be under tight control. Typically, only the webmaster should be allowed to move new programs and other objects into this library.

To better audit the CGI library, use the following steps:

1. Verify that the QAUDLVL system value includes the *CREATE option. Creating new objects can only be controlled by this system value. This also logs in the security journal creation of new objects in all libraries.
2. Change the description for your CGI library to log all change access to objects by all users in this library:

   ```
   CHGLIB LIB(ITSOIC400) CRTOBJAUD(*ALL)
   ```

3. To quickly display objects created, changed, or deleted, use the DSPAUDJRNE command with journal entry type CO, ZC, or DO.
4. To display changes to the CGI library only, use the following steps:
   a. Duplicate the model database outfile to auditing library:

      For new object creation or replacement of existing object:

      ```
      CRTDUPOBJ OBJ(QASYCOJE) FROMLIB(QSYS) OBJTYPE(*FILE) TOLIB(AUDITLIB)
      ```

      For all delete operations:

      ```
      CRTDUPOBJ OBJ(QASYDOJE) FROMLIB(QSYS) OBJTYPE(*FILE) TOLIB(AUDITLIB)
      ```

For changes to library objects:

```
CRTDUPOBJ OBJ(QASYZCJE) FROMLIB(QSYS) OBJTYPE(*FILE) TOLIB(AUDITLIB)
```

b. Periodically, (once a day) display journal entries:

```
DSPJRN JRN(QAUDJRN) ENTTYP(CO) OUTPUT(*OUTFILE) OUTFILFMT(*TYPE2) +
OUTFILE(AUDITLIB/QASYCOJE)

DSPJRN JRN(QAUDJRN) ENTTYP(DO) OUTPUT(*OUTFILE) OUTFILFMT(*TYPE2) +
OUTFILE(AUDITLIB/QASYCOJE)

DSPJRN JRN(QAUDJRN) ENTTYP(ZC) OUTPUT(*OUTFILE) OUTFILFMT(*TYPE2) +
OUTFILE(AUDITLIB/QASYCOJE)
```

c. Write a QUERY (or use SQL/400) to monitor for objects created or replaced in your CGI library (ITSOIC400 in our example):

```
5716QU1  V3R7M0  961108           IBM Query/400            SYSNAM   12/16/96
  Query . . . . . . . . . . . . . . . . CRTNEWOBJ
    Library . . . . . . . . . . . . . . AUDITLIB
  Query text  . . . . . . . . . . . . . Objects Created or Replaced in ITSOIC

Selected files
  ID     File           Library       Member        Record Format
  T01    QASYCOJE       AUDITLIB      *FIRST        QASYCOJE

Select record tests
  AND/OR   Field             Test     Value (Field, Numbers, or 'Characters')
           COOLIB            EQ       'ITSOIC400'

Ordering of selected fields
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  CODATE                                       Date of entry
  COTIME                                       Time of entry
  COENTT                                       Entry type
  COETYP                                       Type of entry
  COONAM                                       Name of object
  COOLIB                                       Library name
  COOTYP                                       Object type
  COUSPF                                       User profile name

                Objects Created or Replaced in ITSOIC400

DATE       TIME    TYPE  ENTRY  OBJECT      LIBRARY     OBJECT    USER
                         TYPE   NAME        NAME        TYPE      PROFILE
121696  163,622    CO      N    CGIO01      ITSOIC400   *PGM      WEBMASTER
121696  163,742    CO      R    CGIO01      ITSOIC400   *PGM      BADGUY
121696  164,049    CO      R    IMVR501     ITSOIC400   *MODULE   WEBMASTER
```

Figure 65. Objects Created or Replaced in ITSOIC400 - CGI Library

d. Write a QUERY (or use SQL/400) to monitor for objects deleted in your CGI library (ITSOIC400 in our example):

```
5716QU1  V3R7M0  961108           IBM Query/400            SYSNAM   12/16/96
  Query . . . . . . . . . . . . . . . . DLTOBJ
    Library . . . . . . . . . . . . . . AUDITLIB
  Query text  . . . . . . . . . . . . . Objects Deleted from ITSOIC400

Selected files
  ID     File          Library       Member        Record Format
  T01    QASYDOJE      AUDITLIB      *FIRST        QASYDOJE

Select record tests
  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')
           DOOLIB           EQ       'ITSOIC400'

Ordering of selected fields
  Field          Sort      Ascending/  Break  Field
  Name           Priority  Descending  Level  Text
  DODATE                                      Date of entry
  DOTIME                                      Time of entry
  DOETYP                                      Type of entry
  DOONAM                                      Name of object
  DOOLIB                                      Library name
  DOOTYP                                      Object type
  DOUSPF                                      User profile name




                                      Objects Deleted from Library ITSO
                                   QUERY NAME . . . . . DLTOBJ
                                   LIBRARY NAME . . . . AUDITLIB
                                   FILE         LIBRARY      MEMBER
                                   QASYDOJE     AUDITLIB     QASYDOJE
                                   DATE . . . . . . . . 12/16/96
                                   TIME . . . . . . . . 21:28:29
                                             Objects Deleted from ITSOIC40


12/16/96  21:28:29      Objects Deleted from ITSOIC400      PAGE    1
DATE       TIME   ENTRY  OBJECT      LIBRARY     OBJECT    USER
                  TYPE   NAME        NAME        TYPE      PROFILE
121696  145,003     A    IMVR501     ITSOIC400   *PGM      WEBMASTER
121696  153,716     A    MYFILE      ITSOIC400   *FILE     WEBMASTER
121696  162,634     A    CGI001      ITSOIC400   *PGM      BADGUY
```

Figure 66. Objects Deleted in ITSOIC400 - CGI Library

e. Write a QUERY (or use SQL/400) to monitor for objects changed in your CGI library (ITSOIC400 in our example):

```
5763QU1  V3R2M0  960517           IBM Query/400            SYSNAM    3/10/97   16:09:04        Page    1
  Query . . . . . . . . . . . . . . . . CHGOBJ
    Library . . . . . . . . . . . . . . AUDITLIB
  Query text  . . . . . . . . . . . . . Objects Changed in CGI library ITSOIC400

Selected files
  ID     File          Library       Member        Record Format
  T01    QASYZCJE      AUDITLIB      *FIRST        QASYZCJE
Select record tests
  AND/OR   Field            Test     Value (Field, Numbers, or 'Characters')
           ZCOLIB           EQ       'ITSOIC400'
Ordering of selected fields
  Field           Sort      Ascending/  Break  Field
  Name            Priority  Descending  Level  Text
  ZCDATE                                       Date of entry
  ZCTIME                                       Time of entry
  ZCENTT                                       Entry type
  ZCONAM                                       Name of object
  ZCOLIB                                       Library name
  ZCOTYP                                       Object type
  ZCUSPF                                       User profile name


                                    QUERY NAME . . . . . CHGOBJ
                                    LIBRARY NAME . . . . AUDITLIB
                                    FILE         LIBRARY      MEMBER       FORMAT
                                    QASYZCJE     AUDITLIB     QASYZCJE     QASYZCJE
                                    DATE . . . . . . . . 03/10/97
                                    TIME . . . . . . . . 16:10:56
                                         Objects Changed in CGI library ITSOIC400


03/10/97  16:10:56                                          PAGE    1
DATE       TIME   TYPE  OBJECT      LIBRARY     OBJECT    USER
                        NAME        NAME        TYPE      PROFILE
030497   84,429    ZC   CGIPGM1     ITSOIC400   *PGM      WEBMASTER
031097  144,422    ZC   CGIDTAARA   ITSOIC400   *DTAARA   WEBMASTER
031097  154,729    ZC   CGIPAYROLL  ITSOIC400   *PGM      BADGUY
031097  154,753    ZC   CLP41871    ITSOIC400   *PGM      WEBMASTER
* * *  E N D  O F  R E P O R T  * * *
```

Figure 67. Objects Changed in ITSOIC400 - CGI library

3.4.4 Audit Web Server Directories and Files

The heart of your Web server is the directories and files that are accessed from the Internet and your CGI libraries. In Section 3.4.3, "Auditing Objects in Your CGI Library (ITSOIC400)" on page 86, we discuss auditing your CGI library.

Use the Change Auditing Value (CHGAUD) command setup or change auditing on your Web server directory. If you set auditing at the Web server root directory level, the sub-directories and files under it inherit the auditing value.

```
CHGAUD OBJ('\HARM') OBJAUD(*ALL)
```

Displaying or printing the audit records for IFS directories and stream files is trickier than for objects in the QSYS file system. The path name of the file is not displayed. If you use the DSPAUDJRNE command, you get the following results:

```
                                Display Report
Query . . . :   QSYS/QSECZC                 Report width . . . . . :
Position to line  . . . . .               Shift to column  . . . . . .
Line    ....+....1....+....2....+....3....+....4....+....5....+....6....
           USER       OBJECT     LIBRARY    OBJECT   JOB        JOB
           PROFILE    NAME       NAME       TYPE     NAME       USER
000285 ZC WEBMASTR   *N         *N         *STMF    QPWFSERVSO QUSER
000286 ZC BADGUY     *N         *N         *STMF    QPWFSERVSO QUSER
000287 ZC WEBMASTER  *N         *N         *STMF    QPWFSERVSO QUSER
000288 ZC WEBMASTER  ITSOIC400  QSYS       *LIB     QTFTP00939 QTCP
000289 ZC WEBMASTER  INETCGI    ITSOIC400  *FILE    QTFTP00939 QTCP
000290 ZC BADGUY     PAYCGI     ITSOIC400  *FILE    QTFTP00939 QTCP
```

Figure 68. Using DSPAUDJRN to Display Audit Journal Entries for Web Server Directories and Files

As shown in Figure 68, we can tell that BADGUY has made changes to a stream file but we do not know which one. The **Qp0lGetPathFromFileID()** function determines an absolute path name of the file identified by *fileid*. For a description of this function, refer to the *AS/400 System API Reference*. See also the *Parent File ID* and *Object File ID* described in the *AS/400 Security - Reference*.

Figure 69 shows the output of a sample tool written using the Qp0lGetPathFromFileID() function.

```
CALL  DACP0LCU
*** No AF audit records to process or End of AF audit records found.
*** Finished processing AF audit records. ***

0000001733  CO N Path does not exist on system
0000001755  CO N /HARM/Cfg/Auth/Shipped/WblUsr.cfg
0000003118  CO N Path does not exist on system
0000014782  CO N /QTCPTMM/MAIL/qlockbox
0000014912  CO N /qserver/streamfile.end
*** No CO audit records to process or End of CO audit records found. **
*** Finished processing CO audit records. ***

0000001654  DO A Path does not exist on system
 *** No DO audit records to process or End of DO audit records found. **
 *** Finished processing DO audit records. ***

 0000002412  ZC C /HARM/secprf.txt
 0000002458  ZC C /HARM/payroll
 0000002370  ZC C /HARM/WEBDOCS/private/missle.gif
 0000002376  ZC C /HARM/WEBDOCS/private/pvthome.htm
 0000002382  ZC C Path does not exist on system
 0000002383  ZC C Path does not exist on system
 0000002394  ZC C /HARM/claus/HTTPSVR.PRZ
 *** No ZC audit records to process or End of ZC audit records found. **
 *** Finished processing ZC audit records. ***
```

Figure 69. Use of Qp0lGetPathFromFileID() to Retrieve IFS Path Name from AuditJournal

Our simple tool writes the Audit journal entry number, entry type (AF, CO, ZC or DO only), detailed entry, and the path name in your job log. You can use this information to locate the original journal entry. A path name of *Path does not exist on system* indicates that the stream file or directory has been deleted since the journal entry was recorded.

For a complete printout of the programs used to retrieve the path names and produce the output shown in Figure 69, refer to Figure 70 on page 91.

```
//BCHJOB   JOB(DACPOLCU) JOBD(C2SECOFR) ENDSEV(99)

/*********************************************************************/
/*  Create the DACPOLCU test program.                                */
CRTCLPGM   PGM(IFSC2LIB/DACPOLCU) SRCFILE(DACPOLCU)              +
           AUT(*USE)              REPLACE(*YES)

/*********************************************************************/
/*  Create the AF audit record verification program DACPOLCUAF       */
CRTCLPGM   PGM(IFSC2LIB/DACPOLCUAF) SRCFILE(AF)                   +
           AUT(*USE)                REPLACE(*YES)

/*********************************************************************/
/*  Create the CO audit record verification program DACPOLCUCO       */
CRTCLPGM   PGM(IFSC2LIB/DACPOLCUCO) SRCFILE(CO)                   +
           AUT(*USE)                REPLACE(*YES)

/*********************************************************************/
/*                                                                   */
/*  Below is the "source file" for the actual test program DACPOLCU  */
/*                                                                   */
/*********************************************************************/
//DATA     FILE(DACPOLCU)  FILETYPE(*SRC)

             PGM
/*********************************************************************/
/*                                                                   */
/* PGM DACPOLCU                                                      */
/*                                                                   */
/*                                                                   */
/*  PURPOSE:                                                         */
/*                                                                   */
/*  PROGRAM INSTALLATION:                                            */
/*                                                                   */
/*     To create the DACPOLCU program and all of its supporting      */
/*     audit validation programs, place this part in a source        */
/*     physical file on the test AS/400 in the member DACPOLCU and   */
/*     enter the following command:                                  */
/*                                                                   */
/*         SBMDBJOB FILE(srcf-lib-name/srcf-name) MBR(DACPOLCU)      */
/*                                                                   */
/*                                                                   */
/*                                                                   */
/*  PROGRAM EXECUTION:                                               */
/*                                                                   */
/*                                                                   */
/*    Enter the command:          CALL DACPOLCU                      */
/*                                                                   */
/*                                                                   */
/*                                                                   */
/*********************************************************************/
```

Figure 70 (Part 1 of 6). Listing of Program Used to Display Path of IFS Files in Journal Entry Types AF, CO, ZC and DO

```
/*********************************************************************/
/*          Variables with test object names.                        */
/*********************************************************************/

            DCL        &TSTLIB   *CHAR 10 'DACPOLCU  '
            DCL        &FILENAME *CHAR 10 'FILE      '
            DCL        &PGMNAME  *CHAR 10 'PGM       '

            MONMSG     CPF0000 /* Generic monitor, ignore exceptions*/

/*********************************************************************/
/*          Dump the AF audit records into an outfile AND call the   */
/*          program to validate them.                                */
/*                                                                   */
/*********************************************************************/
            CRTDUPOBJ  OBJ(ASYAFJE) FROMLIB(QSYS) OBJTYPE(*FILE) +
                         TOLIB(QTEMP) NEWOBJ(AF)
            DSPJRN   JRN(QAUDJRN) ENTTYP(AF) OUTPUT(*OUTFILE) +
                     OUTFILE(QTEMP/AF) OUTFILFMT(*TYPE2) +
                     ENTDTALEN(*OUTFILFMT)
            CALL       DACPOLCUAF    PARM(AF QTEMP)

/*********************************************************************/
/*          Dump the CO audit records into an outfile AND call the   */
/*          program to validate them.                                */
/*                                                                   */
/*********************************************************************/
            CRTDUPOBJ  OBJ(ASYCOJE) FROMLIB(QSYS) OBJTYPE(*FILE) +
                         TOLIB(QTEMP) NEWOBJ(CO)
            DSPJRN   JRN(QAUDJRN) ENTTYP(CO) OUTPUT(*OUTFILE) +
                     OUTFILE(QTEMP/CO) OUTFILFMT(*TYPE2) +
                     ENTDTALEN(*OUTFILFMT)
            CALL       DACPOLCUCO    PARM(CO QTEMP)

/*********************************************************************/
/* Program exit point.                                               */
/*********************************************************************/
            ENDPGM              /* End of test. */
//
```

Figure 70 (Part 2 of 6). Listing of Program Used to Display Path of IFS Files in Journal Entry Types AF, CO, ZC and DO

```
/*****************************************************************/
/*Source for AF audit record verification program DACPOLCUAF*/
/*****************************************************************/
//DATA     FILE(AF)          FILETYPE(*SRC)
          PGM        PARM(&FILENAME &FILELIB)
          /*********************************************************/
          /*                                                       */
          /*  This program validates the AF records.               */
          /*                                                       */
          /*********************************************************/

          /*********************************************************/
          /* Declare input file name and library parameters.       */
          /*********************************************************/
          DCL        &RTNNAME   *CHAR   100
          DCL        &NULLTERM  *CHAR   1   X'00'
          DCL        &NULLFLID  *CHAR   16  X'80000000000000000000000000000000'
          DCL        &FILENAME  *CHAR   10
          DCL        &FILELIB   *CHAR   10

          DCLF       QASYAFJE
          MONMSG     CPF0000

          /*********************************************************/
          /*  Issue Override DB file command so the OUTFILE can    */
          /*  be read.                                             */
          /*********************************************************/
          OVRDBF     QASYAFJE TOFILE(&FILELIB/&FILENAME)

/*****************************************************************/
/*  Validate the 1st record.  (AF-A )                            */
/*****************************************************************/
          RCVF
          MONMSG     CPF0864 EXEC(DO)
             SNDPGMMSG '*** No AF audit records to process.  No +
               AF audit records found. ***'  TOPGMQ(*PRV (DACPOLCU))
             GOTO DONEV2
          ENDDO

/*****************************************************************/
/*  From the File ID in the audit record get the path name,      */
/*  if no File ID exists use name library and object type.       */
/*****************************************************************/
          IF ((%SST(&AFOFID  1  16 ) *EQ &NULLFLID)) THEN(DO)
             SNDPGMMSG (&AFONAM) TOPGMQ(*PRV (DACPOLCU))
             SNDPGMMSG (&AFOLIB) TOPGMQ(*PRV (DACPOLCU))
             SNDPGMMSG (&AFOTYP) TOPGMQ(*PRV (DACPOLCU))
          ENDDO
          ELSE (DO)
            CALL GETPATH (&RTNNAME &AFOFID)
            SNDPGMMSG (&RTNNAME) TOPGMQ(*PRV (DACPOLCU))
          ENDDO

DONEV2:
          SNDPGMMSG '*** Done processing AF audit records. +
                ***'  TOPGMQ(*PRV (DACPOLCU))
          ENDPGM
//
```

Figure 70 (Part 3 of 6). Listing of Program Used to Display Path of IFS Files in Journal Entry Types AF, CO, ZC and DO

```
/*****************************************************************/
/*  Source for CO audit record verification program DACPOLCUCO*
/*****************************************************************/
//DATA     FILE(CO)          FILETYPE(*SRC)
          PGM        PARM(&FILENAME &FILELIB)
          /*********************************************************/
          /*                                                       */
          /*  This program validates the CO records.               */
          /*                                                       */
          /*********************************************************/

          /*********************************************************/
          /* Declare input file name and library parameters.       */
          /*********************************************************/
          DCL        &RTNNAME   *CHAR   100
          DCL        &NULLTERM  *CHAR   1   X'00'
          DCL        &NULLFLID  *CHAR   16  X'80000000000000000000000000000000'
          DCL        &FILENAME  *CHAR   10
          DCL        &FILELIB   *CHAR   10

          DCLF       QASYCOJE
          MONMSG     CPF0000

          /*********************************************************/
          /*  Issue Override DB file command so the OUTFILE can    */
          /*  be read.                                             */
          /*********************************************************/
          OVRDBF     QASYCOJE TOFILE(&FILELIB/&FILENAME)

/*****************************************************************/
/*  Validate the 1st record.  (CO-N )                            */
/*****************************************************************/
          RCVF
          MONMSG     CPF0864 EXEC(DO)
             SNDPGMMSG '*** No CO audit records to process.  No +
               CO audit records found. ***'  TOPGMQ(*PRV (DACPOLCU))
             GOTO DONEV2
          ENDDO
```

Figure 70 (Part 4 of 6). Listing of Program Used to Display Path of IFS Files in Journal Entry Types AF, CO, ZC and DO

```
/*****************************************************************/
/*  From the File ID in the audit record get the path name.      */
/*****************************************************************/
          IF ((%SST(&COOFID  1  16 ) *EQ &NULLFLID)) THEN(DO)
             SNDPGMMSG (&COONAM) TOPGMQ(*PRV (DACPOLCU))
             SNDPGMMSG (&COOLIB) TOPGMQ(*PRV (DACPOLCU))
             SNDPGMMSG (&COOTYP) TOPGMQ(*PRV (DACPOLCU))
          ENDDO
          ELSE (DO)
            CALL GETPATH (&RTNNAME &COOFID)
            SNDPGMMSG (&RTNNAME) TOPGMQ(*PRV (DACPOLCU))
          ENDDO

DONEV2:

          SNDPGMMSG '*** Done processing CO audit records. +
                ***'  TOPGMQ(*PRV (DACPOLCU))
          ENDPGM
//

//ENDBCHJOB
*********************************************************************
```

Figure 70 (Part 5 of 6). Listing of Program Used to Display Path of IFS Files in Journal Entry Types AF, CO, ZC and DO

```
*********************************************************************
GETPATH ************************************************************************
/*********************************************************************/
/*                                                                   */
/*  PROGRAM NAME: GETPATH                                            */
/*                                                                   */
/*                                                                   */
/*  PURPOSE:  The purpose of this program is to retrieve the         */
/*            pathname from a fileid in an audit record.             */
/*                                                                   */
/*  INPUT:    Parm 1:  A 50 character buffer to hold the name of     */
/*                     path returned.                                */
/*                                                                   */
/*            Parm 2:  The 16 character file id (taken from the      */
/*                     audit record).                                */
/*                                                                   */
/*  EXAMPLE:  CALL GETPATH PARM(&VARNAME &FILEID)                    */
/*                                                                   */
/*  PROGRAM INSTALLATION:                                            */
/*                                                                   */
/*     To create the GETPATH program, simply place this file         */
/*     into a source physical file on the AS/400 in the member       */
/*     GETPATH and enter the following command:                      */
/*                                                                   */
/*         CRTBNDC  PGM(target-lib-name/GETPATH) +                   */
/*                  SRCFILE(srcf-lib-name/srcf-name) +               */
/*                  SRCMBR(GETPATH)                                  */
/*                                                                   */
/*                                                                   */
/*                                                                   */
/*********************************************************************/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <except.h>
#include <lecond.h>
#include <errno.h>
#include <qusec.h>
#include <QpOlstdi.h>
#include <sys/stat.h>
#include <qmhrcvpm.h>

/* define a printf to use for debugging purposes */
#define debugprintf printf

/*********************************************************************/
/* Main                                                              */
/* This function is the main function of the program.                */
/*********************************************************************/
int main (int argc, char *argv[])

{
       QpOlGetPathFromFileID(argv[1], 50, argv[2]);
 /* if (QpOlGetPathFromFileID(argv[1], 50, argv[2]LL) */
   /* printf ("errno = %s \n", strerror(errno)); */
}
```

Figure 70 (Part 6 of 6). Listing of Program Used to Display Path of IFS Files in Journal Entry Types AF, CO, ZC and DO

In the example, we first display the audit journal entry types AF, CO, ZC, and DO. We use the C program GETPATH to determine the path name of the files, passing the file ID obtained from the audit journal entry.

3.4.5 Prevent Unwanted TCP/IP Servers from Starting Automatically

Make sure that only the HTTP server and **maybe** FTP start on your Web server AS/400 system. Verify that unwanted services are not being started:

- Make sure that the start-up program specified in the QSTRUPPGM startup program system value is not starting unwanted TCP/IP servers.
- Change the default of the STRTCPSVR command from *ALL to **only** the desired servers (HTTP and, maybe, FTP).

  ```
  CHGCMDDFT CMD(STRTCPSVR) NEWDFT('SERVER(*HTTP)')
  ```

- Audit the *PUBLIC authority to the STRTCPSVR command. The ship value is *PUBLIC EXCLUDE and should remain that way.

  ```
  DSPOBJAUT OBJ(STRTCPSVR) OBJTYPE(*CMD)
  ```

```
                          Display Object Authority

Object . . . . . :   STRTCPSVR       Owner  . . . . . :   QSYS
  Library  . . . . :     QSYS          Primary group  . . :   *NONE
Object type  . . . :   *CMD

Object secured by authorization list  . . . . . . . . . . . :   *NONE

                         Object
User        Group       Authority
QSYS                    *ALL
QPGMR                   *USE
QSYSOPR                 *USE
QSRVBAS                 *USE
QSRV                    *USE
*PUBLIC                 *EXCLUDE
```

- Use the Change xxx Attributes command to prevent other servers from starting automatically:

  ```
  CHGSNMPA AUTOSTART(*NO)
  CHGTELA AUTOSTART(*NO)
  CHGFTPA AUTOSTART(*NO)
  CHGSMTPA AUTOSTART(*NO)
  CHGLPDA AUTOSTART(*NO)
  CHGWSGA AUTOSTART(*NO)
  CHGPOPA AUTOSTART(*NO)
  ```

- You can also use the following command to stop unwanted servers that are already started:

  ```
  ENDTCPSVR SERVER(server-name)
  ```

3.4.6 Web Server Log Files

The Web server logs access and error information in the files specified in the *AccessLog* and *ErrorLog* directives in the HTTP server configuration. The authority to the access and error log files in the QUSRSYS library is based on whether the *LogFormat* directive is set to DDS or COMMON.

- When *LogFormat* is set to DDS, the authority for the log files is taken from the authority associated with QTCP/QATMHLOG for the access log and QTCP/QATMHERR for the error log.

  Prior to starting the server for the first time, the administrator should set the desired authorities for QTCP/QATMHLOG and QTCP/QATMHERR.

Tip

We recommend the following authorities for the access and error log files:

```
QTMHHTTP                *ALL
*PUBLIC                 *EXCLUDE
```

- When the *LogFormat* is set to COMMON, the *PUBLIC authority to access the log and error log files is based on the system value QCRTAUT. We recommend that the administrator (webmaster) use the following steps the first time the log files are created:

  1. Start the HTTP Server to create the access and error log files:

     ```
     STRTCPSVR SERVER(*HTTP)
     ```

  2. End the HTTP server immediately:

     ```
     ENDTCPSVR SERVER(*HTTP)
     ```

  3. Change the authority of the access and error log files. We recommend setting *PUBLIC *EXCLUDE and QTMHHTTP *ALL (the default):

     ```
     RVKOBJAUT OBJ(QUSRSYS/ACCLOG01) OBJTYPE(*FILE) USER(*PUBLIC) AUT(*CHANGE)
     ```

  4. Re-start the HTTP server:

     ```
     STRTCPSVR SERVER(*HTTP)
     ```

3.4.6.1 IC/400 HTTP Server Access Log Analysis

You can analyze the information in the HHTP access and error logs to find out who is accessing your server and the URLs people are interested in. You can also look for abnormal patterns such as unusually high numbers of errors that indicate that someone is trying to guess and find a hole in your HTTP server configuration, unusually high numbers of hits from one particular site that might mean someone is performing a denial-of-service attack, and so on.

We have written a C program to convert the IC/400 HTTP server access log from COMMON format to an externally described file, ACCOUT. We can now write queries on ACCOUT to analyze the information in the access log. An example of such a query is found in Figure 71 on page 98. This is just a section of an 80-page query result of analyzing the data in the access log of our Web server.

```
03/27/97  21:23:54              SERVER ACCESS LOG
PAGE    1

      DATE         Request From                    TYPE  Status  URL

      1997/03/14   w3proxy-b.rchland.ibm.com       GET    200    /
                   w3proxy-a.rchland.ibm.com       GET    304
                   fw.fi.ibm.com                   GET    200
                   CANVM2.mkm.can.ibm.com          GET    200
                   9.29.129.94                     GET    200
                   9.29.161.1                      GET    200
                   csocks1.server.ibm.com          GET    200
                   w3proxy-a.rchland.ibm.com       GET    200
                   jolanki.sto.se.ibm.com          GET    200
                   w3proxy-a.rchland.ibm.com       GET    200
                   bsocks2.server.ibm.com          GET    200
                   gsocks1.server.ibm.com          GET    304
                   asocks2.server.ibm.com          GET    200
URL Summary
COUNT 13                                                          13

      1997/03/14   w3proxy-b.rchland.ibm.com       GET    200    /itsoroch/pics/as400g.gif
                   w3proxy-a.rchland.ibm.com       GET    304
                   hns.h.ch.ibm.com                GET    200
                   fw.fi.ibm.com                   GET    200
                   9.29.129.94                     GET    200
                   yoda.toraix.can.ibm.com         GET    200
                   w3proxy-a.rchland.ibm.com       GET    200
                   jolanki.sto.se.ibm.com          GET    200
                   w3proxy-a.rchland.ibm.com       GET    304
                   bsocks2.server.ibm.com          GET    200
                   asocks2.server.ibm.com          GET    200
URL Summary
COUNT 11                                                          11

Daily Statistics
COUNT 287                                                         287
```

Figure 71. Query Data in IC/400 HTTP Server Access Log

3.4.6.2 Access Log Analysis Tool - Sample Program

To be able to query the data in the access log, we wrote the C program HTTPACCLOG. This program reads from the HTTP server access log file in COMMON format, parses the data into tokens, and writes it back to the externally described file, ACCOUT. Figure 72 on page 99 shows the source of the HTTPACCLOG C program.

```
/* Program NAME: HTTPACCLOG                                          */
/* PURPOSE: Read HTTP Server AccessLog file and write to an          */
/* externally described file to be used for queries to analyze the */
/* AccessLog data                                                    */
/* The input file (AccessLog) is in COMMON format.                   */
/* Each input record is a string that this program parses to find  */
/* the following tokens (that will be written into the output file */
/* fields):                                                          */
/*TOKEN         LEADING DELIMITER      TRAILING DELIMITER          */
/* logadd       column 1                space                      */
/* date         7 past logadd           :                          */
/* time         :                       space                      */
/* DONTCARE     space                   ″                          */
/* type         ″                       space                      */
/* url          space                   space                      */
/* DONTCARE     space                   space                      */
/* error        space                   space                      */
/* bytes        space                   space or NewLine           */
/* You must display the AccessLog physical file member to understand*/
/* the string layout explained above                                 */

#include <stdio.h>
#include <stdlib.h>
#include <recio.h>
#include <string.h>

/* Header file for ILE C time conversion functions -                 */
/* See SYSTEM API REFERENCE manual                                   */
#include <leawi.h>

/* Pre-Processor automatically creates C structure typedefs from
external file description                                          */
/* ACCOUT         : Output file (externally described)           */
/* OutLib         : Output Library                               */
/* RACCOUT        : Record Format                                */
#pragma mapinc(″AccLog″,″OutLib/ACCOUT(RACCOUT)″,″both″, ″d″)

/* Include the automatically created C structure typedefs         */
#include ″AccLog″

#define InRecSize 513

main () {
/* Declare pointers to input and output file that are initialized */
/* when files are opened.                                         */
  FILE*     inFile;
  _RFILE*   outFile;
  _FEEDBACK fc;

/* Input and output buffers                                       */
  char                      inRec[InRecSize];
/* Name of the structure automatically created by the compiler    */
  OutLib_ACCOUT_RACCOUT_both_t outRec;

  int  lilianDate, isTypeAndUrl;

  char *logAdd, *date, *time, *type, *url, *error, *bytes;
```

Figure 72 (Part 1 of 2). HTTPACCLOG - Converts Access Log from COMMON Format to Externally Described File

```
  /*************************************************************************/
  /* Open input and output files                                           */
  /*************************************************************************/
  inFile  = fopen("QUSRSYS/ACCLOG", "r");
  outFile = _Ropen ("OutLib/ACCOUT", "wr");

  /*************************************************************************/
  /* Read input record                                                     */
  /*************************************************************************/
  while ( fgets (inRec, InRecSize, inFile) != NULL ) {

    memset (&outRec,' ',sizeof(outRec));          /* Clear output record   */
  /* Some records do not include "type" and "url"  */
    isTypeAndUrl = (strstr(inRec,"\"\"") == NULL); /* Missing URL and type? */

    /*************************************************************************/
    /* Parse input record                                                    */
    /* strtok function returns pointer to token defined by delimiter */
    /* C standard library function                                   */
    /*************************************************************************/
    logAdd=                 strtok(inRec, " "   );
    date  =                 strtok(NULL , ":"   ) + 5;
    time  =                 strtok(NULL , " "   );
                            strtok(NULL , "\""  );  /* Skip token */
    type  = (isTypeAndUrl) ? strtok(NULL , " "   )   : "????";
    url   = (isTypeAndUrl) ? strtok(NULL , " "   )   : "????";
                            strtok(NULL , " "   );  /* Skip token */
    error =                 strtok(NULL , " "   );
    bytes =                 strtok(NULL , " \n" );

    /*************************************************************************/
    /* Convert and copy fields to output record                              */
    /*************************************************************************/
    strncpy (outRec.LOGADD, logAdd, strlen(logAdd));
    strncpy (outRec.TIME  , time  , strlen(time)  );
    strncpy (outRec.TYPE  , type  , strlen(type)  );
    strncpy (outRec.URL   , url   , strlen(url)   );
    strncpy (outRec.ERR   , error , strlen(error) );
    CEEDAYS (date       , "DD/MMM/YYYY", &lilianDate, &fc);/* Convert date
                                                      Lillian format */
    CEEDATE (&lilianDate, "YYYY/MM/DD ", outRec.DATE, &fc);/* Convert date from
                                                       Lillian to YYYY/MM/D
    /* sscanf C standard library function to convert from character to
       numeric format                                                  */
    sscanf  (bytes, "%li", &outRec.BYTES);

    /*************************************************************************/
    /* Write output record                                                   */
    /*************************************************************************/
    _Rwrite(outFile, &outRec, sizeof(outRec));
  }
}
```

Figure 72 (Part 2 of 2). HTTPACCLOG - Converts Access Log from COMMON Format to Externally Described File

Figure 73 shows the DDS specifications for the file ACCOUT.

```
A*  File ACCOUT
A          R RACCOUT
A            LOGADD        30
A            DATE          11
A            TIME           8
A            TYPE           4
A            ERR            3
A            BYTES          9B
A*   URL could be as long as 447 bytes
A*   To make it easier to print we limit it to 100 bytes
A            URL          100
```

Figure 73. ACCOUT - Externally Described File to Query Access Log Data

3.4.6.3 Web Server Access Log Analysis Tools

There are many HTTP server access log analysis tools that you can purchase to produce reports or graphics using the data in the IC/400 access log. Most tools are available to run on UNIX or PC operating systems. Using Client Access/400, we transferred the data from our HTTP server access log in COMMON format and used a Windows 95® tool to analyze the data. Figure 74 shows a graph that represents the percentage of hits to our server by status. It also shows the absolute number of entries for a particular status. From the security point of view, this is important information to look at. Many hits that return a status of *FORBIDDEN* or *Not FOUND* indicate suspicious circumstances.

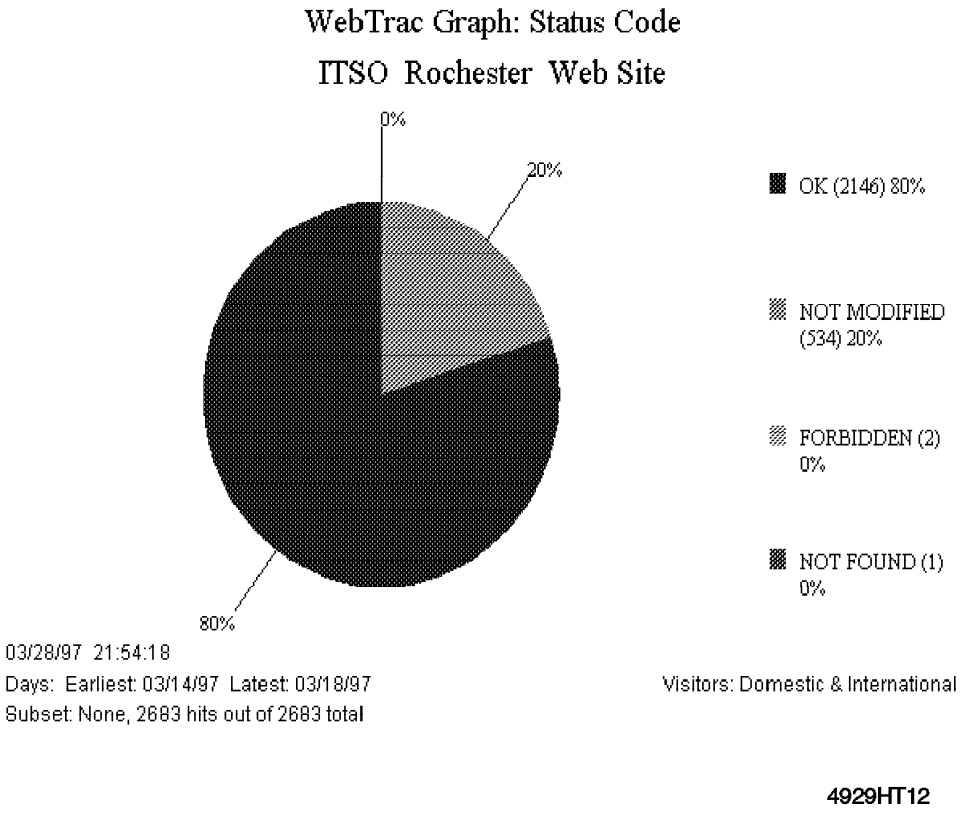


Figure 74. Status Code

Just to provide some more examples of the kind of data you can extract from the access log, we include Figure 75 on page 102. This figure shows the number of bytes transmitted in one-hour intervals.

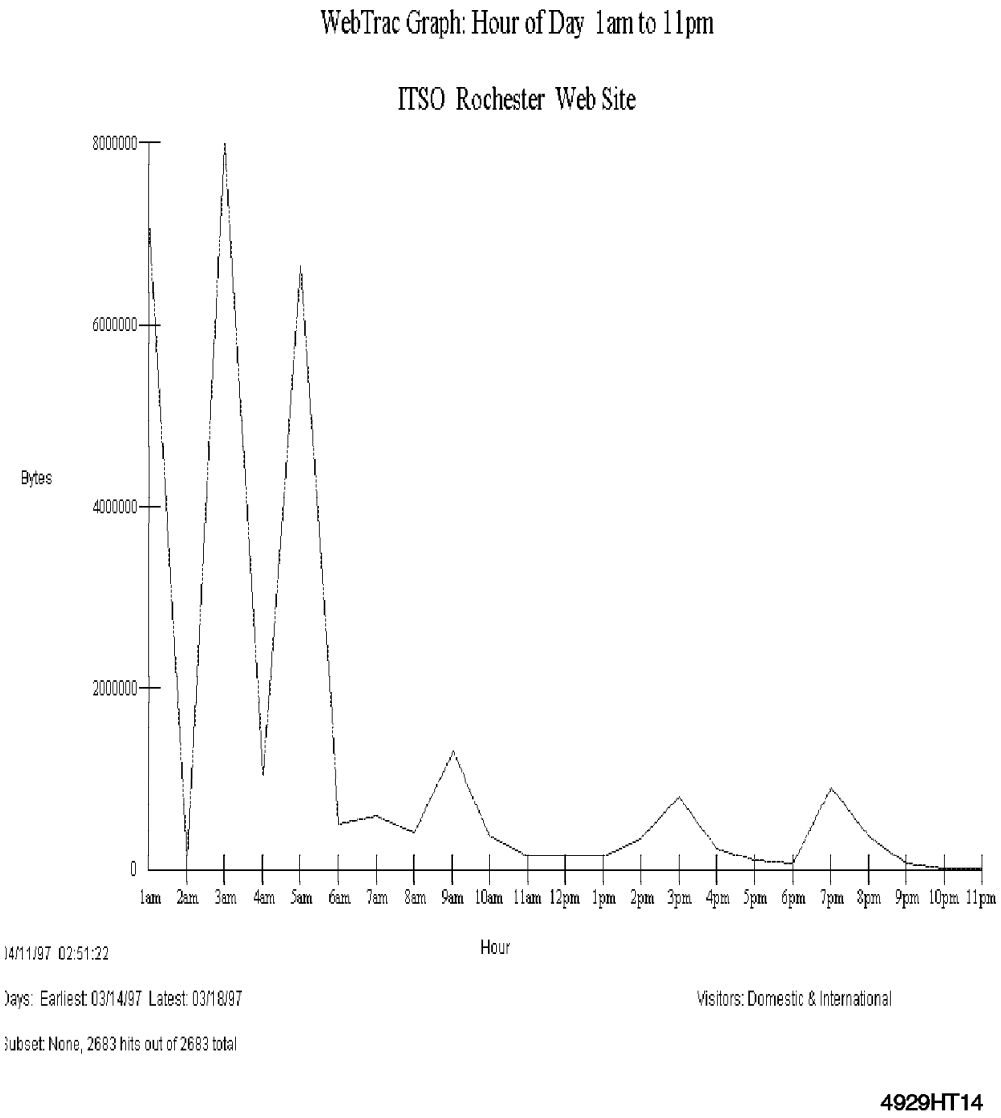


Figure 75. Bytes Transmitted in One-Hour Intervals

You might be interested in the HTTP server access log information from a management point of view or purely from a security point of view. From a security view point, it is important to gather baseline information and watch for abnormal patterns on a regular basis. For example, an abnormally high number of hits from one particular site might indicate suspicious circumstances. Figure 76 on page 103 shows the top five visitors to our Web site.

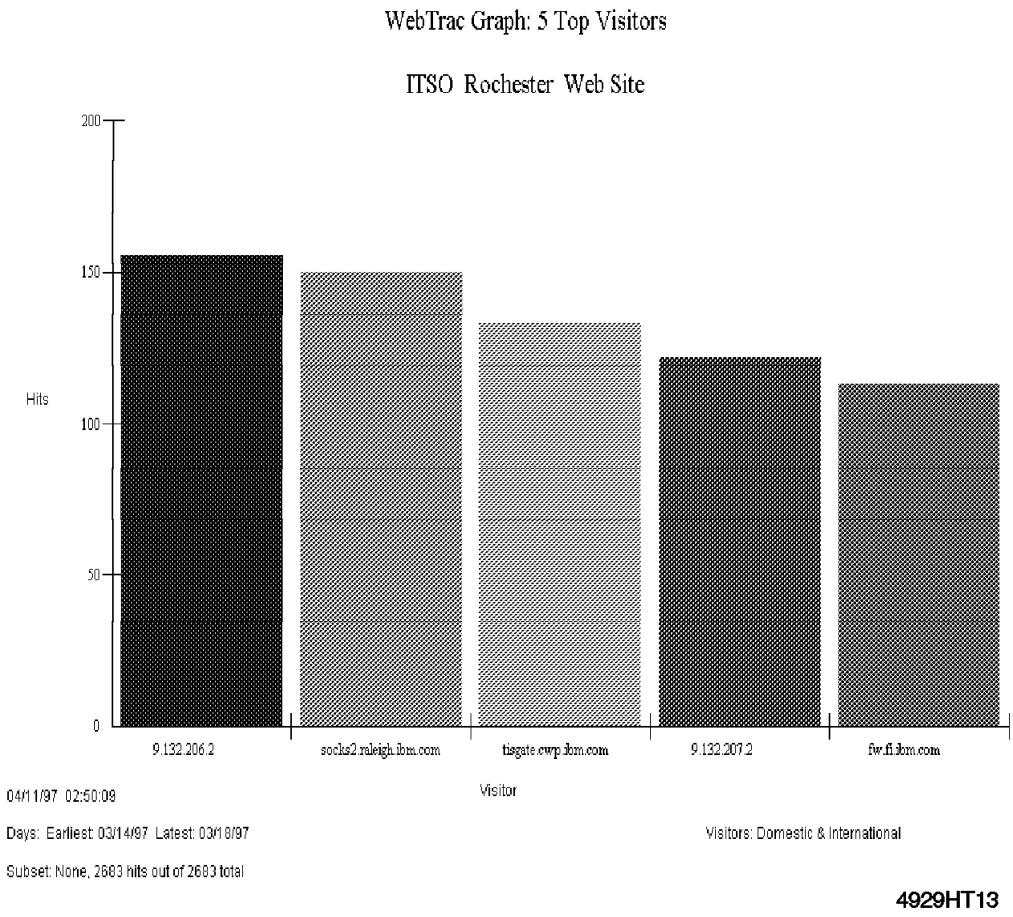


Figure 76. Five Top Visitors

3.4.7 Summary

To summarize our recommendations:

- Server configuration and management:
  1. Tightly control users with *IOSYSCFG special authority.
  2. The HTTP server configuration file QUSRSYS/QATMHTTPC and the HTTP attribute file QUSRSYS/QATMHTTP must be secured and access to these files must be monitored.
  3. Only the webmaster configures the server and uploads HTML pages and CGI-BIN programs.
  4. No compilers should be allowed on the server.
  5. Do **not** start unwanted services.
  6. Minimize the number of user profiles on the server.
  7. Restrict the access of local users to the server.
- Server directives:
  1. PASS controls which files can be accessed.
  2. Use MAP and PASS to provide an alias for the file location and hide the server's file and library structure.
  3. EXEC controls which cgi-bin programs can be run.
  4. Keep all the HTML pages to be served by the server under a single directory tree.
  5. Keep all of the cgi-bin programs in a single library and do not mix them with other programs.
- User profiles:
  1. QTMHHTTP controls HTML access.
  2. QTMHHTP1 controls running cgi-bin programs.
  3. The QTMHHTTP and QTMHHTP1 user profiles must:
     - Have a password value of *NONE.
     - Have user class = *USER.
     - Not be member of a group profile.
     - Not be used as a group profile.
     - Not have special authority.
  4. QTMHHTTP and QTMHHTP1 should **not** have authority to objects that are not to be served to the Internet.
- Monitor and auditing:
  1. Audit critical objects and regularly verify that your security policies are met.
  2. Regularly analyze the data in the access and error log files.
- Connectivity:
  1. Consider using a separate machine for serving.
  2. Consider using an Internet firewall.

Chapter 4. 5250-to-HTML Workstation Gateway Security

Most World Wide Web (WWW) servers today require that you write scripts or programs to create interactive forms and applications. This means that you have to learn new tools and procedures if you want to support the WWW. Not so for AS/400 customers. The AS/400 Workstation Gateway function of the Internet Connection for AS/400 (IC/400) allows you to use your current development tools to create WWW applications. AS/400 WWW applications help you use the worldwide reach of the Internet to open new marketing opportunities. Even existing AS/400 applications run over the WWW with little or no modification. You do not even have to run a conversion program. Just install and configure Internet Connection for AS/400, and the WWW applications on the AS/400 system are ready to go!

You should, however, be aware of a difference in how WSG and 5250 terminals work. The WSG sends a display (a transmission) to the browser and takes down the session. The 5250 terminal has a session with the program where transactions are sent back and forth between the 5250 terminal and the program until the user finishes the job. This difference may call for some modifications.

4.1 Potential Exposures versus Benefits

Using the IC/400 WSG, it is possible to quickly enable an existing AS/400 application to function over the Internet. Functionally, the WSG is similar to a Telnet session. The main difference is that the WSG sends a 5250 screen to a browser rather than to a 5250 emulation program. This provides the benefit of not having to be too concerned about client software. Although the results vary somewhat, almost any browser software works with the WSG server.

The main security exposure is also similar to a Telnet session. User IDs and passwords are transmitted in the clear. Since WSG does not provide for encryption, there is also no privacy of information. One way to prevent the detection of user IDs and passwords is to implement an anonymous WSG application as discussed in Section 4.2, “Anonymous Workstation Gateway Configuration.”

4.2 Anonymous Workstation Gateway Configuration

Because WSG does not support encryption, we do not recommend using it across the Internet for applications that require user authentication (user ID and password). For these applications, you should consider a product that supports encryption such as I/NET′s Webulator discussed in Section 9.3.6, “Example 3 - WEBULATOR” on page 227. We recommend that you use WSG with anonymous access the same way you use anonymous FTP for applications that you want to make public and for which user authentication is not required.

Complete the following steps to implement an anonymous WSG application:

1. Set the WSG attributes.
2. Create the logon exit program.
3. Register the logon exit program.
4. Create the user profile that the exit program uses.

5. Modify the 5250 application (if needed) and test from a browser.

4.2.1 Set the WSG Attributes

Use the Change WSG Attributes (CHGWSGA) command shown in Figure 77 to set Display sign on panel to *NO. This prevents a sign-on display from being shown.

4.2.1.1 Workstation Gateway Attributes

The Inactivity timeout value specifies the number of minutes the system allows a Workstation Gateway session to remain inactive before it is ended (see Figure 77).

The Display sign-on panel value specifies whether to allow the AS/400 sign-on display to be shown when a Workstation Gateway request comes in from a World Wide Web (WWW) browser.

Set the Inactivity timeout attribute to something that is reasonable. In other words, set it to what you allow as the maximum amount of time that a user might pause between entering data.

```
                        Change WSG Attributes (CHGWSGA)

Type choices, press Enter.

Autostart  . . . . . .*YES       *YES, *NO, *SAME
Number of clients per server . . 20        1-50, *SAME, *DFT
Inactivity timeout . . . . . . . 10        0-60 minutes, *SAME,
Data request timeout . . . . . . 10        1-1200 seconds, *SAME,
Display sign on panel  . . . . .*NO        *SAME, *NO, *YES
Access logging . . . . . .*YES       *SAME, *NO, *YES
Top banner URL . . . . . .*NONE



Bottom banner URL  . . . . . .   *NONE
```

Figure 77. Set WSG Attributes

4.2.2 Create the Logon Exit Program

Obviously, you use this technique for applications that you want to make public and where user authentication is not required.

An application logon exit program (registered at the exit point QIBM_QTMT_WSG using the Work with Registration Information, WRKREGINF, command) allows bypassing the AS/400 sign-on display and invoking an application program directly without the client browser having to send a user profile or password. This allows you the option of providing any application to client browsers without requiring a sign on.

The WSG server uses the output of the logon exit program and performs the sign-on action on behalf of the browser. When the exit program is given control, it may perform any desired validation using the supplied IP address and any of the information extracted from the string in the URL. Setting the ″Allow Operation″ output determines whether the automatic logon is performed, or whether an error message is returned to the browser. If the operation is

allowed, the exit program returns the user profile, password, current library, and initial program.

See Section 4.2.4, “Create WSG Logon User Profile” on page 108 for information about the user profile.

4.2.2.1 Logon Exit Program

An application logon exit program (to be registered in exit point QAPP0100) allows bypassing the AS/400 sign-on display and invoking an application program directly without the client browser having to send a user ID or password. For this example, the exit program provides the following values:

- User Name
- Password
- Current Library
- Initial Program
- Return URL

The first four values are fixed values returned to the AS/400 system from the exit program shown in Figure 81 on page 110. The return URL value is used to direct the browser when the WSG session is ended. In our example, the browser is pointed back to the Web Server at WWW.HARM.COM.

It is not a security issue, but if no Return URL is specified in the exit program, the browser user receives the message shown in Figure 78.

Figure 78. WSG Session End Message

You can visit the following Web site for information on AS/400 networking capabilities and click on the link to the HTML Gateway for a sample of a logon exit program that shows this technique.

http://www.as400.ibm.com/products/internet.htm

Also refer to the logon exit program listing in Figure 81 on page 110.

4.2.3 Register the Logon Exit Program

Only one exit program can be registered for the exit point QAPP0100. The book *TCP/IP Configuration and Reference*, contains information about the registration of this exit point. Use the Work with Registration Information (WRKREGINF) command to register the logon exit program.

Locate the entry called:

```
QIBM_QTMT_WSG    QAPP0100     *YES     WSG Server Sign-On Validation
```

Select option 8 and enter the program and library names.

```
                          Work with Exit Programs

Exit point:   QIBM_QTMT_WSG            Format:   QAPP0100

Type options, press Enter.
  1=Add   4=Remove   5=Display   10=Replace

              Exit
             Program     Exit
Opt           Number     Program        Library

                  1     EXIT2          ITSOIC400
```

Figure 79. Registering the WSG Logon Exit Program

Figure 81 on page 110 shows the source code of the logon exit program EXIT2.

4.2.4 Create WSG Logon User Profile

The user profile that is used by the logon exit program should be configured to give the minimum authority needed to use the desired 5250 application. The following display is a subset of the user profile parameters.

```
                        Display User Profile - Basic

User profile . . . . . . . . . . . . . . . :   SUNRISE

Previous sign-on . . . . . . . . . . . . . :   02/04/97  09:17:43
Status . . . . . . . . . . . . . . . . . . :   *ENABLED
Set password to expired  . . . . . . . . . :   *NO
User class . . . . . . . . . . . . . . . . :   *USER
Special authority  . . . . . . . . . . . . :   *NONE
Group profile  . . . . . . . . . . . . . . :   *NONE
Owner  . . . . . . . . . . . . . . . . . . :   *USRPRF
Group authority  . . . . . . . . . . . . . :   *NONE
Group authority type . . . . . . . . . . . :   *PRIVATE
Supplemental groups  . . . . . . . . . . . :   *NONE
Current library  . . . . . . . . . . . . . :   *CRTDFT
Initial program  . . . . . . . . . . . . . :   *SIGNOFF
  Library  . . . . . . . . . . . . . . . . :
Initial menu . . . . . . . . . . . . . . . :   YOURMENU
  Library  . . . . . . . . . . . . . . . . :     YOURLIB
Limit capabilities . . . . . . . . . . . . :   *YES
Text . . . . . . . . . . . . . . . . . . . :

Display sign-on information  . . . . . . . :   *SYSVAL
Limit device sessions  . . . . . . . . . . :   *SYSVAL
Job description  . . . . . . . . . . . . . :   QDFTJOBD
  Library  . . . . . . . . . . . . . . . . :     QGPL
Message queue  . . . . . . . . . . . . . . :   SUNRISE
  Library  . . . . . . . . . . . . . . . . :     QUSRSYS
Message queue delivery . . . . . . . . . . :   *NOTIFY
Message queue severity . . . . . . . . . . :   00
Output queue . . . . . . . . . . . . . . . :   *WRKSTN
  Library  . . . . . . . . . . . . . . . . :
Attention program  . . . . . . . . . . . . :   *NONE
  Library  . . . . . . . . . . . . . . . . :
```

Figure 80. User Profile for the Anonymous WSG Application

4.2.5 Modify (if Necessary) and Test the 5250 Application

The following items should be considered.

- All possible function keys for undesirable effects
- Attention and SysRq key processing
- URL request string processing:

  Whether the action is intentional (hackers) or unintentional, you need to make sure that incorrect values that are sent to the logon exit program are not allowed to cause undesirable results.

For our implementation example, we modified the Sample ″C″ User Exit Program source file that we retrieved from the following Web location:

`http://www.as400.ibm.com/products/wsg/exit2.htm`

Modifications are highlighted from statement 30100 in Figure 81 on page 110.

See the redbooks *Cool Title About the AS/400 and Internet* and *Unleashing AS/400 Applications on the Internet* for detailed information on how to enable WSG operations.

```
5716PW1 V3R7M0  961108                SEU SOURCE LISTING                          02/06/97 10:24:44               PAGE    1
SOURCE FILE . . . . . .  ITSOIC400/QCSRC
MEMBER  . . . . . . . .  EXIT2
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100 /*********************************************************************/
  200 /*                           ** NOTE **                              */
  300 /* This material contains programming source code for your           */
  400 /* consideration. These examples have not been thoroughly tested     */
  500 /* under all conditions. IBM, therefore, cannot guarantee or imply   */
  600 /* reliability, serviceability, performance or function of these     */
  700 /* programs.  All programs contained herein are provided "AS IS".    */
  800 /* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A       */
  900 /* PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.                      */
 1000 /*********************************************************************/
 1100 #define _QTMTLGNEXT_C
 1200
 1300 /*********************************************************************/
 1400 /* All file scoped includes go here                                  */
 1500 /*********************************************************************/
 1600 #include <stdio.h>
 1700 #include <string.h>
 1800 #include <stdlib.h>
 1900
 2000 #ifdef __ILEC400__
 2100 #include <qusec.h>         /* Include for API error code stRUCT  ruct */
 2200 #else
 2300 #include <qusec.cleinc>    /* Include for API error code stRUCT  ruct */
 2400 #endif
 2500
 2600 /*********************************************************************/
 2700 /* All file scoped Constants go here                                 */
 2800 /*********************************************************************/
 2900 #define  SIZE    10
 3000 #define  FNAME   21              /* Qualified database file name size */
 3100 #define  FWIDTH  240             /* Width of one database file record */
 3200 #define  BLANK   ' '
 3300 #define  EQ      ==
 3400 #define  NEQ     !=
 3500
 3600 /*********************************************************************/
 3700 /* All file scoped type declarations go here                          */
 3800 /*********************************************************************/
 3900 /* Structure for data passed to Server Logon exit program.           */
 4000 typedef struct
 4100 {
 4200   char *OperSpecInfo_p;         /* Operation Specific Info    (Input) */
 4300   int  Lgth_OperSpecInfo;       /* Operation Spec Info length (Input) */
 4400   char ClientIPaddr[15];        /* Client IP Addr.            (Input) */
 4500   int  CCSID;                   /* CCSID of operation info    (Input) */
 4600   char AllowOper[1];            /* Allow Operation '0'=N,'1'=Y(Output)*/
 4700   char UserProfile[SIZE];       /* User Profile.             (Output) */
 4800   char Password[SIZE];          /* Password.                 (Output) */
 4900   char ProgramLib[SIZE];        /* Library of program to run.(Output) */
 5000   char ProgramName[SIZE];       /* Program to invoke.        (Output) */
 5100   char InitialMenu[SIZE];       /* Initial menu to invoke.   (Output) */
 5200   char ReturnURL[300];          /* URL upon session close.   (Output) */
 5300 } QAPP0100_I_t;
```

Figure 81 (Part 1 of 7). Source for the Logon Exit Program

```
5716PW1 V3R7M0  961108                SEU SOURCE LISTING                        02/06/97 10:24:44               PAGE    2
SOURCE FILE . . . . . .  ITSOIC400/QCSRC
MEMBER  . . . . . .  EXIT2
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
 5400
 5500 /*********************************************************************/
 5600 /* All file scoped Macro invocations go here                         */
 5700 /*********************************************************************/
 5800
 5900 /*********************************************************************/
 6000 /* All internal function prototypes go here                          */
 6100 /*********************************************************************/
 6200 static void qtmtlgnext
 6300 (char *,int,char *,int,char *,char *,char *,char *,char *,char *,char *);
 6400
 6500 /*********************************************************************/
 6600 /* All file scoped variable declarations go here                     */
 6700 /*********************************************************************/
 6800
 6900 /* Function Specification ********************************************/
 7000 /*                                                                   */
 7100 /* Function Name: Main                                               */
 7200 /*                                                                   */
 7300 /* Descriptive Name: Application Logon exit program sample program.  */
 7400 /*                                                                   */
 7500 /*   This test exit program provides control over signon panels via  */
 7600 /*   the WSG server in the V3R2 release.                             */
 7700 /*                                                                   */
 7800 /*         Consider the method for passing them back to the caller.  */
 7900 /*                                                                   */
 8000 /*   Dependencies:                                                   */
 8100 /*      WSG Applicaton Logon exit point QIBM_QTMT_WSG format QAPPO100 */
 8200 /*       was registered during WSG V3R2 installation.                */
 8300 /*                                                                   */
 8400 /*   Restrictions:                                                   */
 8500 /*                                                                   */
 8600 /*       None                                                        */
 8700 /*                                                                   */
 8800 /*   Messages:                                                       */
 8900 /*                                                                   */
 9000 /*       None                                                        */
 9100 /*                                                                   */
 9200 /*   Side Effects:                                                   */
 9300 /*                                                                   */
 9400 /*       None                                                        */
 9500 /*                                                                   */
 9600 /*   Functions/Macros called:                                        */
 9700 /*                                                                   */
 9800 /*       TRACE - Write one data record to test results file.         */
 9900 /*                                                                   */
10000 /* Input:                                                            */
10100 /*   char * argv[1]        - Operation specific information          */
10200 /*   int    argv[2]        - Length of operation specific information */
10300 /*   char * argv[3]        - IP address of the remote host system.   */
10400 /*   int    argv[4]        - CCSID of the operation specific info    **/
10500 /*   char * argv[5]        - Allow operation '0'=No, '1'=Yes(output) */
10600 /*   char * argv[6]        - User profile to be used (output)        */
```

Figure 81 (Part 2 of 7). Source for the Logon Exit Program

```
5716PW1 V3R7M0  961108                SEU SOURCE LISTING                        02/06/97 10:24:44               PAGE    3
SOURCE FILE . . . . . .  ITSOIC400/QCSRC
MEMBER  . . . . . .  EXIT2
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
10700 /*   char * argv[7]        - Password to be used (output)            */
10800 /*   char * argv[8]        - Program library to be used (output)     */
10900 /*   char * argv[9]        - Program name to be used (output)        */
11000 /*   char * argv[10]       - Menu panel to be used (output)          */
11100 /*   char * argv[11]       - Return URL when session closed (output) */
11200 /*                                                                   */
11300 /* Exit Normal: Return AllowOper value to server application.        */
11400 /*                                                                   */
11500 /* Exit Error: None                                                  */
11600 /*                                                                   */
11700 /* End Function Specification ***************************************/
11800 void main(int argc, char *argv[])
11900 {
12000   /*****************************************************************/
12100   /* Code                                                          */
12200   /*****************************************************************/
12300 #ifdef DEBUG
12400   printf("\n");
12500   printf("main: >>>>> entry\n");
12600   printf("main: WSG Logon Exit, number of inputs is: %d\n", argc);
12700   printf("main: OperSpecInfo      = >%s<\n", argv[1]);
12800   printf("main: Lgth_OperSpecInfo = %d\n", *((int *)(argv[2])));
12900   printf("main: ClientIPaddr      = >%s<\n", argv[3]);
13000   printf("main: CCSID             = %d\n", *((int *)(argv[4])));
13100   printf("main: AllowOper         = '%c'\n",  *argv[5]);
13200   printf("main: UserProfile       = >%.10s<\n", argv[6]);
13300   printf("main: Password          = >%.10s<\n", argv[7]);
13400   printf("main: ProgramLib        = >%.10s<\n", argv[8]);
13500   printf("main: ProgramName       = >%.10s<\n", argv[9]);
13600   printf("main: InitialMenu       = >%.10s<\n", argv[10]);
13700   printf("main: URL               = >%.300s<\n", argv[11]);
13800 #endif
13900
14000   qtmtlgnext(argv[1],
14100              *((int *)(argv[2])),
14200              argv[3],
14300              *((int *)(argv[4])),
14400              argv[5],
14500              argv[6],
14600              argv[7],
14700              argv[8],
14800              argv[9],
14900              argv[10],
15000              argv[11]);
15100
15200 #ifdef DEBUG
15300   printf("main: <<<<< exit\n");
15400   printf("\n");
15500 #endif
15600   return;
15700 }                                                  /* End main */
15800
15900
```

Figure 81 (Part 3 of 7). Source for the Logon Exit Program

```
5716PW1 V3R7M0  961108                SEU SOURCE LISTING                        02/06/97 10:24:44               PAGE    4
SOURCE FILE . . . . . .  ITSOIC400/QCSRC
MEMBER  . . . . . .  EXIT2
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
16000 /* Function Specification ********************************************/
16100 /*                                                                   */
16200 /* Function Name: qtmtlgnext                                         */
16300 /*                                                                   */
16400 /* Descriptive Name: Workstation Gateway Server (WSG) Logon exit.    */
16500 /*                                                                   */
16600 /*   This test exit program provides control over user authentication */
16700 /*   to a workstation gateway in the V3R2 release.                   */
16800 /*                                                                   */
16900 /* Notes:                                                            */
17000 /*                                                                   */
17100 /*   Dependencies:                                                   */
17200 /*                                                                   */
17300 /*       Workstation Gateway Logon exit point QIBM_QTMT_WSG was      */
17400 /*        registered during WEB V3R2 installation.                   */
17500 /*                                                                   */
17600 /*   Restrictions:                                                   */
17700 /*                                                                   */
17800 /*       None                                                        */
17900 /*                                                                   */
18000 /*   Messages:                                                       */
18100 /*                                                                   */
18200 /*       None                                                        */
18300 /*                                                                   */
18400 /*   Side Effects:                                                   */
18500 /*                                                                   */
18600 /*       None                                                        */
18700 /*                                                                   */
18800 /*   Functions/Macros called:                                        */
18900 /*                                                                   */
19000 /*       None                                                        */
19100 /*                                                                   */
19200 /* Input:                                                            */
19300 /*  char * OperSpecInfo_p     - Operation Specific Information.      */
19400 /*  int    Lgth_OperSpecInfo  - Length (in bytes) of Operation       */
19500 /*                              Specific Information.                */
19600 /*  char   ClientIPaddr       - Client Internet Protocol Address.    */
19700 /*  int    CCSID              - CCSID of operation info              */
19800 /*                                                                   */
19900 /* Output:                                                           */
20000 /*  char * AllowOper          - Allow Operation ('0' = Reject),      */
20100 /*                                              ('1' = Accept).      */
20200 /*  char * UserProfile        - User Profile to be used for sign on. */
20300 /*  char * Password           - Password to be used for sign on.     */
20400 /*  char * ProgramLib         - Library of program to invoke.        */
20500 /*  char * ProgramName        - Name of program to invoke.           */
20600 /*  char * InitialMenu        - Initial menu to invoke.              */
20700 /*  char * URL                - Return URL when session closed.      */
20800 /*                                                                   */
20900 /* Exit Normal: (See OUTPUT)                                         */
21000 /*                                                                   */
21100 /* Exit Error:  None                                                 */
21200 /*                                                                   */
```

Figure 81 (Part 4 of 7). Source for the Logon Exit Program

```
5716PW1 V3R7M0  961108                SEU SOURCE LISTING                          02/06/97 10:24:44               PAGE    5
SOURCE FILE . . . . . .  ITSOIC400/QCSRC
MEMBER  . . . . . .  EXIT2
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
21300 /* End Function Specification ****************************************/
21400 static void qtmtlgnext(char *OperSpecInfo_p,          /* Entry point */
21500                        int Lgth_OperSpecInfo,
21600                        char ClientIPaddr[15],
21700                        int  CCSID,
21800                        char AllowOper[1],
21900                        char UserProfile[SIZE],
22000                        char Password[SIZE],
22100                        char ProgramLib[SIZE],
22200                        char ProgramName[SIZE],
22300                        char InitialMenu[SIZE],
22400                        char URL[300])
22500 {
22600
22700
22800   /*********************************************************************/
22900   /* Closing URL's for applications                                    */
23000   /*                                                                   */
23100   /* These strings should be compiled to CCSID 500 to insure they      */
23200   /* translate properly when converted from EBCDIC -> ASCII.           */
23300   /*                                                                   */
23400   /* If you are not sure what CCSID is used for compiling, then        */
23500   /* stick with the invariant character set.                           */
23600   /*********************************************************************/
23700   /*                                                                   */
23800   /* Invariant characters:  characters guaranteed to be at the same    */
23900   /* code points across all EBCDIC CCSID:                              */
24000   /*                                                                   */
24100   /*       A-Z, 0-9, + < = > % & * ' ( ) , _ - . / : ; ? "             */
24200   /*                                                                   */
24300   /* 0-9  0xF0-0xF9   +  0x4E   % 0x6C    (  0x4D   -  0x60   ;  0x5E */
24400   /* A-I  0xC1-0xC9   <  0x4C   & 0x50    )  0x5D   .  0x4B   ?  0x6F */
24500   /* J-R  0xD1-0xD9   =  0x7E   * 0x5C    ,  0x6B   /  0x61   "  0x7F */
24600   /* S-Z  0xE2-0xE9   >  0x6E   ' 0x7D    _  0x6D   :  0x7A          */
24700   /*********************************************************************/
24800   /* There is on exception to this rule - Japanese Kanji/Katakana      */
24900   /* CCSID 5026 (1172 CS, 290 CP).  The lower case characters only     */
25000   /* are NOT invariant.                                                */
25100   /*********************************************************************/
25300   char *pszRushURL   = "http://www.harm.com";
25600   char *pszURL;
25700
25800   /*********************************************************************/
25900   /* Can't trust pointer, can only trust length value                  */
26000   /*********************************************************************/
26100   if (0 == Lgth_OperSpecInfo) {
26200     OperSpecInfo_p = "";
26300   } /* endif */
26400
26500   /*********************************************************************/
```

Figure 81 (Part 5 of 7). Source for the Logon Exit Program

```
5716PW1 V3R7M0  961108                SEU SOURCE LISTING                          02/06/97 10:24:44               PAGE    6
SOURCE FILE . . . . . .  ITSOIC400/QCSRC
MEMBER  . . . . . .  EXIT2
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
26600   /* Echo feature - send in a URL and I'll use it upon close          */
26700   /*******************************************************************/
26800   if (Lgth_OperSpecInfo &&
26900       (!strncmp("http://", OperSpecInfo_p, 7) ||
27000        !strncmp("HTTP://", OperSpecInfo_p, 7))) {
27100     memcpy(UserProfile, "          ", SIZE);
27200     memcpy(Password,    "          ", SIZE);
27300     memcpy(ProgramLib,  "          ", SIZE);
27400     memcpy(ProgramName, "          ", SIZE);
27500     memcpy(InitialMenu, "          ", SIZE);
27600     strcpy(URL, OperSpecInfo_p);
27700     memcpy(AllowOper  , "1", 1);
27800   } else if (!strncmp(OperSpecInfo_p, "SIGNON",6) ||
27900              !strncmp(OperSpecInfo_p, "signon",6)) {
28000     memcpy(UserProfile, "          ", SIZE);
28100     memcpy(Password,    "          ", SIZE);
28200     memcpy(ProgramLib,  "          ", SIZE);
28300     memcpy(ProgramName, "          ", SIZE);
28400     memcpy(InitialMenu, "          ", SIZE);
28500     strcpy(URL, pszDilbert);
28600     memcpy(AllowOper  , "1", 1);
28700   } else if (!strncmp(OperSpecInfo_p, "REJECT",6) ||
28800              !strncmp(OperSpecInfo_p, "reject",6)) {
28900     memcpy(UserProfile, "REJECTED  ", SIZE);
29000     memcpy(Password,    "REJECTED  ", SIZE);
29100     memcpy(ProgramLib,  "REJECTED  ", SIZE);
29200     memcpy(ProgramName, "REJECTED  ", SIZE);
29300     memcpy(InitialMenu, "REJECTED  ", SIZE);
29400     strcpy(URL, pszRushURL);
29500     memcpy(AllowOper  , "0", 1);
29600   /*******************************************************************/
29700   /* Applications                                                    */
29800   /*******************************************************************/

29900    } else if (!strncmp(OperSpecInfo_p, "SUNRISE",7) ||

30000              !strncmp(OperSpecInfo_p, "sunrise",7)) {
30100     memcpy(UserProfile, "SUNRISE   ", SIZE);
30200     memcpy(Password,    "TEST1     ", SIZE);
30300     memcpy(ProgramLib,  "ITSOIC400 ", SIZE);
30400     memcpy(ProgramName, "SHTMLR    ", SIZE);
30500     memcpy(InitialMenu, "          ", SIZE);
30600     strcpy(URL, pszRushURL);
30700     memcpy(AllowOper  , "1", 1);
30800  }
30900
31000 #ifdef DEBUG
31100   printf("qtmtlgnext: Set URL: >%s<\n", URL);
31200   printf("qtmtlgnext: <<<<< exit\n");
31300 #endif
31400
31500   return;
31600 }
31700
31800 #undef _QTMTLGNEXT_C
```

Figure 81 (Part 6 of 7). Source for the Logon Exit Program

```
5716PW1 V3R7M0  961108                SEU SOURCE LISTING                          02/06/97 10:24:44               PAGE    7
SOURCE FILE . . . . . .  ITSOIC400/QCSRC
MEMBER  . . . . . .  EXIT2
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
                            * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 81 (Part 7 of 7). Source for the Logon Exit Program

4.3 Anonymous Workstation Gateway Example

Review Chapter 2, "Start Here by Securing OS/400®" on page 37 to make sure system security is configured properly and that no unnecessary TCP servers are running.

The example that we show here allows users on the Internet to access a single AS/400 application while preventing them from getting to a sign-on display. For our test site, we created a link on an HTML page on the HTTP server at WWW.HARM.COM to a WSG server at WWW.HARM2.COM. There was no particular reason to do this other than to demonstrate the ease of linking resources. Another way of starting the WSG session is to point the browser to the correct URL. The HTML source on the HTTP server page is shown in Figure 82.

```
<HTML>
<HEAD>
   <TITLE>WSG LINK - Welcome</TITLE>

<H3>Welcome To:</H3><BR>
<H2>The Anonymous Work Station Gateway Application</H2>

<A HREF="http://WWW.HARM2.COM:5061/WSG/QAPP0100?sunrise">Personnel
 Information

</BODY>
</HTML>
```

Figure 82. HTML Source for the WWW.HARM.COM IC/400 HTTP Server

Notice that the URL contains the name of the exit point as well as a variable that the logon exit program uses at execution time. Later, you see how we instruct the AS/400 system to use the exit program.

To show how this might appear to a user, see the following browser displays. From the first display that is on server WWW.HARM.COM, the user clicks on the link Personnel Information.

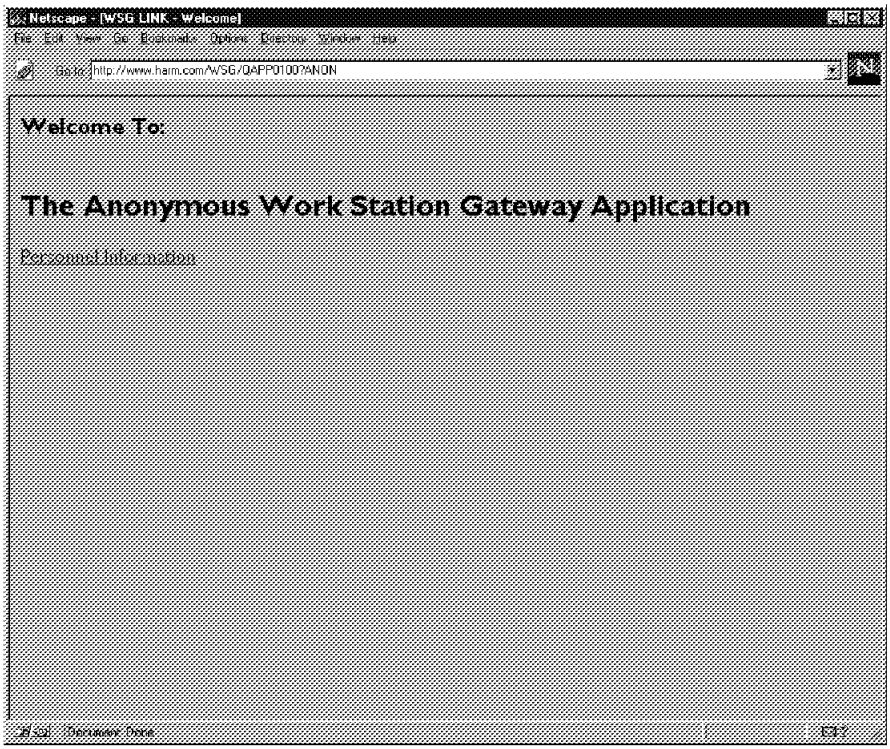


Figure 83. Anonymous WSG Logon Example

This action creates a connection to the WSG server on WWW.HARM2.COM. When the WSG session starts, the AS/400 system exits the normal session bring up sequence and executes the logon exit program. The logon exit program passes the following variables to the sign-on process.

- USERID=SUNRISE
- PASSWORD=TEST1
- PROGRAMLIB=ITSOIC400
- PROGRAMNAME=SHMTLR

Tip

This user ID and password are in a compiled program and cannot be viewed with an editor. They **are** viewable in the source file that should be secured.

An automatic sign on to the system occurs with no chance for the user to guess a user ID or password and no capability to specify a different library, menu, or program. We call this process an anonymous application because all users who invoke the WSG get exactly the same results.

Users will probably get the following message since more than one browser can be accessing the application simultaneously.

```
                          Display Program Messages
Job 056050/SUNRISE/QPADEV0012 started on 02/04/97 at 11:11:06 in
subsystem. Message queue SUNRISE is allocated to another job.
 Press Enter to continue.
```

Figure 84. Display Program Message

Next, the application display is sent to the browser rather than a sign-on display.

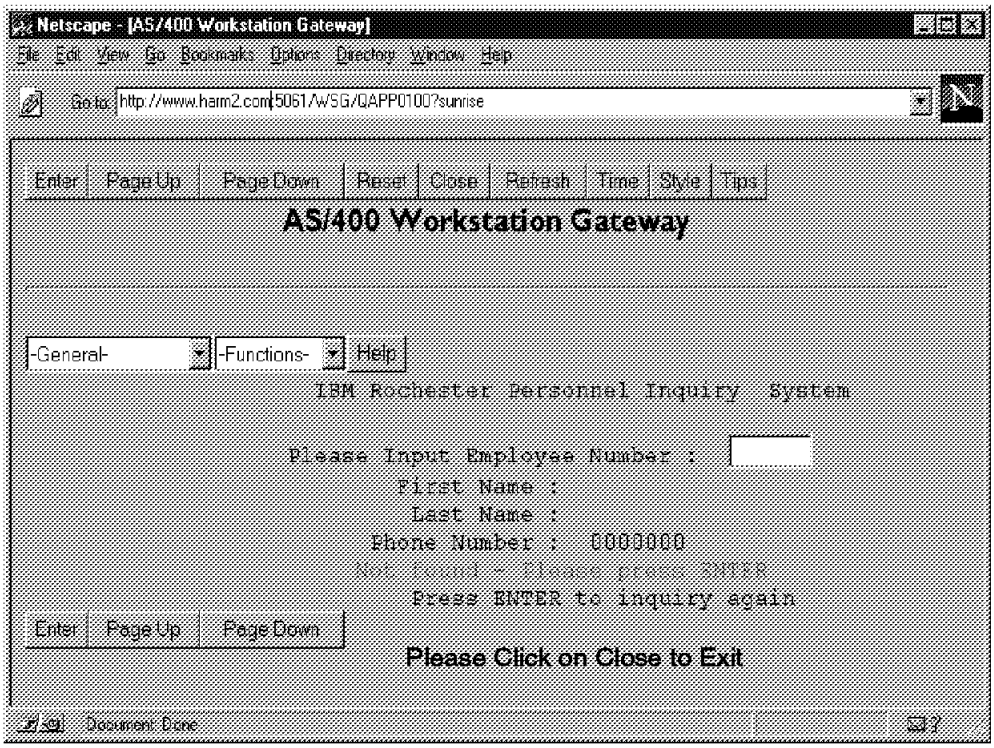


Figure 85. Anonymous WSG Application Example

We do not explain application program design here but want to offer these suggestions. On Page 105, we stated that existing AS/400 applications run over the WWW with little or no modifications. Although that is true for a non-secure implementation of WSG, some modification may be necessary for a secure anonymous WSG scenario. For example, if your 5250 application allows an exit to the command line, you need to modify your code to disable or alter that function. Some additional things to keep in mind are to keep the display simple; do not enable function keys unless absolutely necessary, and verify handling of I/O errors.

In this example, the user needs only to be able to input a number and click on Enter. The desired way to exit the application is to click on Close. If the user clicks on the Close button as intended, the application ends, program control returns to the exit program, and the browser is directed to a URL provided by the logon exit program.

If the user just closes the browser window or links to some other Web site, the WSG session times out in the number of minutes that have been set in the WSG *Inactivity timeout* attribute. If the user tries to get a sign-on display by altering the URL and resending the request as shown in the following figure, the logon exit program causes the request to fail and returns the message as illustrated by the browser window in Figure 86 on page 119.

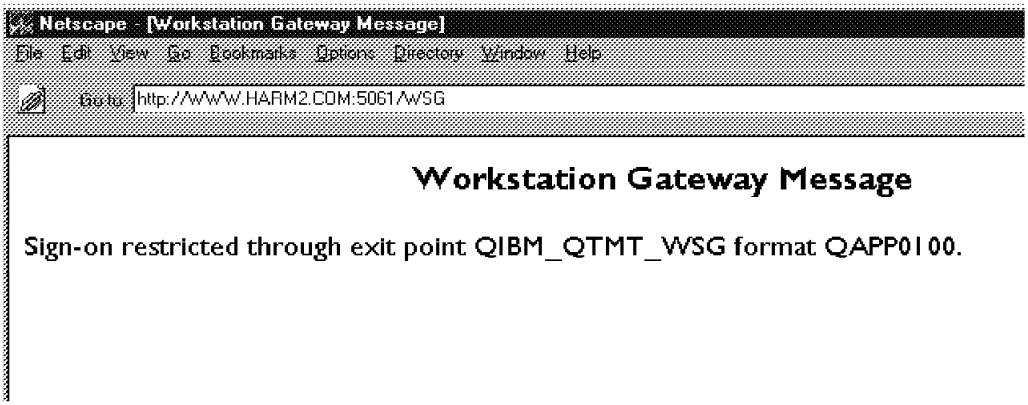


Figure 86. Using Exit Point QIBM_QTMT_WSG to Prevent Sign-on Window

4.4 Summary

To summarize:

- Do **not** use passwords from the Internet: disable sign-on with CHGWSGA DSPSGN(*NO).
- Provide exit program to select user profile (for example, ANYWSGUSER).
- Strictly limit access of Workstation Gateway users.
- Log access through the gateway using CHGWSGA ACCLOG(*YES).
- Limit the authority of the user profile used to run the WSG session to only the resources that you want to make available through the Internet. Make sure this profile does not belong to a group profile or has special authorities.

Chapter 5. Electronic Mail Security

Electronic mail (or e-mail) is often the first Internet service companies want to bring to their users. E-mail enables your users to exchange information with other companies and users around the world that often makes e-mail (from the user′s point-of-view) the most important reason to connect your AS/400 system to the Internet. With so many people on the Internet, and all of them having e-mail addresses, this is a service that you must implement sooner or later.

Until recently, the AS/400 e-mail support was limited to OfficeVision/400™ providing the ability to handle SNA-based text messages with only limited support for Internet e-mail. In V3R2 and V3R7, the AS/400 system′s e-mail capabilities were greatly expanded:

- The AS/400 system as a mail server can accept mail from external hosts or send it to external hosts through SMTP.
- AnyMail/400 mail server framework acts as a delivery agent and user exits or **snap-ins** put the mail in the correct mailbox in the local AS/400 system.
- The e-mail clients supported are Internet′s Post Office Protocol Version 3 (POP3), OV/400 clients, and Client Access/400 for Windows 95/NT (which includes Lotus Mail formerly known as cc:Mail for the Internet).

Because it talks directly to the external world, the mail server is vulnerable to attacks. In this chapter, we explore the main risks for the AS/400 system as a mail server connected to the Internet and we discuss the measures that you can take to minimize the risks and recover from an e-mail attack.

We assume that you are familiar with the details on how to configure SMTP, POP server, and OV/400. For information on implementation of mail on the AS/400 system, refer to *TCP/IP Configuration and Reference, E-Mail Capabilities with OS/400 V3R2 and V3R7, OV/400*.

5.1 SMTP, POP3, MIME, and SMTP-to-SNA Gateway Support

The core of the AS/400 e-mail support is SMTP/POP3 mail servers with MIME (Multipurpose Internet Mail Extensions) support. This support is part of OS/400 V3R7 and provided by PTF SF34433 and PTF SF34435 for V3R2.

The SMTP server allows your AS/400 system to exchange mail with other SMTP servers on the Internet.

The POP3 server allows your AS/400 system to hold mail in a ″mail box″ that a POP3-capable client such as Netscape Navigator can access. The POP server is a simple store-and-forward mail system. It provides electronic mail boxes on the AS/400 system from which the clients can retrieve mail. It uses AnyMail/400 mail server framework and the system distribution directory to process and distribute e-mail. SMTP is used to forward mail.

OS/400 can also accept messages with attachments in MIME format. MIME is an Internet content standard for nontext content such as rich text, images, audio, and video. MIME attachments can be sent to AS/400 POP3 users and even OV/400 users.

Finally, if you have a SNADS network with several AS/400 systems, you need only to install TCP/IP on one of them and that system can serve as a gateway between SNADS and SMTP networks. Refer to the *TCP/IP Configuration and Reference* for information on how to configure the bridge.

5.2 Risks of E-Mail

There are a few risks associated with electronic mail; some examples are forging mail or snooping mail that might contain confidential or private information. But accepting e-mail opens the door to three major exposures that we cover in more detail in this chapter:

- Denial-of-service attacks:

  Incoming mail, if it takes the form of mail bombing, can tie up your computer resources (disk space and processor) to the point where your AS/400 system is put out of commission. Although we worry about this type of attack, in practice, you can probably have similar effects from an ″accident″ such as a chain letter or a few huge images (MIME attachments) sent to your users.

  The symptoms of a denial-of-service attack or inadvertent floods are:

  1. Flooding the disk space.
  2. Overloading the CPU capacity.
  3. Fill up the system distribution directory if the remote users are automatically registered and the mail comes from many new users.

- Downloading viruses:

  A virus is a program that can change other programs to include a copy of itself. The virus program usually performs operations such as taking up system resources or destroying data. The AS/400 architecture prevents any AS/400 object from being replaced by a virus. However, attachments sent in electronic mail can be stored in a shared folder or in the integrated file system of the AS/400 POP3 server and from there, they can be downloaded to other users′ PCs or POP3 clients.

- Snooping on POP3 user ID/password:

  Standard POP clients send the user′s ID and password in the clear; therefore, anyone snooping on the connection can see them. On the AS/400 system, each POP user needs a user profile and directory entry so if someone is able to capture the POP user′s ID and password, they also get the User ID and password of an AS/400 user. If the intruder manages to get hold of a powerful user profile (for example, one with *ALLOBJ special authority), the intruder can cause much damage to your system.

- Snooping on sensitive e-mail:

  You need to think about the exposure of sending sensitive or confidential information over the Internet. Depending on your own environment, you might need to use alternative methods to exchange sensitive information.

5.2.1 Security Solutions

In this section, we describe some of the resources you have on the AS/400 system to minimize the effect of a denial-of-service attack or an ″accident″ that produces similar problems.

5.2.1.1 Auxiliary Storage Threshold Limit

The AS/400 system requires some free disk space to operate. If the disks are filled to the brim, the system stops. To recover from a crash produced by completely filling up the disk is not an easy task and one that you certainly want to avoid. To prevent unwanted objects from flooding your system to the point where it cannot operate, set the auxiliary storage threshold to 90% or lower. The threshold value informs the system when to notify you that the storage pool is almost full.

When the ASP threshold is reached, sending and receiving mail using SNADS or SMTP is halted. It does not restart until the disk utilization once again is below the ASP threshold. A message is sent to the QSYSOPR message queue every hour informing you that the threshold value is exceeded. Some action to reduce the amount of disk storage used is required.

Be aware that OfficeVision/400 **outgoing** mail is still generated even after the ASP threshold is reached.

You can alter the threshold values by using either the system service tools (STRSST) or the dedicated service tools (DST). By following the System Service Tools (SST) menus, we get to the displays shown in Figure 87.

```
                        Work with Disk Configuration

Select one of the following:

     1. Display disk configuration
     2. Add units to ASPs
     3. Work with ASP threshold
     4. Include unit in device parity protection
     5. Enable remote load source mirroring
     6. Disable remote load source mirroring
```

Figure 87 (Part 1 of 2). Using SST to Change ASP Thresholds

```
                          Change Storage Threshold

                           ----Protected---  ---Unprotected--
ASP  Threshold  Overflow        Size  %Used       Size  %Used
  1      90%    No             21346 47.73%          0  0.00%

This is a mirrored ASP.  The threshold represents the amount
of protected storage used before an attention message is sent
to the system operator.

Type choice, press Enter.

  New threshold . . . . . .   86    1-100
```

Figure 87 (Part 2 of 2). Using SST to Change ASP Thresholds

The *Backup and Recovery - Advanced* manual provides more information on how to change ASP thresholds.

5.2.1.2 System Distribution Directory Configuration

If your AS/400 system is expected to route e-mail to other systems in your network, explicitly configure an entry for every valid system in your network and avoid *ANY *ANY entries in the system distribution directory. Without an *ANY *ANY entry, your system rejects mail that is not addressed to a user in a valid system. You can take this a step further and explicitly configure every user also but it probably makes the maintenance cumbersome. Figure 88 shows that we do not have an *ANY *ANY entry in our AS/400 system distribution directory and every system our AS/400 system is expected to route mail to is explicitly configured.

```
                        Work with Directory Entries

Type options, press Enter.
  1=Add      2=Change   4=Remove   5=Display details   6=Print details
  7=Rename   8=Assign different ID to description   9=Add another descr

Opt  User ID   Address   Description

     *ANY      ARG       All Users in Argentina
     *ANY      MSP       All users in Minneapolis
     *ANY      OSLO      All users in Oslo
```

*Figure 88. Avoid *ANY *ANY Entries in the System Distribution Directory*

5.2.1.3 Automatic Registration of Remote Users

For incoming mail, you can specify whether the remote user ID and address are automatically added to the system directory and, if necessary, to an alias table. If your system is being attacked by a mail bomb coming from zillions of remote users, having automatic registration creates zillions of new entries in the system distribution directory.

To avoid this problem, configure automatic registration (AUTOADD) *NO in the SMTP attributes (see Figure 89).

```
                    Change SMTP Attributes (CHGSMTPA)

Type choices, press Enter.

Autostart server . . . . . . . . AUTOSTART      *YES
Retries by minute:              RTYMIN
  Number of retries  . . . . . .                3
  Time interval  . . . . . . . .                30
Retries by day:                 RTYDAY
  Number of retries  . . . . . .                0
  Time interval  . . . . . . . .                0
Retry remote name server . . . . RTYRMTSVR      *NO
Automatic registration . . . . . AUTOADD        *NO
  User ID prefix . . . . . . . . USRIDPFX       QSM
  Address  . . . . . . . . . . . ADDRESS       QSMRMTAD
                                                          More...
```

*Figure 89. Avoid Flooding the System Distribution Directory with AUTOADD *NO*

5.2.1.4 Job Priority for SMTP and Mail Server Framework Jobs

To minimize the impact that the mail-related jobs (mainly SMTP and mail server framework) have on the other jobs on your system, you can lower the run priority (set it to a higher value) for these jobs.

Both of the mail server framework and SMTP jobs run in the QSYSWRK subsystem and they are:

- QMSF (mail server framework job)
- QTSMTPBRCL (SMTP job)
- QTSMTPBRSR (SMTP job)
- QTSMTPCLNT (SMTP job)
- QTSMTPSRVR (SMTP job)

Table 11. MSF and SMTP Jobs Run Priority and Class

Job	QSYSWRK Compare Value	Shipped Class	Shipped Run Priority	Customized Run Priority
QMSF	'ZMFMSF'	QSYSCLS35	35	55
All SMTP Jobs	'*ANY'	QSYSCLS50	50	55

To change the priority for QMSF, you can change the routing entry in QSYSWRK to point to a class with a run priority higher than any batch job in your system. For example:

1. CRTCLS CLS(QUSRSYS/MAIL) RUNPTY(55)
2. CHGRTGE SBSD(QSYSWRK) SEQNBR(2525) CLS(MAIL)
3. ENDMSF
4. STRMSF

Changing the priority for the four SMTP jobs is a little more difficult because changing the class in the "catch all" (sequence number 9999, compare value *ANY) of QSYSWRK might affect other jobs also. If you decide to do it anyway, these are the steps to follow:

1. CHGRTGE SBSD(QSYSWRK) SEQNBR(9999) CLS(MAIL)
2. ENDTCPSVR *SMTP
3. STRTCPSVR *SMTP

5.2.1.5 Preventing Viruses through E-Mail

As we explained in Section 5.2, "Risks of E-Mail" on page 122, the AS/400 architecture makes it unlikely, if not impossible, to infect your AS/400 system with viruses. An AS/400 program cannot arrive disguised as something else. However, PC viruses can arrive in the mail as MIME attachments.

The only solution to this potential problem is to educate your users about the possibility of receiving viruses through e-mail. They should never receive attachments that come from unknown sources. They should immediately report any suspicious piece of mail that arrives, and they should always run the anti-virus program against the PC file received in a folder or IFS as soon as it is received.

5.2.1.6 Hiding the AS/400 User Profiles of the E-Mail Users

You do not have to let the entire world know the user profiles of your AS/400 users that are also e-mail recipients. You can use an alias table to define the SMTP name the mail users in your AS/400 system are known by. This alias is mapped to the real AS/400 user profile. In the (hopefully) unlikely event that some malicious persons can get hold of a sign-on display on your system, they do not have the advantage of knowing valid AS/400 user profiles to try to penetrate your system.

Use the Work with Names for SMTP (WRKNAMSMTP) command to defined the alias. Figure 90 shows an example of an alias table.

```
                          Work with Names for SMTP
                                                                System:   SYSNAM
Alias table type . . . . . . . . :   System


Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   6=Print

Opt     User ID      Address      SMTP Name

        ADAN         SYSNAM       adanmail@SYSNAM.MARLAND.MORRIS.COM
        DON          SYSNAM       arthur@SYSNAM.MARLAND.MORRIS.COM
        PETERSON     SYSNAM       kris@SYSNAM.MARLAND.MORRIS.COM
        SELBACH      SYSNAM       sverre@SYSNAM.MARLAND.MORRIS.COM
```

Figure 90. Using Alias to Hide AS/400 User Profiles

5.2.1.7 Preventing Unauthorized Users from Reading E-Mail

When e-mail is received in your AS/400 system through SMTP, it is stored at some point in the IFS under the /QTCPTMM directory. For a description of the flow and where you can find mail objects under /QTCPTMM, see Section 5.4.1, “Recovering from E-Mail Attack to OV/400 Users” on page 128 and Section 5.4.2, “Recovering from E-Mail Attack to POP Clients” on page 132. You must audit the authority of the /QTCPTMM directory and subdirectories beneath it to make sure that local users that have access to the IFS cannot snoop in other users′ mail. Figure 91 shows the use of the Work with Authority (WRKAUT) command to verify that *PUBLIC is excluded from the /QTCPTMM directory and no explicit authority has been given to users.

```
                            Work with Authority

Object . . . . . . . . . . . . :   /qtcptmm
Owner  . . . . . . . . . . . . :   QTCP
Primary group  . . . . . . . . :   *NONE
Authorization list . . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                   Data     --Object Authorities--
Opt  User        Authority  Exist  Mgt  Alter  Ref

     *PUBLIC     *EXCLUDE
     QTCP        *RWX         X     X     X     X
     QMSF        *X
```

Figure 91. Verifying Authority of QTCPTMM Directory

5.2.1.8 Protecting POP Users User IDs and Passwords

As we discussed in Section 5.2, “Risks of E-Mail” on page 122, every POP3 user must have a valid AS/400 user profile and password to access the POP3 mail box. Since the POP3 clients send the user ID and password in the clear, you may want to set these user profiles as *SIGNOFF profiles. *SIGNOFF profiles have the *Initial program to call*(INLPGM) parameter set to *SIGNOFF so that if somebody sniffing the user ID and password tries to use a 5250 or TELNET session to signon, it will be signed off immeadiately. You must create another user profile for the user if the user also needs access to other applications on the same AS/400 server. Another problem is that a *SIGNOFF user cannot sign-on to the AS/400 system to change the password when it expires and, therefore, you should think of writing some program to automatically change the user′s password.

If your AS/400 mail server is in the secure network and it is also a production system, you might want to restrict the access to the POP3 mail boxes only through the *Intranet* as opposed to using the Internet to access POP3 mail.

5.2.1.9 Preventing Sensitive Mail from Being Snooped

You must educate your users and make sure they are aware that mail sent over the Internet can be snooped. You might provide an alternative to your users such as the possibility of reaching your secure network through SLIP. Dial-up methods are less susceptible to snooping so you can give POP access for traveling employees through SLIP over a switch connection without allowing it across the Internet.

5.3 How to Stop an E-Mail Attack

If you recognize that you are being attacked through electronic mail, the sensible thing to do is to stop e-mail immediately. To do this, end SMTP and the mail server framework:

```
ENDTCPSVR *SMTP
ENDMSF *IMMED
```

Depending on your mail clients and network configuration, you might also end the SNADS subsystem and the POP server:

```
ENDSBS QSNADS
ENDTCPSVR *POP
```

A rather sophisticated but elegant way to do early detection of an attack through e-mail is to measure the average CPU utilization of the jobs that are involved in e-mail (SMTP jobs, mail server framework jobs, QSNADS, and POP server jobs depending on what mail clients you are using). Once you know this average, periodically monitor the actual CPU utilization of these jobs and if it exceeds a threshold that you set, an operator should be notified. You can use automation tools such as Omegamon to perform this task.

5.4 How to Clean Up after the Attack

After you close your AS/400 system to electronic mail, you must repair the damage that the attack might have caused. If you followed the recommendations described in section 5.2.1, “Security Solutions” on page 123, you probably managed to minimize the impact that the attack had on your system

Important

You should ENDTCPSVR *SMTP and ENDMFS *IMMED **before** you start the cleanup process.

In this section, we discuss two scenarios: the first one when the e-mail clients are OV/400 users and the second one when the e-mail clients are POP clients. In real life, you might have a combination of these scenarios.

5.4.1 Recovering from E-Mail Attack to OV/400 Users

In this section, we provide a high-level description of the flow that a piece of mail follows from arriving at your AS/400 system through SMTP to being delivered to an OV/400 user. Knowing this flow makes it easier to understand the recovery process recommended later in this chapter.

Note

The following description should not be considered IBM official information. The interfaces described here are not officially published and, therefore, might be subject to changes in future releases of the AS/400 system.

1. When a piece of mail arrives at the AS/400 system through SMTP, it is stored in the IFS in /QTCPTMM/SMTPBOX.
2. If the recipient is an OV/400 user, the piece of mail is handled by one of the snap-ins of the AnyMail/400 Mail Sever Framework and split into several entities depending on the original piece of mail (envelop, text attached, binary, and MIME note). These entities are in /QTCPTMM/ATTABOX and the original object is removed (unlinked) from SMTPBOX. Note that while the piece of mail is being processed in this step, it is actually temporarily duplicated in terms of disk space.
3. At this point, another snap-in of MSF moves the mail to the OV/400 user incoming mail box.
4. An *ARV entry is added to the Distribution Log. (the DSPDSTLOG command shows the entries).
5. Mail entities are removed (unlinked) from /QTCPTMM/ATTABOX.

Figure 92 on page 129 shows the flow of a piece of mail as it arrives in your AS/400 system through SMTP and where it is stored on its way to the OV/400 recipients.

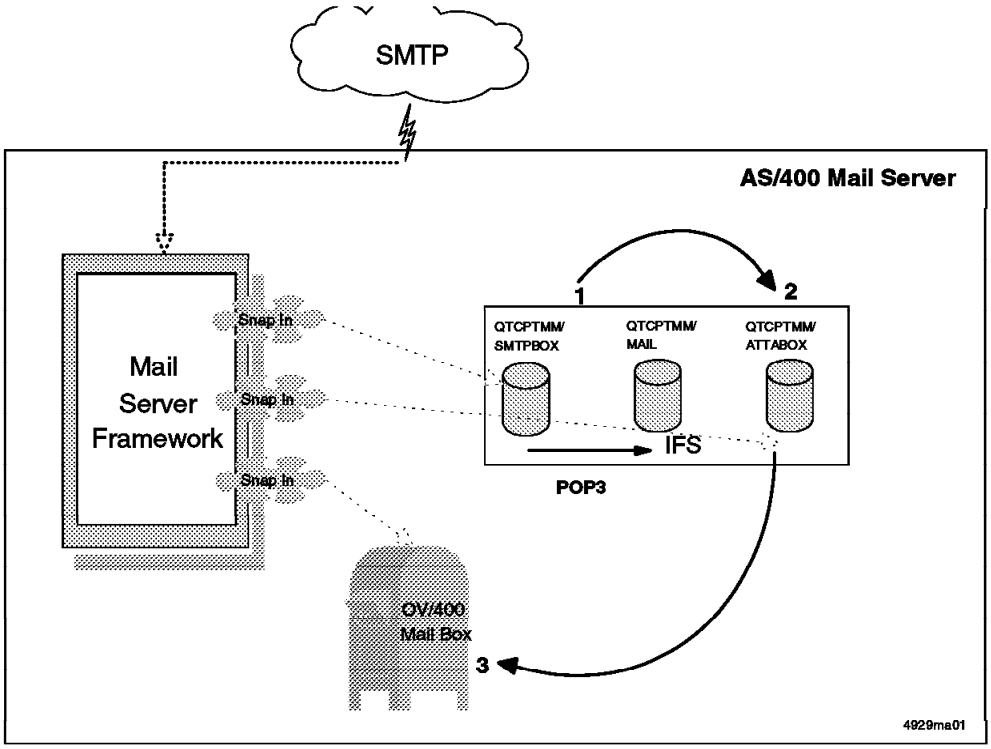


Figure 92. Flow of Mail Delivered through SMTP to OV/400 and POP3 Users

Understanding the flow depicted in Figure 92 helps you plan the process to follow to clean up after a mail bombing attack or an accident. In the most general case, the mail is successfully delivered to a valid OV/400 user mail box and does not get stuck in some stage of the flow, so we recommend that you follow the flow in reverse during the clean up process:

1. Determine if the problem is really caused by e-mail.
2. Find the ″offensive″ piece (or pieces) of mail.
3. Clean up the OV/400 distribution.
4. Check the intermediate stages.

5.4.1.1 Determining the Cause of the Problem

If you suspect that your AS/400 system has been attacked through e-mail, display the Distribution Log using the Display Distribution Services Log (DSPDSTLOG) command. You are looking for many entries from SMTP remote users or a few **large** ones. We recommend using automatic registration *NO. This enables you to see in the distribution log entries, the first eight characters of the originator′s user ID and address. This might be useful information to identify the source of the attack even when it can easily be forged. Figure 93 on page 130 shows several entries from HACKER VNET that look suspicious to us.

```
                    Display Distribution Services Log

Type options, press Enter.
  5=Display details


    Function  Entry  ------Logged------              ----Originator----
Opt   Type    Type   Date      Time      Job Name     User ID   Address
      *RTR    *NRM    4/02/97  12:51:11  QMSF         QNONDELI  SYSNAM
 5    *ARV    *NRM    4/02/97  13:22:39  QDIA         HACKER    VNET
      *RTR    *NRM    4/02/97  13:22:39  QMSF         HACKER    VNET
      *ARV    *NRM    4/02/97  13:22:40  QDIA         HACKER    VNET
```

Figure 93. DSPDSTLOG to Verify E-Mail Attack

5.4.1.2 Finding the Offensive Piece of Mail

At this point, the OV/400 recipient probably already has the mail in the incoming mail box. If the user looks into the incoming mail, the display in Figure 94 is probably shown.

```
                                Work with Mail

Working with mail for  . . . . . :   ADAN     SYSNAM


Type options, press Enter.
  2=Revise a copy   4=Delete     5=View     6=Print   8=Work with detail
  9=Print options   10=Forward   11=Reply   12=File remote   13=File loc
  14=Authority      15=Fill form

                 -------From-------
Opt  Status      User ID   Address   Description                   Date
     NEW         HACKER    VNET      This is a mail bomb           Received
     NEW         HACKER    VNET      (Nv6kint.bak)                 04/02/97
     NEW         HACKER    VNET      This is HUGE                  04/02/97
     NEW         HACKER    VNET      Another MIME attach           04/02/97
     NEW         HACKER    VNET      Mail Chain                    04/02/97
     NEW         HACKER    VNET      FreeLance presentation        04/02/97
```

Figure 94. Incoming Mail for OV/400 Recipient

But you want to avoid having the users delete the mail from this display and run the risk that they might even receive a virus besides the time it takes them to clean up. So, go back to the distribution log shown in Figure 93 and display details to find the larger pieces of mail that have arrived.

By selecting *5=Display details* in Figure 93, you can see the size of the mail object.

```
                    Display Distribution Services Log Entry

Function . . . . . :   Distribution arrived
Job  . . . . . :   060619/QSNADS/QDIA
Date/Time  . . . . . :    4/02/97  13:22:39


Originator:
  User ID/Address  . . . . . :   HACKER    VNET
  System name/Group  . . . . :   SYSNAM
Sequence number  . . . . . :   2841
Origin date/Time . . . . . :    4/02/97  13:21:44
Object size  . . . . . :   7370160
Destination agent  . . . . . :   OfficeVision®
Number of:
  Destinations . . . . . :   1
```

Figure 95. DSTLOG Details - Identify Large Objects and Take Note of Originator

5.4.1.3 Cleaning Up OV/400 Distribution

To be able to clean up mail in the OV/400 user's mail box on behalf of the recipient, we recommend writing down the User ID/Address and Sequence number and use it as the *distribution identifier* in the Delete Distribution (DLTDST) command (see Figure 96).

```
                          Delete Distribution (DLTDST)

Type choices, press Enter.

Distribution identifier  . . . . > 'HACKER  VNET    2841'
              + for more values
Incoming or outgoing . . . . . .   *IN           *IN, *OUT, *ERR
user IDentifier:
  User ID  . . . . . . . . . . . > ADAN          Character value, *CURRE
  Address  . . . . . . . . . . . > SYSNAM        Character value
Distribution ID extension  . . .   *NONE         0-99, *NONE
              + for more values
```

Figure 96. Using DLTDST to Delete Incoming Mail for OV/400 Users

5.4.1.4 Checking the Intermediate Stages

As explained in Section 5.4.1, "Recovering from E-Mail Attack to OV/400 Users" on page 128, a piece of mail delivered through SMTP to OV/400 users is stored in the IFS during the flow for a short period of time. It is possible that the mail "gets stuck" before being delivered to the recipient because the ASP threshold is reached in the process. To assure complete clean up, you should check the point of entrance for SMTP-delivered mail /QTCPTMM/SMTPBOX. Figure 97 on page 132 shows a piece of mail in this stage.

```
                          Work with Object Links

Directory  . . . . :   /QTCPTMM/SMTPBOX

Type options, press Enter.
  3=Copy   4=Remove   5=Next level   7=Rename   8=Display attributes
  11=Change current directory ...

Opt   Object link            Type             Attribute    Text
      Q825495552.NOTE        STMF
```

Figure 97. Incoming Mail through SMTP - First Stage /QTCPTMM/SMTPBOX

You can use *8=Display attributes* to see the size of the entry, or you can use the DSPLNK command with output to *PRINT to see the size of every entry in /QTCPTMM/SMTPBOX.

The next stage you must check is /QTCPTMM/ATTABOX. MSF duplicates the Q825495552.NOTE entry found in /QTCPTMM/SMTPBOX, splits it up, and stores it in /QTCPTMM/ATTABOX as shown in Figure 98. Once again, you can use *8=Display attributes* to see the size of each entry.

```
                          Work with Object Links

Directory  . . . . :   /QTCPTMM/ATTABOX

Type options, press Enter.
  3=Copy   4=Remove   5=Next level   7=Rename   8=Display attributes
  11=Change current directory ...

Opt   Object link            Type             Attribute    Text
      JW201297.TEV           STMF
      JW201328.TXT           STMF
      JW204164.BIN           STMF
      JW214128.MNOTE         STMF
```

Figure 98. /QTCPTMM/ATTABOX/ Contains Mail Before Delivering to OV/400 User

The different object links:

- JW201297.TEV (envelope)
- JW201328.TXT (text attached)
- JW204164.BIN (binary)
- JW214128.MNOTE (mime note)

  Note: JW2xxxxx.yyy shown in Figure 98 are examples. They may have different names on your system.

If you are sure you have been attacked, you can remove the mail found in the IFS directories previously described.

5.4.2 Recovering from E-Mail Attack to POP Clients

The flow of a piece of mail that arrives through SMTP to be delivered to POP clients is simpler than the one described for OV/400 clients in Section 5.4.1, “Recovering from E-Mail Attack to OV/400 Users” on page 128:

1. When a piece of mail arrives at the AS/400 system through SMTP, it is stored in the IFS in /QTCPTMM/SMTPBOX.
2. If the recipient is a POP client, the piece of mail is handled by one of the snap-ins of the AnyMail Mail Sever Framework and linked to the POP mail

boxes for all of the recipients. The POP Mail boxes are in /QTCPTMM/MAIL/user (see Figure 92 on page 129). At this point, the piece of mail is unlinked from /QTCPTMM/SMTPBOX.

Unfortunately, there is no easy way to find out how large the mail delivered to the POP clients is or who the originator is. You must look at the POP clients mail boxes in the IFS one-by-one to find this information. Figure 99 shows a POP user mail box. By using option 8, Display attributes, you can find the size of the object.

```
                              Work with Object Links

Directory  . . . . :   /qtcptmm/MAIL/ADANTEST

Type options, press Enter.
  3=Copy   4=Remove   5=Next level   7=Rename   8=Display attributes
  11=Change current directory ...

Opt   Object link            Type             Attribute    Text
      JW121468.NOT           STMF
```

Figure 99. POP User Mail Box

You must remove the entries to get rid of unwanted mail.

5.4.3 Cleaning Up Unprocessed SMTP Distributions

If you stopped SMTP but there were still many unprocessed distributions from the mail bomb, you should clean them up before re-starting SMTP. SMTP cleans up the unprocessed distributions based on the existence and content of a user-defined data areas called QTMSCLEAN in QUSRSYS.

To produce a ″cold″ start to free all unprocessed mail, create the data area with an upper-case or lower-case ″c″ in it.

```
CRTDTAARA DTAARA(QUSRSYS/QTMSCLEAN) TYPE(*CHAR) LEN(1) VALUE(C)

STRTCPSVR *SMTP
```

Now you can delete the QTMSCLEAN data area. Refer to Appendix E in the *TCP/IP Configuration and Reference* manual for more information on *cleaning up unprocessed SMTP distributions*.

5.4.4 Mail Delivered to Invalid Users

The attacker might not know valid users in your system but the attacker can still attack specifying invalid recipients.

The incoming mail that arrives through SMTP is stored in the IFS directory \QTCPTMM\SMTPBOX for a short time. When the system determined that the destination is an invalid user, the object in the IFS is removed (unlinked). If the object is large enough, it may cause the ASP threshold to be exceeded and a message is sent to the QSYSOPR message queue. When the object is unlinked, the amount of disk used goes below the ASP threshold value. The next piece of mail bomb causes the same action. The QSYSOPR message queue may be filled with messages about ASP threshold being exceeded.

If the mail object temporarily stored in the IFS is not large enough to cause the ASP threshold to be exceeded, the CPU may be overloaded.

5.5 Connecting your AS/400 Mail Server to the Internet - Scenarios

The purpose of this section is to provide some examples of connecting your AS/400 mail server to the Internet. Even though (as explained in Chapter 1, “Internet Security Overview” on page 1) the purpose of this redbook is to discuss AS/400 Internet application security and **not** network security, we include some connectivity security considerations in this section.

5.5.1 Connecting Your Production AS/400 System to Internet through ISP

In this scenario, we assume that you are currently in a traditional SNA network and your users are OV/400 clients. You can connect your current network to the Internet through an Internet Service Provider using your AS/400 system as an SMTP-to-SNA gateway.

May want separate LAN segments to make the transition to TCP/IP easier in the internal network. Otherwise if you have some users trying out TCP/IP on their PCs they could be exposed to internet traffic.

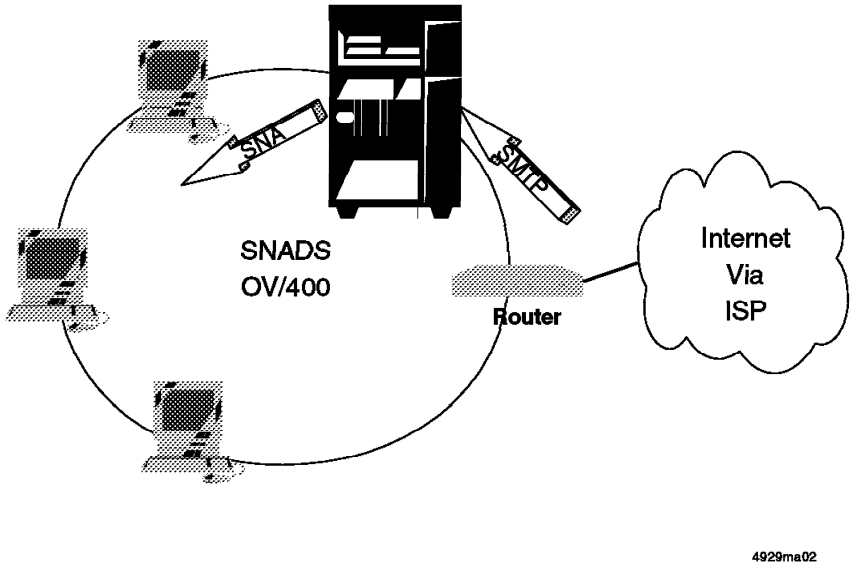


Figure 100. Production AS/400 Mail Server Directly Connected to Internet through Internet Service Provider

- Benefits

  This is a non-disruptive, simple, and fairly inexpensive way to give e-mail access to your users through the Internet. It does not require any changes to your current network and you only need to add a router to connect to the ISP. You need to add TCP/IP configuration and configure Internet users in the system distribution directory. All the software you need is in OS/400 and TCP/IP Connectivity Utilities/400 at no additional cost.

- Exposures

Because your are connecting your production system **directly** to the Internet, it might become an easy target for attacks.

- Recommendations

  Besides implementing the general security recommendations discussed in Section 5.2.1, “Security Solutions” on page 123, you should consider:

  – Using the packet filtering capabilities available today in most modern routers to prevent unwanted services or unwanted IP addresses from coming in.

  – Making sure you **only** start SMTP and no other TCP/IP service. You are still vulnerable to ICMP attacks since you can′t turn it off (for example, ping, trcroute, redirect...)

5.5.2 Connecting Your Production AS/400 System to Internet using Firewall

This scenario is basically the same one as the one previously discussed but here we are adding a firewall between the Internet and your production AS/400 system where the mail server is running. We strongly recommend that you consider adding a firewall for extra protection when you attach a production system to the Internet.

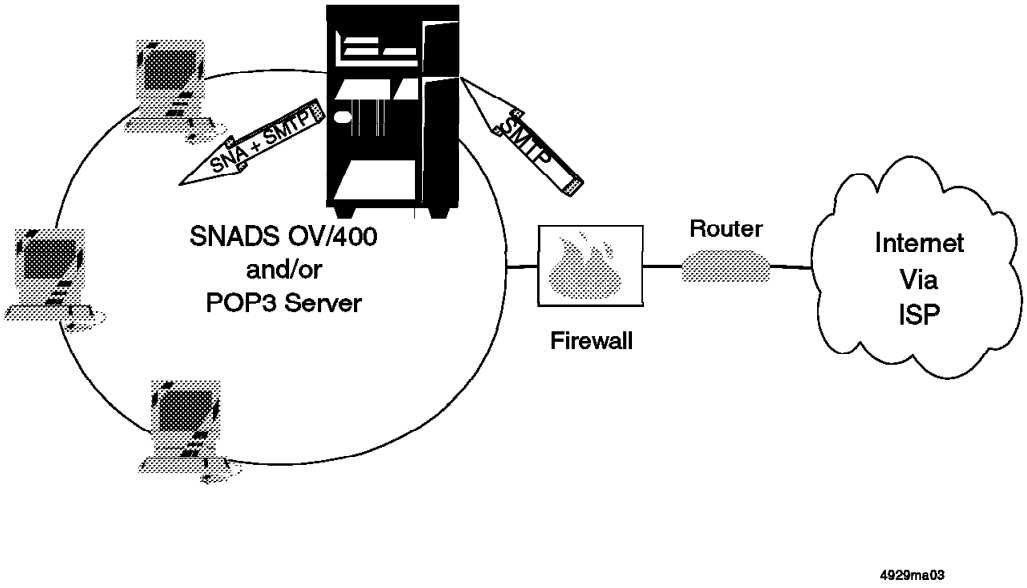


Figure 101. Production AS/400 Mail Server Connected to Internet using a Firewall

As shown in Figure 101, we have added POP3 server capabilities to the AS/400 system; the clients in this scenario are a mix of OV/400 users and POP3 clients. The AS/400 system supports exchanging mail between POP3 and OV/400 clients.

- Benefits

  The firewall adds extra protection and also provides an external mail server that behaves the same as a proxy so that the secure mail server running on your AS/400 system is not directly exposed to the Internet.

  The AS/400 system acts as both an SMTP-to-SNA gateway to transfer mail to and from the Internet and OV/400 users and as a POP3 server.

- Exposures

  The presence of the firewall should not fool you:

  – The firewall does not provide confidentiality; the mail flowing through the firewall can still be sniffed once in the public network.
  – POP3 user IDs and passwords can be sniffed if POP clients are used from the Internet.

- Recommendations

  The same recommendations discussed in this chapter and in Section 5.5.1, “Connecting Your Production AS/400 System to Internet through ISP” on page 134 still apply. To connect your AS/400 system to the Internet through a firewall should in no way relax your security policies and procedures. It is important to keep the layer approach to security.

5.5.3 Connecting Your Production AS/400 System to Internet through IGN

Using the IBM Global Network to connect to the Internet, you get both an ISP and an SMTP-to-SNA gateway. If you have an SNA network with OV/400 users, this is probably a good solution to provide e-mail access through the Internet.

IGN isolates your production AS/400 system from the Internet and because you do not need to start any TCP/IP service (not even SMTP), it is unlikely that a hacker from the Internet can break into your production AS/400 system. Figure 102 depicts this scenario.

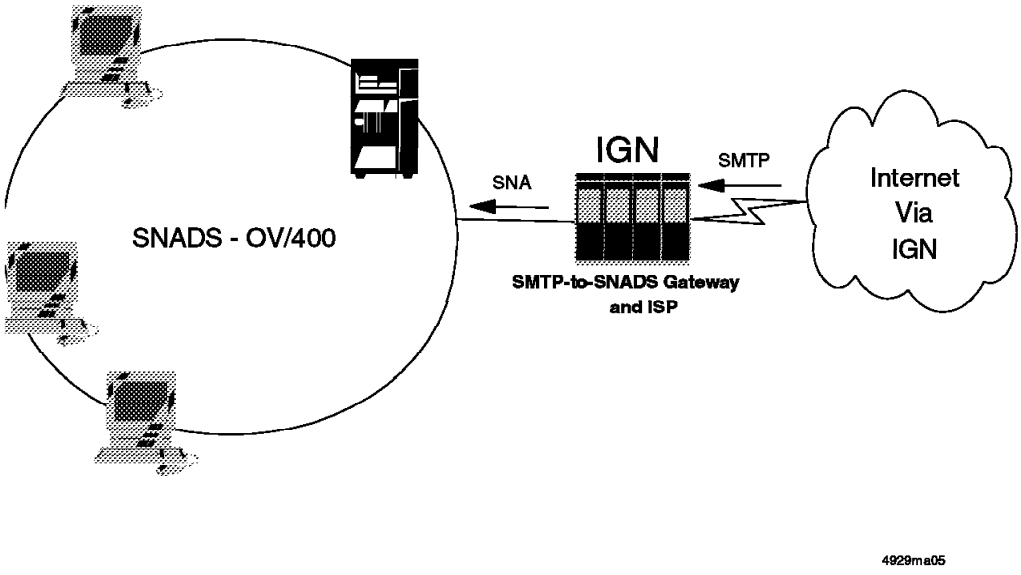


Figure 102. Using IGN as ISP and SMTP-to-SNA Gateway

5.6 Summary

To summarize:

- Set the auxiliary storage threshold.
- Do not use *ANY *ANY entries in the system distribution directory.
- Disable automatic registration of users.
- Educate your users to always check for viruses when receiving MIME attachements.
- Lower the priority of mail-related jobs.
- Have an action plan in case of mail bombing attack.

Chapter 6. FTP Security

The FTP consists of two parts: the client and the server function. The user (or client in this case) initiates FTP subcommands that are sent to the FTP server. The result of these subcommands is either that files are transferred, commands are run on the server, or requested information is displayed. Some of these subcommands are powerful and you **must** control them using exit programs.

The AS/400 system implements both the FTP client and server function. The AS/400 FTP client and server can use many of the different file systems in the *Integrated File System* (IFS). This basically opens up the entire system to FTP except non-file type objects in the QSYS.LIB file system such as programs or user profiles. Such objects may, however, be copied to a file or save file and then transferred.

Although security of your system is both a concern of the client and server functions of FTP on the AS/400 system, this chapter concentrates on only the server side of FTP. That is, we explore the configuration and programming necessary to secure the AS/400 system as an FTP server on the Internet (or intranet), not as an FTP client.

If a client has *USE authority to an object (for example, a file), the client can copy it. Since you have no control over the security implementation on a client, you must secure the information on your AS/400 system. Clients cannot be trusted. FTP security is defined by careful configuration and you must use FTP exit programs for sign-on and subcommand validation.

So, in the situation where we find the AS/400 system as an FTP server, we should understand that many different types of FTP clients are available to the end user. Generally, these FTP clients can be broken down into two implementations:

1. Text-based FTP client that allows an FTP subcommand entry through the command line.
2. Graphical user interface to hide some of the complexity of the underlying FTP subcommands.

FTP client capability is added to most Web browsers also. The Web browsers allow anonymous as well as user ID and password-based access to your AS/400 FTP server.

From a security point of view, it is transparent to the AS/400 system how the user submits their FTP subcommands. The AS/400 system (in all of the preceding cases) receives the same text based FTP subcommands and acts accordingly.

6.1 AS/400 FTP Server

FTP is a good representative of the business side of the TCP/IP applications. FTP offers the Internet user an easy way to access a large amount of data that is located on FTP servers all over the world. Any kind of data (ASCII text files, programs, and multimedia information) can be retrieved from FTP servers.

6.1.1 What Can the AS/400 FTP Server Do?

The AS/400 FTP server allows file exchange with the following file systems: Root (/), QSYS.LIB, QDLS, QOPT, QOpenSys, QLANSrv, and QFileSrv.400.

The QSYS.LIB file system allows an FTP client to directly access the AS/400 database. This means you can access physical files (PF), logical files (LF), source physical files (SRCPF), and save files (SAVF).

There are many arguments that differentiate the AS/400 server implementation from other servers on the market:

- Support of several different file systems on one system (IFS).
- Built in high security standards.
- AS/400 programs and CL commands can be called through the special AS/400 FTP server subcommand RCMD (Remote Command).
- Transfer of folders and documents in the QDLS file system.
- Sending or receiving physical files, source files, logical files, and save files.
- If both the server and client are AS/400 systems, you can distribute most AS/400 object types as SAVFs.
- Creating and deleting libraries, files, and members using the AS/400 FTP server subcommands.
- High availability of the FTP server through mirroring, RAID-5, and other AS/400 system availability options.

Here are some scenarios where the AS/400 system is best used as an FTP server on the Internet and your own intranet:

- A software company offers beta code versions of its new products to be retrieved from the FTP server.
- Enhancements and corrections of the current software products are available from the FTP server or even entire new software versions.
- Retailers can retrieve product information, availability, and price lists to be included in their applications.

6.1.2 How Secure is Your FTP Server?

Sending and retrieving files are the basic functions supported by the AS/400 FTP server. However, before the FTP client may submit FTP subcommands, the client usually provides a user ID and password to be identified and authorized. This user ID relates to an AS/400 user profile, and any action the FTP client requests is accepted or rejected based upon OS/400 given authorities. If, for example, remote FTP users have *USE access to a particular OS/400 object, they cannot change or delete that object, no matter which FTP subcommand they attempt to use.

The AS/400 FTP server allows much more than simple file transfer:

- Creating or deleting libraries, folders, or directories
- Changing current libraries, folders, or directories
- Listing files
- Deleting files
- Sending and retrieving files
- Renaming files

- Executing AS/400 CL commands through the FTP RCMD subcommand. This is a useful subcommand as it allows you to pre-create files in an intranet solution. The RCMD subcommand has the effect of giving a command line to the remote user.

In spite of access control with user ID and password, you may want to implement further measures to secure your FTP server.

In Section 6.2, “Benefits and Potential Exposures,” we discuss the benefits versus the potential exposures to your AS/400 system.

Immediately following that section, we start to explain the layers used to secure FTP on the AS/400 system. Layers such as the network, application security, and ultimately OS/400 security are addressed in Section 6.4, “How to Secure Your FTP Server” on page 145.

6.2 Benefits and Potential Exposures

The AS/400 FTP server offers a strong set of functions not only to transfer files, but also to perform data management functions through a TCP/IP network. It is a potential exposure for your company if you allow this full range of FTP functions when attaching your AS/400 system to the Internet (or even an intranet).

To investigate this issue, we need to review the various functions the FTP server offers to the FTP clients and we need to assess the potential risks.

Table 12. FTP - Benefits and Exposures

FTP Client Function	Benefits to Client	Exposures on Server
Logon	Access control	Unauthorized Access, sniffing
Download from AS/400 system	Serving any kind of data	Privacy
Upload to AS/400 system	Easy way to transfer data	Junk bombing, denial of service, altered data
Change Directory	Work with multiple directories	Unauthorized access
Create/Delete Directory, File Deletion and Rename	Data management functions	Altered/lost data
Execution of CL Commands	Integration of FTP into application	Altered/lost data, open new threats

Important Note

For the remainder of our discussions, please remember that there is **no** support in FTP for encrypting/decrypting passwords. The only way that the data can be encrypted is if FTP uses AnyNet® running over an LU6.2 session that is running LU6.2 session level encryption.

6.2.1 User Access to FTP Server

Benefit: The logon function allows the server to identify and authorize the FTP client. If an FTP client can be securely identified, you can allow individual access capabilities based upon OS/400 user profiles. As previously discussed, there is no way to **securely** identify a client since user IDs and passwords are sent in the clear and may be sniffed by a hacker.

Exposure: In an intranet solution, this user ID and password authorization works well. To some degree, you trust your employees.

Since the Internet audience is unspecified, user identification is difficult to accomplish. There are just too many users that might possibly want to access your information. You need to allow unidentified access (which is often called *anonymous FTP*) to allow you to serve strictly public data only.

TIP

We do not recommend sending user IDs and passwords across the Internet.

If you feel you must provide different kinds of data to different users based upon a user ID and password access control, try to limit it to intranet users.

Do not allow a user profile with *ALLOBJ (such as QSECOFR) or *IOSYSCFG to access your system across an uncontrolled network such as the Internet.

6.2.2 Download from the AS/400 System

Benefit: This is the primary benefit and objective of an Internet FTP server (to provide access to information that is located on your AS/400 server and allow the client to copy this information).

Exposure: When you decide to give a certain group of users read access to information on your system, you must be extremely careful to only authorize what you intend them to use. If you don't configure the security of other objects very tightly, there is a risk that they can either inadvertently or maliciously gain access to them. You must secure the information on your AS/400 system using resource security.

If you don't use anonymous FTP and you require that your users enter user ID and password, there is a chance that those will be sniffed.

6.2.3 Upload to the AS/400 System

Benefit: FTP offers an easy way to set up and perform file transfer to any accessible FTP server in the Internet. Mobile employees, business partners, and customers are able to send you data whenever they are ready to do so.

Exposure: Now that you are giving write access to your AS/400 DASD, the risk increases proportionately.

Someone may upload *junk* (useless information to your business, often of illegal nature) to your AS/400 system and advertise the location of your AS/400 system to all of their friends. Unless your DASD is filled, this is not denial of service, but it does reduce your service as your system may be somewhat busy handling traffic that is not at all part of your business. Thus, *junk bombing* is an insidious

attack that reinforces the idea that you should be actively logging and auditing your AS/400 server to stop this kind of attack as soon as it starts.

Your AS/400 system may also be used as a server to transfer the *junk* to other systems. If this transfer causes damages of any kind to the receiving system, the IP address of your AS/400 system identifies the source of the problems.

There is also the risk of compromising data integrity: data might be downloaded, altered and uploading again.

Tip

You can take the following meassures to deter this type of attack on your system:

- Analyze the files with *PUBLIC(*CHANGE) authority and change to *PUBLIC(*EXCLUDE) or *PUBLIC(*USE) when appropriate.
- Direct all of the files that are uploaded to your system to be placed into a section of the IFS that is write-only. That is, the remote user can write to it but cannot read from it. This action defeats the purpose of using your AS/400 system as an FTP server for junk files. You can control this through the FTP exit programs as we see later.
- Control the upload of files to your AS/400 FTP server by using Anonymous FTP with an FTP exit program. Anonymous FTP is discussed later in this chapter. See the *TCP/IP Configuration and Reference* for additional information about Anonymous FTP.
- Clear the directory periodically with a program.

Worse, the client could simply send large numbers of small (or large) files to your system with the purpose of consuming CPU and DASD to the point that your system cannot handle the normal business traffic. This is defined as denial of service.

TIP

When a client sends you a file using FTP, the content of an existing file is either replaced or a new file is created. If a new file is created, it is owned by the user profile the client used when logging on to your AS/400 system.

You can limit the amount of DASD a client can occupy to store owned objects by changing the MAXSTG parameter in the user profile used by the client.

6.2.4 Directory and File Manipulation

Benefit: Changing the current directory, creating and deleting directories, deleting and renaming files are management kinds of functions that are a useful part of the FTP subcommands. This allows you to pre-create files and members on the AS/400 system to ensure the best performance and translation of the data.

Exposure: The risks, however, are great. Even in an intranet environment, these FTP subcommands are the same as giving the user the power to CRTLIB, DLTLIB, CRTSRCPF, DLTF, and so on. In the Internet environment, this opens up exposures such as denial of service and disclosure of sensitive data.

TIP

If you need to allow some or all these FTP subcommands, you must control their use through the FTP exit programs. Our tip is to define an environment where the end user does not have to use these data and file management commands.

6.2.5 TCP/IP Subcommand RCMD

Benefit: The Remote CoMmanD is powerful.

Exposure: This subcommand should not be allowed in an Internet environment. It is even more dangerous that directory and file manipulation

TIP

Opening up for the RCMD FTP subcommand is the same as giving a client a command line.

You must use an FTP exit program to eliminate the ability of the remote user to use the RCMD (please see Section 6.4.2, “FTP Exit Programs” on page 147).

6.3 FTP and Internet, General Guidelines

Considering the threats from the Internet (and the intranet), your AS/400 FTP server must be carefully designed.

Because you cannot administer user IDs and passwords for each of the potential Internet users and still offer an easy access to the data you want to provide, a simple way to allow access to the public data on an Internet server is to use the *Anonymous FTP*. Anonymous FTP requires two exit programs. The first controls the logon; the second controls the functions allowed.

Most FTP implementations in the Internet offer this support. Normally, when you log on to an FTP server, you are asked for a user ID and password. As common user ID, you enter *anonymous*. Custom and Internet etiquette dictates that you use your e-mail address (for example, myname@ibm.com) as the password. You may verify the e-mail syntax in your exit programs.

After signing on as an anonymous user, the FTP server offers a limited range of information that is considered public accessible and can be read by everyone who wants to. You can control this in your exit program.

So the general guidelines for an FTP server on the AS/400 system fall into these categories:

Network	FTP Server Characteristics
Intranet	Allows anonymous or strictly controlled access, depending on the potential users. A mixture of both is possible as the FTP exit points are flexible in this area.
Internet	Allows anonymous access only.

TIP

One of the pieces of information that is presented to your FTP exit program is the IP address of the client. This IP address cannot be trusted. The IP address of the client can be *spoofed*. Spoofing is when someone on the untrusted Internet pretends to be a different IP address (for example, one you believe you can trust).

6.4 How to Secure Your FTP Server

What are the possible measures to secure your FTP server? There are four:

- **System Security:** Correct system values and appropriate user profile and password management is essential to implementing a secure FTP server environment.

  Note: Invalid sign-on attempts to FTP do not result in any device being varied-off For more information on how to establish your OS/400 configuration as the main line of defense to any form of Internet attack, please see Chapter 2, “Start Here by Securing OS/400®” on page 37.

- **Application Security:** Only start the FTP server when it is needed. If you do not need it during off hours or on weekends, stop it. The FTP exit programs are the means to control and log the FTP requests. This is the focus of the remainder of this chapter.

- **Network Security:** Depending on the services you want to offer to the Internet, filtering multi-protocol routers and firewalls may be required to secure your Internet server and your corporate internal network. Please see 1.7, “Internet Firewalls” on page 13 for more information.

- **Transaction Security:** Keep in mind that currently data being transferred using FTP is not encrypted and privacy cannot be guaranteed.

  An option is to create a program that reads the file to be transferred, encrypts fields in the records using APIs, and creates an output file that is sent using FTP. The receiver must have a similar program that reads the encrypted file, decrypts it using APIs, and creates a new, unencrypted output file.

Tip

To encrypt/decrypt data, you can choose between Cryptographic Support/400 (see *AS/400 Crytographic Support/400 User*), Common Cryptographic Architecture Services/400 (see *Common Cryptographic Architecture Services/400 Installation and Operator*), a third party product, or your own.

For Cryptographic Support/400, you can use the DEA to conceal data through the Cipher Data (CPHDTA) command.

For Common Cryptographic Architecture Services/400, use CSNBDEC (Decipher) and CSNBENC (Encipher) in library QTSS.

Common Cryptographic Architecture Services/400 requires the IBM 2620 or 2628 cryptographic processor

You can run FTP over SNA using ANYNET and take advantage of LU6.2 Session Level Encryption. LU6.2 SLE requires Common Cryptographic Architecture Services/400 and the IBM 2620 or 2628 cryptographic processor.

In the next sections, we look at how you build up the classical anonymous FTP service in the Internet.

6.4.1 FTP Server Attributes

The TCP/IP Connectivity Utilities for AS/400 licensed program comes with TCP/IP servers configured. The Change FTP Attributes (CHGFTPA) command (please see Figure 103) allows you to change the FTP server attributes.

```
                        Change FTP Attributes (CHGFTPA)

Type choices, press Enter.

Autostart servers  . . . . . .   *YES          *YES, *NO,
Number of initial servers  . . .  3            1-20, *SAME, *DFT
Inactivity timeout . . . . . .   300          0-2147483647
Coded character set identifier    00819        1-65533, *SAME, *DFT
Server mapping tables:
  Outgoing EBCDIC/ASCII table  .  *CCSID       Name, *SAME, *CCSID, *D
    Library  . . . . . . . . . .                Name, *LIBL, *CURLIB

  Incoming ASCII/EBCDIC table  .  *CCSID       Name, *SAME, *CCSID, *D
    Library  . . . . . . . . . .                Name, *LIBL, *CURLIB


F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this disp
F24=More keys
```

Figure 103. Change FTP Attributes

Only two of the FTP server attributes have an effect on security: Autostart servers and the Inactivity timeout.

TIP

The Autostart servers parameter default value is *YES. This means whenever the network administrator issues the STRTCP command to test, for example, just the network, the FTP server is automatically started. If the system and the FTP server application is not yet secure, this opens up your system to attack at a time when you are least expecting.

On the Internet, this can be destructive. Hackers and crackers know that the best kind of system to attack is one that has just been attached to the Internet. They monitor publicly published lists of new Internet domains and host names looking for the opportunity to crack a new system before all of its security systems are in place.

You can change the Autostart servers to *NO by specifying

```
CHGFTPA AUTOSTART(*NO)
```

After you have secured your FTP server, you can individually start it with the following command:

```
STRTCPSVR SERVER(*FTP)
```

TIP

The Inactivity timeout parameter specifies the number of seconds the system allows an FTP connection to remain inactive before it is ended. This value is set to the default of 300 seconds. To minimize the risk of spoofing, you might want to reduce this value.

6.4.2 FTP Exit Programs

An *exit point* is a specific point in the TCP/IP application where control may be passed to a user written exit program The FTP exit programs control the use of the FTP server and FTP client. These exit programs offer additional security and transaction logging capabilities on an AS/400 FTP server.

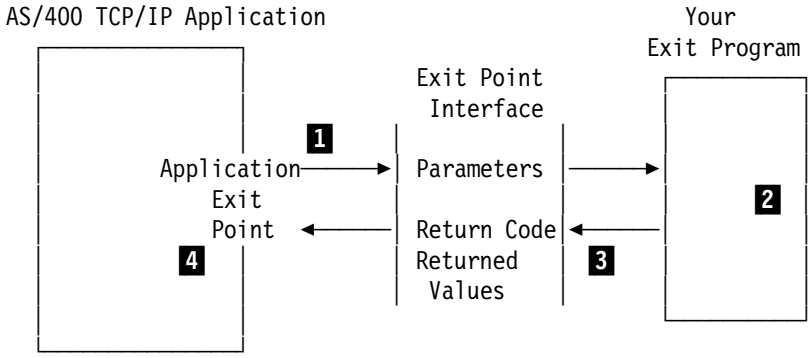


Figure 104. TCP/IP Exit Point Processing

Processing flow:

1 TCP/IP application passes the client's request and its parameters to the exit program.

2 Exit program processes the request and its parameters.
3 Exit program returns information to the TCP/IP application.
4 TCP/IP application performs operation based on the exit program response.

Three different exit points are provided for FTP. The first one is used to validate requests from the FTP client, the second one is used to validate requests from the FTP server, and the third one controls the logon requests to the FTP server.

Table 13. FTP Exit Points

FTP Application	Exit Point	Exit Point Format
FTP Client	QIBM_QTMF_CLIENT_REQ	VLRQ0100 1
FTP Server	QIBM_QTMF_SERVER_REQ	VLRQ0100 1
FTP Server	QIBM_QTMF_SVR_LOGON	TCPQ0100

1 The same format is used for both the FTP Client and FTP server for request validation. This enables the use of one program for both client and server request validation.

Note: To enable an anonymous FTP, you must define exit programs for *both* FTP server exit points.

The FTP server logon exit program permits or denies a logon to an FTP server based on one or more of the following values:

- User ID
- Password
- Remote IP address

FTP request validation exit programs (FTP Client or FTP Server) permit or deny a specific FTP operation based on one or more of the following conditions:

- User profile
- Remote IP address
- Directory, library, files (path names)
- CL commands

Your exit programs must be defined for their exit points using the OS/400 registration facility. You can use the Work with Registration Information (WRKREGINF) command to display a list of exit points or simply use the Add Exit Program (ADDEXITPGM) command.

```
                        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_QRQ_SQL           RSQL0100     *YES     Original Remote SQL Server
 _   QIBM_QSY_CHG_PROFILE   CHGP0100     *YES     Change User Profile
 _   QIBM_QSY_CRT_PROFILE   CRTP0100     *YES     Create User Profile
 _   QIBM_QSY_DLT_PROFILE   DLTP0100     *YES     Delete User Profile - after d
 _   QIBM_QSY_DLT_PROFILE   DLTP0200     *YES     Delete User Profile - before
 _   QIBM_QSY_RST_PROFILE   RSTP0100     *YES     Restore User Profile
 _   QIBM_QTF_TRANSFER      TRAN0100     *YES     Original File Transfer Functi
 _   QIBM_QTMF_CLIENT_REQ   VLRQ0100     *YES     FTP Client Request Validation
 _   QIBM_QTMF_SERVER_REQ   VLRQ0100     *YES     FTP Server Request Validation
 _   QIBM_QTMF_SVR_LOGON    TCPL0100     *YES     FTP Server Logon
 _   QIBM_QTMT_WSG          QAPP0100     *YES     WSG Server Sign-On Validation
                                                                        More...
Command
===>
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Figure 105. Work with Registration Information

6.4.2.1 FTP Server Logon Exit Program

The FTP server logon exit program allows you to enforce your own rules for handling logons to an FTP server. The exit program receives as parameters:

- Application identifier
- User identifier
- Authentication string (password or e-mail address from FTP logon)
- Remote IP address

The exit program returns a parameter whether the logon is rejected, accepted, or if the logon operation should be continued. When the logon is accepted, the user profile is returned, and a new current library can be returned as a parameter to override the one specified in the user profile. When the logon is continued, the program can return a new user profile and a new current library as parameters.

Overriding the initial current library of a user (set in the user profile) is called direct routing. The direct routing support allows you to route the client directly to a specific library immediately following the logon to the AS/400 FTP server. This allows you, for example, to route sales representatives to their own library. In such a private library, you can keep the files that are unique for a specific sales representative.

The exit program can control *the access to the server* based on the requester's address or user ID. The exit program can also be used to force your anonymous FTP user to give a valid-looking e-mail address as a password (one that contains an *"@"* character). The exit program cannot give any informative messages to the clients so they see only that a logon is either accepted or rejected.

Specifying an FTP server logon exit program is one-half of enabling the anonymous FTP. You should create a separate user profile for this. We strongly

suggest that this user profile has a password of *NONE. All server logon requests and all anonymous server logon requests can and *should* be logged to see who accesses your AS/400 system. Because the exit program receives a valid user password as a parameter, care should be taken with what information to log. The importance of logging is great when you are connected to the Internet. You might be compelled later on to maintain a ″black list″ of IP addresses whose owners have tried to do something harmful and block their entrance to your system. These users must spoof an IP address to gain access to your AS/400 system again. Although most of the general Internet users can be considered to be quite harmless, there are the fringe groups that just might have something against your area of business.

6.4.2.2 FTP Client and Server Request Validation Exit Programs

Both client and server request validation exit programs use the same exit point format (parameters).

The FTP request validation exit program gives you control over whether an operation (that is, an FTP subcommand) is performed or not. The decisions made by the exit programs are in addition to any validation performed by the application program.

The exit program receives the following parameters:

- Application identifier (server or client request)
- Operation identifier
- User profile
- Remote IP address
- Some operation specific data (path name, CL command, and so on)
- Length fields for some of the previously mentioned parameters

The operation returns as a parameter whether it is rejected or accepted. An operation can be rejected completely for the remainder of the session, rejected this time, allowed this time, or allowed unconditionally for the remainder of the session.

Your user exit program can be based upon the user′s IP address and provide different access to AS/400 data. For example, your employees may have unlimited access, but your customers are allowed to see (and ″get″) only the product information that is available to the public. You can also limit FTP users from using certain CL commands but allow other commands to be used.

The other half of enabling the anonymous FTP is specifying an FTP server exit program. You should create a good protection scheme against your FTP clients by using OS/400 resource security. The need to upload files must be considered carefully. If you allow it, limit uploading to selected libraries or directories that are not the same as download directories.

Limit the use of CL commands. Block everything that you do not explicitly allow. Open up when it is needed. You might also want to use symbolic links in your ″public″ directory and have the files actually reside in a more secure directory or folder.

For more information, refer to the *Cool Title About the AS/400 and Internet* and *TCP/IP Configuration and Reference* manuals.

6.5 Your FTP Server in the Internet - Scenarios

To establish an anonymous FTP server in an orderly manner, perform the following steps:

1. Define the policy of your FTP server site.
2. Create a user profile for the FTP users.
3. Prepare the server logon exit program.
4. Prepare the request validation exit program.
5. Add exit programs to the exit points.
6. Test your FTP exit programs and further security measures.

To help you define the policy for your AS/400 FTP server, we want to introduce the chart in Figure 106.

To read this chart, draw vertical lines through it at different points. For example, if your AS/400 system is a server on the Internet, we recommend that you only provide anonymous access to your AS/400 system. An intranet server can restrict access based upon user ID and password, but the user ID and password flow in the clear.

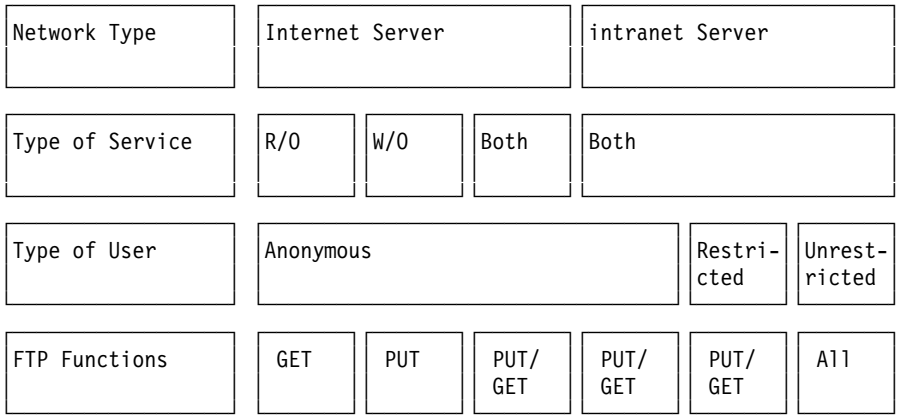


Figure 106. FTP Server Type and Access

Providing general information is the usual application in an Internet environment. Hence, we allow the anonymous user to access a strictly defined set of information in a read-only mode.

In the next sections, we explore implementations of FTP servers on the AS/400 system.

6.6 Anonymous FTP Support, Read-Only

The information made available to an anonymous FTP server is considered to be publicly accessible. The anonymous user is not allowed to transfer files to the FTP server system.

Typically, the only operations allowed are:

- Logging on using FTP.
- Listing the contents of a limited set of directories or libraries.
- Retrieving files from these directories or libraries.

TIP

All information that is not to be accessed from the Internet must have *PUBLIC(*EXCLUDE), or the anonymous user profile must be specifically excluded.

6.6.1 Define Anonymous FTP Server Site Policy

The following rules are for our read-only anonymous FTP server:

- Read-only server
- No authentication, no user IDs, anonymous access
- Served information is publicly available.
- All access to our AS/400 FTP server is logged.
- The logs are reviewed for possible attacks daily or weekly.
- The server is tested for security holes once a month.

6.6.2 User Profile for Anonymous User

Enter the following CL command to create the user profile for the anonymous FTP user:

```
CRTUSRPRF USRPRF(ANONYMOUS) PASSWORD(*NONE) INLMNU(*SIGNOFF) LMTCPB(*YES)+
SPCAUT(*NONE) HOMEDIR('/public') AUT(*EXCLUDE) TEXT('anonymous FTP user')
```

6.6.3 Prepare Anonymous / Public Directory

We can create a new directory on our AS/400 server that is used to store the files that we allow our ANONYMOUS user profile read-only access to.

Note: The user profile that you use to create this directory is the owner of the directory. The should not be a group profile.

```
MKDIR DIR('/public') DTAAUT(*EXCLUDE) OBJAUT(*NONE)
```

Next, we need to give the user profile ANONYMOUS read-only access to all objects in the /public directory:

```
CHGAUT OBJ('/public') USER(ANONYMOUS) DTAAUT(*R) OBJAUT(*NONE)
```

Use the Work with Authority (WRKAUT) command to verify:

```
WRKAUT OBJ('/public')
```

```
                          Work with Authority

Object . . . . . . . . . . . . :   /public
Owner  . . . . . . . . . . . . :   BRSMITH1
Primary group  . . . . . . . . :   *NONE
Authorization list . . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                    Data     --Object Authorities--
Opt  User        Authority  Exist  Mgt  Alter  Ref

     *PUBLIC     *EXCLUDE
     BRSMITH1    *RWX         X     X     X     X
     ANONYMOUS   *R
```

Figure 107. Work with Authority Display that Shows ANONYMOUS with Read-Only

6.6.4 Prepare Server Logon Exit Program

Table 14 shows the parameters in the FTP logon exit program. See the *AS/400 System API Reference* for more details. The values shown here may not apply to the version of OS/400 used on your AS/400 system.

Table 14 (Page 1 of 2). Parameters in Server Logon Exit Program

Position	Parameter	Input Output	Type(Length)	Read Only Anonymous Actions
1	Application identifier	Input	Binary(4)	This is always 1 = FTP server.
2	User identifier	Input	Char(*)	Must be ANONYMOUS, in any case.
3	Length of user identifier	Input	Binary(4)	Must be 9, the length of ANONYMOUS.
4	Authentication string	Input	Char(*)	Some anonymous FTP server implementations check for the commercial at sign (@) to verify that the remote user has typed in a valid e-mail address. This does not prove anything. You can demand a valid e-mail address and log the same, providing additional information that can prove useful later. This sample FTP server logon exit point program does not.
5	Length of authentication string	Input	Binary(4)	
6	Client IP address	Input	Char(*)	You cannot rely on the client IP address to be valid unless encryption is used.
7	Length of client IP address	Input	Binary(4)	

Table 14 (Page 2 of 2). Parameters in Server Logon Exit Program

Position	Parameter	Input Output	Type(Length)	Read Only Anonymous Actions
8	Return code	Output	Binary(4)	Logic: IF (User_identifier EQUAL 'ANONYMOUS') AND (Length_of_user_identifier EQUAL 9) THEN Return_code is 6 ELSE Return_code is 0 **Notes:** 1. Return_code of 0 rejects the logon operation. 2. Return_code of 6 accepts the logon operation. OS/400 overrides the user profile and initial current library with those that are returned in the output parameters of your exit program. The output parameter is ignored.
9	User profile	Output	Char(10)	Always: 'ANONYMOUS'
10	Password	Output	Char(10)	Ignored. For both Return codes 0 and 6, there is no need to specify the password.
11	Initial current library	Output	Char(10)	Always: '/public'

6.6.5 Prepare Request Validation Program

Table 15 shows the parameters in the FTP request validation exit program. You must see the *AS/400 System API Reference* if you want to create an exit program. The values shown here may not apply to the version of OS/400 used on your AS/400 system.

Table 15 (Page 1 of 2). Parameters in Server Request Validation Exit Program

Position	Parameter	Input Output	Type(Length)	Read Only Anonymous Actions
1	Application identifier	Input	Binary(4)	This is always 1 = FTP server.
2	Operation identifier	Input	Binary(4)	Must be ANONYMOUS, in any case.
2	User identifier	Input	Char(*)	Must be ANONYMOUS, in any case.
3	Length of user identifier	Input	Binary(4)	Must be 9, the length of ANONYMOUS.
4	Authentication string	Input	Char(*)	Some anonymous FTP server implementations check for the commercial at sign ('@') to verify that the remote user has typed in a valid e-mail address. This does not prove anything. You can demand a valid e-mail address and log the same, providing additional information that can prove useful later. This sample FTP server logon exit point program does not.
5	Length of authentication string	Input	Binary(4)	
6	Client IP address	Input	Char(*)	You cannot relay on the client IP address to be valid unless encryption is used.
7	Length of client IP address	Input	Binary(4)	Do not check.

Table 15 (Page 2 of 2). Parameters in Server Request Validation Exit Program

Position	Parameter	Input Output	Type(Length)	Read Only Anonymous Actions
8	Return code	Output	Binary(4)	Logic: IF (User_identifier EQUAL 'ANONYMOUS') AND (Length_of_user_identifier EQUAL 9) THEN Return_code is 6 ELSE Return_code is 0 **Notes:** 1. Return_code of 0 rejects the logon operation. 2. Return_code of 6 accepts the logon operation. OS/400 overrides the user profile and initial current library with those that are returned in the output parameters of your exit program. The output parameter is ignored.
9	User profile	Output	Char(10)	Always: 'ANONYMOUS'
10	Password	Output	Char(10)	Ignored. For both Return codes 0 and 6, there is no need to specify the password.
11	Initial current library	Output	Char(10)	Always: '/public'

6.6.6 Add Exit Programs to Exit Points

Use the Work with Registration Information (WRKREGINF) command to add your exit programs (see Figure 105 on page 149).

Make sure that the exit program source code is secured and that exit point maintenance is restricted.

6.6.7 Test Anonymous FTP Environment

Before you open up to the Internet, log on and test what kind of information you can access. Hack your own system and do it thoroughly.

6.7 Anonymous FTP Support, Write-Only

In case your Internet users are going to provide you with information, they must be allowed to store files on your Internet server. Examples for business applications are:

- Customers providing order data
- Text documents or any type of PC files

Allowing storing of data has some danger in it:

- Users are able to read other users' data that is stored in the same library or directory.
- Your disk space may be flooded.
- Any data with *PUBLIC(*CHANGE) can be altered.

Denial of service must be considered. Users might put large files onto your system and fill it up. Limiting the auxiliary storage of the anonymous user profile may handle this threat but you cannot control the content of what is sent to you.

The operations allowed are:

- Logging on using FTP.
- Listing the contents of a limited set of directories or libraries.
- Sending files to these directories or libraries.

Write-only Anonymous, Exit Programs

We need:

- Same server logon exit program as with Read-only anonymous.
- The public write directory should be separate from the public read directory.
- Server request exit program only allowing operation ID 3, 4, and 7 with library PUBLIC and directory /public and its subdirectories.

  All requests being logged to a PF, logging all data available.

  It is a variation of the read-only anonymous request validation exit program.

6.8 Logging and Audit

You must log what is happening when you connect your AS/400 system to the Internet, and you must study the reports from the logs. You cannot afford to let things get out of hand.

You have two options:

- Use the QAUDJRN audit journal.
- Write log entries to your own log file from your exit programs.

We recommend that you use both options.

6.8.1.1 Use QAUDJRN Audit Journal

You can use the QAUDJRN audit journal to log attempts to violate the security implementation on your AS/400 system such as:

- Unsuccessful attempts to log on
- Attempts to access objects without sufficient authority
- Attempts to change the authority implementation

You can use the functions in the Security Toolkit to start logging to the QAUDJRN and to print the reports from the audit journal. Refer to Section 3.4, “Logging and Auditing” on page 81 for an examples of how to audit TCP/IP applications.

6.8.1.2 Write Log Entries in Your Exit Programs

You can let your exit programs log information about the client to a log file specified in the program. You must decide what kind of information you want to log, and you must provide the programs to write the reports or display information from the log file.

6.9 Summary

To summarize our recommendations:

1. Do not use passwords on the Internet.
2. Only support Anonymous FTP.
3. Provide an exit program to select user profile.
4. Provide an exit program to determine allowed operations (for example, GET only).
5. Strictly limit access of FTP users.
6. Do not bother to check client's IP address.
7. Limit ANONYMOUS access to one directory or library. The access should be read-only.
8. Set the inactivity timer (INACTTIMO) in the FTP attributes to a low value to reduce the exposure when a user leaves an FTP session unattended.
9. Always keep track of what is going on (logging and auditing).

Chapter 7. TELNET Security

The AS/400 TELNET server allows a TCP/IP user on a remote TELNET client to sign on and run applications on the AS/400 system. The AS/400 system also supports the TELNET client but, from the security stand point, incoming TELNET, the TELNET server, represents the biggest exposure. In this chapter, we talk about how to protect your TELNET server.

The TELNET application is a useful tool on your intranet but it can be a big exposure to your system when it is connected to the Internet. The AS/400 TELNET server allows a TCP/IP user on a remote TELNET client system to sign-on and run applications on the AS/400 system.

We strongly recommend that you **do not** start the TELNET server on the Internet, but if you must, we explain what steps should be done to secure as much as possible TELNET on the AS/400 system.

7.1 Potential Exposures versus Benefits

Table 16. TELNET - Benefits versus Exposures

Benefits	**Exposures**
Access control	Unauthorized access
Use of an user ID and Password	Sniffing
Automatic virtual devices configuration	Increase of unauthorized attempts
Data management functions	Denial of service

The TELNET server provides many benefits to your network:

- TELNET provides an interactive sign-on display to anyone who attempts to enter your system.
- Once you have a sign-on display, you need a user ID and password to log on to your system. You need a password if you are using security level 20 or greater. If you can securely identify a TELNET client user, you can define individual access capabilities based upon the user profile.
- Easy to configure. TELNET automatically configures virtual devices, providing a large number of TELNET sessions to be started. A virtual device is a device description that does not have hardware associated with it.
- Easy access for remote technical support personnel. User profiles with special authorities such as *IOSYSCFG and *ALLOBJ can perform important functions on your TCP/IP configuration and object management remotely.
- TELNET provides better performance than using the IBM Workstation Gateway or the I/NET Webulator 5250 applications.
- If the client supports TN5250 or TN3270, the TELNET server provides good keyboard mapping and useability.

You need to be aware of the potential exposures you can have when you enable a TELNET server:

- The TELNET server cannot restrict a user from getting a sign-on display if the TELNET server is already started. There is no ″anonymous″ TELNET support.

- When you type your user ID and password, both flow ″in the clear″ across your network. Hackers on the Internet or on your intranet can use sniffers (line-tracing equipment) to access your logon passwords.
- The number of sign-on attempts is equal to the number of system sign-on attempts allowed multiplied by the number of virtual devices that can be created. This increases the number of attempts a hacker can try to log on to your system. Because of this, ″door-knob twisting″ attacks can turn into denial of service.
- The TELNET server application does not provide good logging procedures other than those provided by OS/400.

7.2 Tips and Techniques

This chapter provides some tips and techniques to protect your TELNET server:

- Consider using a Firewall. Firewalls protect a secure network from an untrusted network. Firewalls can control traffic by packet filtering and by disabling routes. They also can monitor traffic. Most routers offer some level of protection if configured properly.
- If you want to make your TELNET server available only in your intranet and your router has packet filtering capabilities, set up the router to reject TCP/IP sessions with an origin IP address that is outside your network.
- If you are thinking about linking your system to the Internet, you probably need to begin by revising some of your thinking about security. Please refer to Chapter 2, “Start Here by Securing OS/400®” on page 37 to make sure your system is secure enough to be linked with the Internet. Protecting your objects by using the OS/400 security functions protects your system if you make a mistake in protecting your TELNET server.
- Consider changing the default value for the SERVER parameter on the STRTCPSVR command. The default ships as *ALL. Change it to start only the applications you really need (*TELNET, for example).
- The command for starting the TCP/IP (STRTCP) and starting the TCP/IP servers (STRTCPSVR) are shipped with the public authority set to *EXCLUDE. Make sure that the authority has not been changed and review the list of users who are authorized to use the commands.
- Control *IOSYSCFG special authority to restrict who can configure TCP/IP. Restrict who can use the Start TCP/IP (STRTCP) command.
- Consider creating user profiles that use the TELNET server with as little user authority as possible. You can also set up these user profiles to access just a particular application or menu with the INLPGM or INLMNU parameters on the Create or Change User Profile commands (CRTUSRPRF or CHGUSRPRF).
- Although the QMAXSIGN system value applies to TELNET, you reduce the effectiveness of this system value if you set up your system to configure virtual devices automatically. When the QAUTOVRT system value has a value greater than 0, the unsuccessful TELNET user can reconnect and attach a newly-created virtual device. This can continue until one of the following occurs:
  1. All virtual devices are disabled and the system has exceeded the limit for creating new virtual devices.
  2. All user profiles are disabled.

3. The hacker succeeds in signing on to your system.

- You can use the QLMTSECOFR system value to restrict users with *ALLOBJ or *SERVICE special authority. The user or QSECOFR must be explicitly authorized to a device to sign-on. Thus, you can prevent anyone with *ALLOBJ special authority from using TELNET to access your system by ensuring that QSECOFR does not have authority to any virtual devices.
- Use the system values specified in Section 2.3.2, “Password Rules” on page 39 to promote non-trivial passwords since this is the first exposure for a hacker to sign-on to your system.
- Prevent someone from associating a user application (such as a Socket application) with the TCP/IP port that the system normally uses for TELNET (port 23).

7.3 Implementation Examples

You can use these implementation examples:

- Changing the SERVER parameter on the Start TCP/IP Servers (STRTCPSVR) command to start only the applications you really want. Figure 108 shows an example. You can use the following command:

  ```
  CHGCMDDFT CMD(STRTCPSRV) NEWDFT('SERVER(*TELNET)')
  ```

```
                    Change Command Default (CHGCMDDFT)

Type choices, press Enter.

Command  . . . . . . . . . . . . > STRTCPSRV     Name
  Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
New default parameter string . . > 'SERVER(*TELNET)'




F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this disp
F24=More keys
```

Figure 108. CHGCMDDFT Command Example

Note

If a system upgrade is done, the parameters you changed are set to their default values. You can create a CL program to change these parameters after every system upgrade using the CHGCMDDFT command.

- Changing the following system values to their recommended values, the user profile and the device are disable when the third incorrect attempt to sign-on to your system is reached. Refer to Table 17 on page 162 to see the recommended values.

Table 17. Recommended Values

System Value	Description	Recommended Setting
QMAXSIGN	Maximum consecutive, incorrect sign-on attempts (user profile or password incorrect).	3
QMAXSGNACT	What the system does when the QMAXSIGN limit is reached.	3 (Disable both user profile and device.)
QAUTOVRT	The number of virtual device descriptions that the system automatically creates if no device is available for use.	0
QDSPSGNINF	The information sign-on display is shown to the user.	1 (Display)
QLMTSECOFR	Limit security officer device access.	1 (Limit)

Note

We also recommend changing the QLMTSECOFR system value to 1 (explicit device access needed) so the most powerful profiles are not used from TELNET.

You can use the following commands to change these system values:

```
CHGSYSVAL SYSVAL(QMAXSIGN) VALUE(3)
CHGSYSVAL SYSVAL(QMAXSGNACN) VALUE(3)
CHGSYSVAL SYSVAL(QAUTOVRT) VALUE(0)
CHGSYSVAL SYSVAL(QDSPSGNINF) VALUE(1)
CHGSYSVAL SYSVAL(QLMTSECOFR) VALUE(1)
```

- Consider creating specific user profiles to use the TELNET application. Set it with as little authority as you can. Setting an initial program or menu is a good procedure to limit the user access. Also consider limiting the maximum storage allowed for this user profile. This prevents an unauthorized user from overflowing your system auxiliary storage. The following example shows a user profile you can create:

  ```
  CRTUSRPRF USRPRF(TELNETUSER) INLPGM(lib-name/prog-name)
            INLMNU(lib-name/menu-name) MAXSTG(1024)
  ```

  You can also use the Change Activation Schedule Entry (CHGACTSCDE) command to set this user profile to be available only between 7AM and 10PM, for example. Type the following command:

  ```
  CHGACTSCDE USRPRF(TELNETUSER) ENBTIME('7:00') DSBTIME('22:00')
             DAYS(*MON *TUE *WED *THU *FRI)
  ```

- Do not use trivial passwords. Create your own password rules or use the Configure System Security (CFGSYSSEC) command to do it for you. You can see more details about the CFGSYSSEC command in Section 2.3.2, "Password Rules" on page 39.
- Better yet, give your travelling personnel a list of passwords to be used serially every time they sign on to the AS/400 system. Create a program that (at sign off time) changes the user profile to use the next password in the list. This way, if the password is sniffed, it is not valid after sign off.

This technique has the exposure of the travelling emplyees writing down the passwords and somebody else accessing the information. Also, you must be extremelly careful when securing the file where the one-time passwords are stored. If possible, this information should be encrypted.

- Set the system value QLMTDEVSSN (Limit device session) to 1 (limit). This system value controls whether a user can sign on at more than one workstation. If a password was sniffed at the beginning of the session, no one else can sign on with the same user ID at the same time.
- Change message descriptions that provide information about the reason why a sign on fails. A common practice that hackers use is to create an application program that uses the TELNET port to get the user ID and the password. This application program can imitate a real sign-on display just to register your user ID and Password. The user can receive a silly message such as *Password is not correct* and go to the real sign-on display. The user did not realize that the password was logged on to a hacker program.

  Hackers want 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 6 on page 46 shows the recommended text.

Table 18. 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.

You can change the message text using the Change Message Description (CHGMSGD) command. Example:

```
CHGMSGD MSGID(CPF1107) MSGF(QCPFMSGF) MSG('Sign-on information is not
        correct.') SECLVL(*NONE)
CHGMSGD MSGID(CPF1120) MSGF(QCPFMSGF) MSG('Sign-on information is not
        correct.') SECLVL(*NONE)
```

Note

If a system upgrade is done, the text of the messages you changed are set to the default values. You can create a CL program to change these message descriptions after every system upgrade using the CHGMSGD command.

- Restrict the TELNET listening port to a QTCP user profile. To prevent someone from associating a user application such as a Socket application with the application the system normally uses for the TELNET server application, do the following steps:

  1. Type GO CFGTCP to display the Configure TCP/IP menu.
  2. Select option 4 (Work with TCP/IP port restriction).

3. On the Work with TCP/IP Port Restriction display, specify 1 (ADD).
4. For the lower port range, specify the decimal number of the application you want to protect. The TELNET port number is 23.
5. For the upper port range, you can specify *ONLY or the maximum range for more than one application.
6. For the protocol, specify *TCP.
7. 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 for a specific user, you automatically exclude all other users.

Figure 109 shows an example of how to restrict the use of the TELNET port (port 23) for the TCP protocol. Only a QTCP user profile can access port 23.

```
                      Work with TCP/IP Port Restrictions

                                                          System: SYSTEMA
Type options, press Enter.
  1=Add   4=Remove

         --Port Range---                  User
Opt      Lower     Upper     Protocol     Profile
 _       _____     *ONLY     ________     _______
 _          23     *ONLY       *TCP       QTCP




F3=Exit   F5=Refresh   F6=Print list   F12=Cancel   F17=Top   F18=Bottom
```

Figure 109. Restricting TELNET Port

- Set the TELNET Inactivity Timeout (INACTTIMO) value to a short time. This value is set using the Change TELNET Attributes (CHGTELNA) command and specifies the number of seconds the system allows a TELNET connection to remain inactive before it is ended. When a TELNET connection is inactive longer than the specified length of time, it is ended. This will prevent unattended TELNET sessions from being used by unauthorized persons.

  Note: TELNET does not use the QINACTITV, Inactive job time-out, system value.

7.4 Logging and Audit

- Make sure that the public authority to the STRTCP and STRTCPSVR commands are set to *EXCLUDE. You can use the following commands:

  ```
  DSPOBJAUT OBJ(STRTCP) OBJTYPE(*CMD)
  ```

```
                                    Display Object Authority                                    Page     1
5716SS1 V3R7M0  961108                                                          SYSTEMA   12/16/96  10:54:53
 Object . . . . . :   STRTCP                          Owner . . . . . :   QSYS
   Library  . . . . :     QSYS                        Primary group . . . :   *NONE
 Object type  . . . . :   *CMD
 Object secured by authorization list  . . . . . . . . . . . . :   *NONE
                          Object      -------------Object--------------    -------------------Data-------------------
 User          Group      Authority    Opr    Mgt    Exist    Alter    Ref    Read    Add    Update    Delete    Execute
 QSYS                     *ALL          X      X       X        X       X      X       X       X         X          X
 QSRV                     *USE          X                                      X                                     X
 QSRVBAS                  *USE          X                                      X                                     X
 QSYSOPR                  *USE          X                                      X                                     X
 QPGMR                    *USE          X                                      X                                     X
 *PUBLIC                  *EXCLUDE
                                  * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 110. DSPOBJAUT Report - Public Authority to STRTCP Command

```
DSPOBJAUT OBJ(STRTCPSVR) OBJTYPE(*CMD)
```

```
                                    Display Object Authority                                    Page     1
5716SS1 V3R7M0  961108                                                          SYSTEMA   12/16/96  10:55:05
 Object . . . . . :   STRTCPSVR                       Owner . . . . . :   QSYS
   Library  . . . . :     QSYS                        Primary group . . . :   *NONE
 Object type  . . . . :   *CMD
 Object secured by authorization list  . . . . . . . . . . . . :   *NONE
                          Object      -------------Object--------------    -------------------Data-------------------
 User          Group      Authority    Opr    Mgt    Exist    Alter    Ref    Read    Add    Update    Delete    Execute
 QSYS                     *ALL          X      X       X        X       X      X       X       X         X          X
 WEBMASTER                *USE          X                                      X                                     X
 QSRV                     *USE          X                                      X                                     X
 QSRVBAS                  *USE          X                                      X                                     X
 QSYSOPR                  *USE          X                                      X                                     X
 QPGMR                    *USE          X                                      X                                     X
 *PUBLIC                  *EXCLUDE
                                  * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 111. DSPOBJAUT Report - Public Authority to STRTCPSVR Command

- Verify that users have *IOSYSCFG special authority. They can change your TCP/IP configuration. You can use the following command:

  ```
  PRTUSRPRF TYPE(*AUTINFO) SELECT(*SPCAUT) SPCAUT(*IOSYSCFG) USRCLS(*ALL)
  ```

```
                                    User Profile Information                                    Page     1
5716SS1 V3R7M0  961108                                                          SYSTEMA   12/16/96  11:58:25
 Report type  . . . . . . . . . . :   *AUTINFO
 Select by  . . . . . . . . . . . :   *SPCAUT
 Special authorities  . . . . . :   *IOSYSCFG
                        ------------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
 SILVIO      *NONE        X     X     X    X     X     X     X     X    *SECOFR     *USRPRF     *NONE       *PRIVATE    *NO
 QFAXMSF     *NONE              X     X    X     X     X     X     X    *SECOFR     *USRPRF     *NONE       *PRIVATE    *NO
 QLPAUTO     *NONE        X           X    X     X     X                *SYSOPR     *USRPRF     *NONE       *PRIVATE    *NO
 QLPINSTALL  *NONE        X           X    X     X     X                *SYSOPR     *USRPRF     *NONE       *PRIVATE    *NO
 QSECOFR     *NONE        X     X     X    X     X     X     X     X    *SECOFR     *USRPRF     *NONE       *PRIVATE    *NO
 QSYS        *NONE        X     X     X    X     X     X     X     X    *SECOFR     *USRPRF     *NONE       *PRIVATE    *NO
 WEBMASTER   *NONE                    X                                 *USER       *USRPRF     *NONE       *PRIVATE    *NO
                                * * * * *  E N D  O F  L I S T I N G  * * * * *
```

*Figure 112. PRTUSRPRF Report - Users with Special Authority *IOSYSCFG*

- The system writes message ID CPF2234 to the QHST log for each unsuccessful attempt (password incorrect). You can monitor the QHST history log with the Display Log (DSPLOG) command or you can write a program to monitor the QHST log for these messages. If the program detected repeated unsuccessful attempts, it can end the TELNET server, for example. You can use the following command to monitor the CPF2234 message ID:

  ```
  DSPLOG MSGID(CPF2234)
  ```

Figure 113 on page 166 shows an example:

```
                          Display History Log Contents

Password from device QPADEV0001 not correct for user SILVIO.
Password from device QPADEV0004 not correct for user MARY.









Press Enter to continue.

F3=Exit    F10=Display all    F12=Cancel
```

Figure 113. DSPLOG Report - Searching for CPF2234 Messages

- If you start QAUDJRN and select *AUTFAIL as the object auditing value in the QAUDLVL system value, you can search for a PW journal entry (this represents someone trying to use an incorrect password or user ID) and find the user profile that attempted to sign on and the date and time the attempt was made. Figure 114 shows an example. You can use the following command:

  ```
  DSPAUDJRNE ENTTPY(PW)
  ```

```
                                                LIBRARY NAME . . . . QSYS
                                                FILE         LIBRARY      MEMBER       FORMAT
                                                QASYPWJE     QTEMP        QASYPWJE     QASYPWJE
                                                DATE . . . . . . . . 12/18/96
                                                TIME . . . . . . . . 09:54:00

12/18/96  09:54:00                                                                                                  PAGE    1
  VIOLATION USER       USER       DEVICE                  REMOTE   LOCAL    NETWORK  JOB        JOB        JOB    DATE       TIME
  TYPE      PROFILE    NAME       NAME                    NAME     NAME     ID       NAME       USER       NUMBER
PW    P     QSYS       JOHN       QPADEV0005                                         QINTER     QSYS       012476 121696 13:20:04
PW    P     QSYS       SILVIO     QPADEV0001                                         QINTER     QSYS       012476 121796  7:11:06
PW    U     QSYS       MARY       QPADEV0004                                         QINTER     QSYS       012476 121796  9:50:58
PW    P     QSYS       CLAUS      QPADEV0004                                         QINTER     QSYS       012476 121796  9:52:45
* * *  E N D  O F  R E P O R T  * * *
```

Figure 114. DSPAUDJRNE Report - PW Entries in QAUDJRN Journal

7.5 Summary

- Avoid allowing incoming TELNET over the Internet.
- If you **must** do it, consider a firewall to filter incoming TELNET sessions.
- Carefully review all of the system values related to interactive sessions and passwords.

Chapter 8. SLIP Security

SLIP stands for Serial Line Internet Protocol. SLIP is used when you run TCP/IP over dial-up connections through an asynchronous RS232 port. SLIP provides an easy way for remote systems to access your AS/400 system using TCP/IP over a switched line. The AS/400 system supports dial-out to remote systems and dial-in from remote systems to your AS/400 system. The system that dials out is known as the **SLIP client** and the system that dials in is known as the **SLIP server**.

In this chapter, we discuss how to configure SLIP with security in mind:

- What are the main configuration components that impact the security of your SLIP connection.
- Auditing changes to your initial set up.

8.1 Potential Exposures versus Benefits

SLIP is only a protocol that can be used to establish a link to your AS/400 system from the Internet through an ISP. SLIP can also be used outside the Internet to set up a connection between your AS/400 system and remote systems. However, you cannot be sure that the remote caller is an authorized user; it might be a hacker who attempts to get access to your AS/400 system. After the SLIP link is established, the normal TCP/IP data flow is accomplished. Therefore, the applications you plan to run must be secured. See the corresponding chapters for securing the application you want to run on your AS/400 system. SLIP is used mostly for a more or less closed user group (for example, a service or sales force) who needs occasional access to the AS/400 system.

8.2 Tips and Techniques

SLIP is controlled by configuration files. You must configure (using the Work TCP Point-To-Point (WRKTCPPTP) command) your AS/400 system to receive calls (*ANS) and make calls (*DIAL) to initiate a SLIP connection.

Tip

The WRKTCPPTP command requires *IOSYSCFG special authority.

The point-to-point TCP/IP configuration file stores information that is used to establish a SLIP connection with an AS/400 system. When you start a SLIP connection to an AS/400 system, you simply establish a link. You have not yet signed on and started any TCP/IP application. Therefore, you do not necessarily need an AS/400 user profile to start a SLIP connection to an AS/400 system as you do in Client Access/400. After the link is established, you can browse the HTTP server or sign on to TELNET, FTP, or any active TCP/IP application. The following list provides security hints for SLIP configurations:

- Avoid having both *DIAL and *ANS point-to-point TCP/IP configuration files on the same system. This prevents your system from being used as a stepping stone for hackers. You do not want a hacker to use your system to break into other systems.

- Configure your SLIP *ANS connection to disable IP forwarding. This prevents a remote user from getting access to your corporate network and attack other systems connected to this network.
- Answer (*ANS) point-to-point TCP/IP configuration files can be secured with a system access authorization list. If an authorization list is specified, only the user profiles specified in the authorization list are allowed to connect to the AS/400 system from a remote system. This user profile should not allow an actual sign-on, but just establish the link. You can do this by setting INLNMNU(*SIGNOFF) and INLPGM(*NONE).
- Users on your AS/400 system might want to establish dial-out connections to systems that require user validation. The connection dialog script on your AS/400 system must send a user ID and password to the remote system. The AS/400 system provides a secure method for storing the password in encrypted form. It decrypts the password before sending it. This is done by setting the Retain server security data (QRETSVRSEC) system value to **1 (Retain data)** and specifying the remote user ID and password in the dial out configuration profile. SLIP passwords (such as TELNET and FTP passwords) are sent unencrypted or ″in the clear″. However, unlike TELNET and FTP, the SLIP password is sent before the systems establish the SLIP link.
- SLIP uses customized SLIP connection dialog scripts. SLIP dialog scripts are used to pass parameters such as user ID and passwords between systems. See the following conversation example:

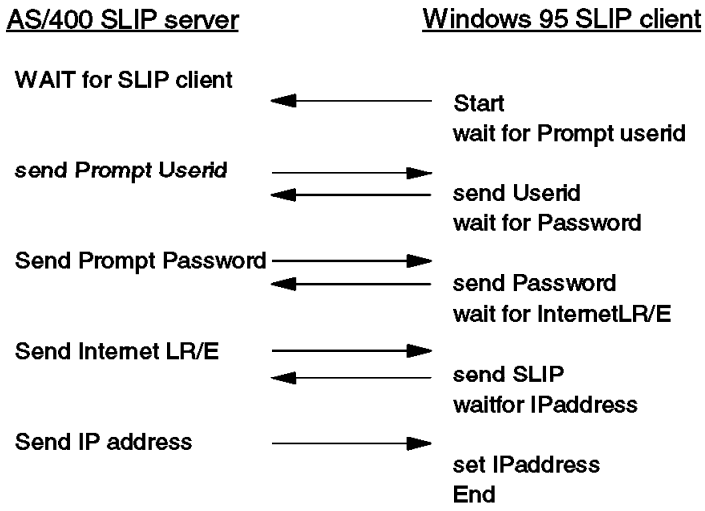


Figure 115. SLIP Conversation Example

The default file for storing SLIP connection dialog scripts is QUSRSYS/QATOCPPSCR. The public authority for this file is *USE. We recommend using a source physical file located in a separate library such as ITSOIC400 in our example. Use the CRTSRCPF command to copy the examples delivered by IBM from QUSRSYS/QATOCPPSCR to your

newly-created file (such as HARMSLIP in our example) and change a dialog script to fit the requirements of the remote system if needed.

8.3 Implementation Examples

The objective of this section is to provide specific examples that show how to apply the general security-related recommendations for SLIP connections discussed in the previous section.

8.3.1 Securing Dial-In SLIP Connections

The client that wants to connect to your AS/400 system must present itself by sending a user ID and a password. The user ID and password is only used to establish a connection; thereafter, TCP/IP applications such as TELNET, FTP, and so on may be started.

In this example, we show how to set up a **dial-in point-to-point TCP/IP configuration file** to validate remote systems that dial in to your AS/400 system by requesting a password and user ID.

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.

   The user profile should have limited authority on the system. Set the following values to prevent the user from signing on to the system:

   Initial menu (INLMNU) of *SIGNOFF

   Initial program (INLPGM) of *NONE

```
                        Create User Profile (CRTUSRPRF)

Type choices, press Enter.

User profile . . . . . > HARMDIALIN    Name
User password  . . . . .   *SAME         Name, *SAME, *NONE
Set password to expired  .   *NO          *SAME, *NO, *YES
Status . . . . . .   *ENABLED      *SAME, *ENABLED, *DISAB
User class . . . . . .   *USER         *SAME, *USER, *SYSOPR..
Assistance level . . . . . .   *SYSVAL       *SAME, *SYSVAL, *BASIC.
Current library  . . . . . .   *CRTDFT       Name, *SAME, *CRTDFT
Initial program to call  . . .   *NONE         Name, *SAME, *NONE
  Library  . . . . . .                    Name, *LIBL, *CURLIB
Initial menu . . . . . .   *SIGNOFF      Name, *SAME, *SIGNOFF
  Library  . . . . . .                    Name, *LIBL, *CURLIB
Limit capabilities . . . . . .   *YES          *SAME, *NO, *PARTIAL, *
Text 'description' . . . . . .   User Profile to establish a SLIP connection


                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 116. User Profile HARMDIALIN used for Dial-In SLIP Connection

2. Create an authorization list referred by the ***ANS point-to-point TCP/IP configuration file**. This authorization list contains the AS/400 user IDs used by the SLIP remote user (HARMDIALIN).

```
                    Create Authorization List (CRTAUTL)

Type choices, press Enter.

Authorization list . . . . . > SLIPAUTL      Name
Text 'description' . . . . . > 'Authorization list for SLIP user'


                           Additional Parameters

Authority  . . . . . .   *exclude      *CHANGE, *ALL, *USE, *E



F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this disp
F24=More keys
```

*Figure 117. Authorization List used by *ANS Configuration File*

3. Add the AS/400 user profile **HARMDIALIN** created in the first step to the authorization list **SLIPAUTL** created in the second step. You can create a unique authorization list for each point-to-point TCP/IP configuration file, or you can create an authorization list that several point-to-point TCP/IP configuration files share. Set ***PUBLIC** to **EXCLUDE** to make sure that only specific authorized users can use the dial-in connection.

```
                          Edit Authorization List

Object . . . . . :   SLIPAUTL        Owner  . . . . . :   A96032
  Library  . . . . :     QSYS          Primary group  . . . :   *NONE

Type changes to current authorities, press Enter.

             Object    List
User        Authority  Mgt
HARMDIALIN  *USE
WEBMASTER   *ALL        X
*PUBLIC     *EXCLUDE


F3=Exit   F5=Refresh   F6=Add new users                                 Bottom
F11=Display detail object authorities   F12=Cancel   F24=More keys
(C) COPYRIGHT IBM CORP. 1980, 1996.
```

Figure 118. EDTAUTL - Add Users to Initiate SLIP Dial-In Connections

4. Use the WRKTCPPTP command to set up an ***ANS point-to-point TCP/IP configuration file** that has the following characteristics:
   - The point-to-point TCP/IP configuration file 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.

     Tip

     The system ships with several sample dialog scripts that provide this function.

     Keep in mind that the AS/400 dialog script that is chosen must match the remote user's dialog script.

```
                    Add TCP/IP Point-to-Point *ANS Profile
                                                          System:   SYSNAME
Name:   SLIPDIALIN
Text

Type choices, press Enter.

TCP/IP information:
  Protocol type . . . . . . . . . . :   *SLIP
  Local interface address . . . . . .   x.x.xx.xxx        Address, F4 for list
  Remote IP address . . . . . . . . .   x.x.xx.xxx        Address
  Maximum transmission unit . . . . .   576               576-1006
  Allow proxy ARP . . . . . . . . . .   N                 Y=Yes, N=No
  Add default route . . . . . . . . .   N                 Y=Yes, N=No

Physical line information:
  Line description  . . . . . . . . .   SLIP01            Name
  Line type . . . . . . . . . . . . :   *ASYNC
  Autocreate controller and device      Y                 Y=Yes, N=No
    Remote location name  . . . . . .                     Name
                                                                      More...
F2=Change modem information   F3=Exit   F4=List   F9=Command line
F12=Cancel
```

*Figure 119. WRKTCPPTP - Create the *ANS Point-to-Point TCP/IP Configuration File*

- Set the parameter ″Use connection dialog script″ to **Y** and specify the location of your answer dialog script. In our example, this is member **ANSWIN95** in file **HARMSLIP** in library **ITSOIC400**.

```
                    Add TCP/IP Point-to-Point *ANS Profile
                                                          System:   SYSNAME
Name:   SLIPDIALIN
Text

Type choices, press Enter.

Modem information:
  Use a modem . . . . . . . . . . . .   Y                 Y=Yes, N=No
    Modem information name                                F4 for list
      IBM 28800 7852-010

Script source information:
  Use connection dialog script  . . .   Y                 Y=Yes, N=No
    Member  . . . . . . . . . . . . .   ANSWIN95A         Name
    File  . . . . . . . . . . . . . .   SLIPSCRIPT        Name
      Library . . . . . . . . . . . .     ITSOIC400       Name
    ASCII character set identifier      00819             1-65533, *DFT


                                                                      More...
F2=Change modem information   F3=Exit   F4=List   F9=Command line
F12=Cancel
```

Figure 120. WRKTCPPTP - SLIP Answer Script Specification

- Insert the name of the authorization list **SLIPAUTL** created in Figure 117 on page 170.
- Set the Allow IP datagram forwarding to **N** to prevent remote users from accessing your local network.

```
                    Add TCP/IP Point-to-Point *ANS Profile
                                                          System:   SYSNAME
Name:   SLIPDIALIN
Text

Type choices, press Enter.

Local system security:
  Allow IP datagram forwarding  . . .   N                 Y=Yes, N=No
  System access authorization list      SLIPAUTL        *NONE, Name

                                                                      Bottom
F2=Change modem information   F3=Exit   F4=List   F9=Command line
F12=Cancel
```

Figure 121. WRKTCPPTP - SLIPAUTL Contains Remote SLIP User

Figure 122 shows the AS/400 script from Member **ANSWIN95A**, File **SLIPSCRIPT**, Library **ITSOIC400**.

```
        *************** Beginning of data ********************************
0001.00 * SERVER CONNECTION SCRIPT EXAMPLE WITH LOGIN AND PASSWORD / WIN95
0002.00 (PROMPT)
0003.00 & Userid:
0004.00 (USERID)
0005.00 & Password?
0006.00 (PASSWORD)
0007.00 & InternetLR/E>
0008.00 (PROMPT)
0009.00 & Your address is (IPADDR)
0010.00 * END OF SERVER CONNECTION SCRIPT EXAMPLE
        ****************** End of data **********************************
```

Figure 122. AS/400 Server Script for Windows 95 Connection

Figure 123 on page 173 shows the default SLIP dialog script used on the Windows 95 client shipped by Windows 95.

```
 This is a script file that demonstrates how
; to establish a slip connection with a host.
;
; A script file must have a 'main' procedure.
; All script execution starts with this 'main'
; procedure.
;


; Main entry point to script
;
proc main

   ; Delay for 2 seconds first to make sure the
   ; host doesn't get confused when we send the
   ; two carriage-returns.
   delay 2
   transmit "^M^M"

   ; Wait for the login prompt before entering
   ; the user ID

   waitfor "serid:"
   transmit $USERID
   transmit "^M"

   ; Enter the password

   waitfor "assword?"
   transmit $PASSWORD
   transmit "^M"

   waitfor "InternetLR/E>"
   transmit "slip"
   transmit "^M"

   ; An alternative to the following two lines is
   ;
   ;    set ipaddr getip 2
   ;
   ; since we know that my address is the second one given.

    waitfor "Your address is "
    set ipaddr getip

endproc
```

Figure 123. Windows 95 Default Script File: Slip.scp.

8.3.2 Securing Dial-Out SLIP Connections

When you create a point-to-point TCP/IP configuration file for a remote session that requires validation, do the following steps:

1. Ensure that the Retain Server Security Data (QRETSVRSEC) system value is set to 1 (Yes). This system value must be set to Yes to specify the remote service access password. Use the WRKSYSVAL QRETSVRSEC command.

```
                              Change System Value

System value . . . . . :   QRETSVRSEC
Description  . . . . . :   Retain server security data


Type choice, press Enter.

  Retain server security
    data . . . . . .   1             0=Do not retain data
                                      1=Retain data


F3=Exit   F5=Refresh   F12=Cancel
```

Figure 124. System Value QRETSVRSEC Set to Retain Data

2. Use the WRKTCPPTP command to create a *DIAL point-to-point TCP/IP configuration file. In our example, we get the local and remote TCP/IP address provided by your Internet Service Provider (ISP) to *DYNAMIC. If you get a fixed IP address from your ISP, insert the appropriate IP address in the fields.

```
                    Add TCP/IP Point-to-Point *DIAL Profile
                                                          System:   SYSNAME
Name:   HARMDIALOU
Text

Type choices, press Enter.

TCP/IP information:
  Protocol type . . . . . . . . . . :   *SLIP
  Local interface address . . . . . .   *DYNAMIC          Address, *DYNAMIC
  Remote IP address . . . . . . . . .   *DYNAMIC          Address, *DYNAMIC
  Request header compression  . . . .   Y                 Y=Yes, N=No
  Maximum transmission unit . . . . .   576               576-1006
  Add default route . . . . . . . . .   N                 Y=Yes, N=No
  Additional name server  . . . . . .   *NONE             Address, *NONE

                                                                      More...

F2=Change modem information   F3=Exit   F4=List   F9=Command line
F12=Cancel
```

Figure 125. Dial In Point-to-Point TCP/IP Configuration File

3. Set the Use connection dialog script to **YES** and specify the dialog script member **DIALIGN** in file **SLIPSCRIPT** in Library **ITSOIC400**. The script sends the user ID and password to the remote system. Your AS/400 system ships with several sample dialog scripts that provide this function.

```
                  Add TCP/IP Point-to-Point *DIAL Profile
                                                        System:   SYSNAME
Name:   HARMDIALOU
Text

Type choices, press Enter.

Script source information:
  Use connection dialog script  . . .   Y               Y=Yes, N=No
    Member  . . . . . . . . . . . . .   DIALIGN         Name
    File  . . . . . . . . . . . . . .   SLIPSCRIPT       Name
      Library . . . . . . . . . . . .     ITSOIC400     Name
    ASCII character set identifier      00819           1-65533, *DFT


                                                                  More...

F2=Change modem information   F3=Exit   F4=List   F9=Command line
F12=Cancel
```

*Figure 126. Specify *Dial Dialog Script*

4. Specify the remote service phone number, the remote service access name, and remote service access password provided by the ISP (in our example, IBM Global Network (IGN)). The remote service access password is encrypted in a protected area on your AS/400 system (system value **QRETSVRSEC**). When the system runs the script, the system decrypts the password and sends it to the remote system.

```
                  Add TCP/IP Point-to-Point *DIAL Profile
                                                        System:   SYSNAME
Name:   HARMDIALOU
Text

Type choices, press Enter.

Remote system access information:
  Remote service phone number
    nnnnnnnnnn
  Remote service access name
    IGN user
  Remote service access password
    xxxxxxxx

                                                                    Bottom

F2=Change modem information   F3=Exit   F4=List   F9=Command line
F12=Cancel
```

Figure 127. Specify Remote Access Information

For a more detailed description about SLIP scripts, please refer to the *TCP/IP Configuration and Reference*.

Because of the different security practices and capabilities of your communication partners, you might want to create different configuration profiles for different requesting environments. Use the STRTCPPTP command to set up your system 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.

8.4 Logging and Audit

Once you have configured a secure SLIP connection following the discussions and examples in the previous sections of this chapter, you should make sure that your configuration is not being altered by unauthorized users and there are no holes in your security strategy. The purpose of this section is to list the main AS/400 components that affect SLIP security and how you should be auditing and logging any changes to them that might compromise the security of your system.

The following objects are security sensitive and should be audited regularly:

- Message CPF2234 (Password not correct.)
- SLIP point-to-point TCP/IP configuration files (for example, SLIPDIALIN)
- SLIP AS/400 user profile (for example, HARMDIALIN)
- SLIP connection dialog script, (SLIPSCRIPT containing the dialog scripts)
- Authorization list for SLIP user (for example, SLIPAUTL)

1. The system writes message ID **CPF2234** to the QHST log for each unsuccessful attempt to sign on to your system. Use the DSPLOG MSGID(CPF2234) command to display all rejected attempts to access your system. You can also write a program to monitor the QHST log for these messages. If the message appears frequently, the program can trigger any action (for example, end the point-to-point line by using the ENDTCPPTP CFGPRF(SLIP01) OPRMODE(*DIAL) command).

```
                          Display History Log Contents

Password from device DFV-Einheit not correct for user HARMDIALIN.
Password from device DFV-Einheit not correct for user HARMDIALIN.
Password from device DFV-Einheit not correct for user HARMDIALIN.
Password from device DFV-Einheit not correct for user HARMDIALIN.


Press Enter to continue.                                              Bottom

F3=Exit   F10=Display all   F12=Cancel
```

Figure 128. Display QHST Shows CPF2234 - Password Not Correct

2. Check that no one changes your point-to-point TCP/IP configuration file by using the change/usage information from the object description. Use the following command to set the object auditing value of your point-to-point TCP/IP configuration file to ***CHANGE** to activate change auditing:

   ```
   CHGOBJAUD OBJ(QSYS/SLIPDIALIN) OBJTYPE(*LIND) OBJAUD(*CHANGE)
   ```

```
                      Change Object Auditing (CHGOBJAUD)

Type choices, press Enter.

Object . . . . . . . . . . . . . > SLIPDIALIN    Name, generic*, *ALL
  Library  . . . . . . . . . . . >   QSYS        Name, *LIBL, *USRLIBL..
Object type  . . . . . . . . . . > *LIND         *ALL, *ALRTBL, *AUTHLR.
Object auditing value  . . . . . > *CHANGE       *NONE, *USRPRF, *CHANGE


                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

*Figure 129. CHGOBJAUD - Object Auditing Value Set to *CHANGE*

3. The remote SLIP user should not be authorized to log on to your AS/400 system as an AS/400 user if there is no need for it. To make sure that no one changes the AS/400 user profiles used by the remote SLIP user, verify the user profiles by using the following command:

   PRTUSRPRF TYPE(*ENVINFO) SELECT(*USRCLS) USRCLS(*USER)

```
                           Display Spooled File
File  . . . . . :   QPSECUSR                        Page/Line   1/1
Control . . . . .                                   Columns     1 - 78
Find  . . . . . .
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
                                        User Profile Information
5763SS1 V3R2M0  960517
 Report type  . . . . . . . . . . :   *ENVINFO
 Select by  . . . . . . . . . . . :   *USRCLS
 User class . . . . . . . . . . . :   *USER
                            Initial        Initial           Job
 User           Current     Menu/          Program/       Description
 Profile        Library     Library        Library          Library
 ANONYMOUS      *CRTDFT     MAIN           *NONE           QDFTJOBD
                            *LIBL                          QGPL
 WEBMASTER      *CRTDFT     MAIN           *NONE           QDFTJOBD
                            *LIBL                          QGPL
 GUEST          *CRTDFT     MAIN           *NONE           QDFTJOBD
                            *LIBL                          QGPL
 HARMDIALIN     *CRTDFT     *SIGNOFF       *NONE           QDFTJOBD
                                                           QGPL
                                                                   More
F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys
```

Figure 130. PRTUSRPRF - Check HARMDIALIN User Profile

8.5 Summary

SLIP is only used to establish a connection between a client and your AS/400 system. When the connection is established, any TCP/IP application such as TELNET or FTP can be used.

To secure SLIP:

- Avoid having both *ANS and *DIAL configuration files on the same system. This prevents your AS/400 system from being used as a stepping stone.
- Disable IP forwarding.

- Secure the *ANS configuration file using a system access authorization list. Do not allow the user profile specified in this authorization list to sign on to your AS/400 system.
- If you have configured a *DIAL connection, store the password in an encrypted form by setting the QRETSVRSEC system value to 1.
- Store your SLIP connection dialog script in your own source physical file and secure this file.
- Monitor the QHST log for the CPF2234 message (Password not correct).
- Set the ″Object auditing value″ to *CHANGE for your configuration files and use QAUDJRN to check if they are changed.
- Use QAUDJRN to check if SLIP user profiles are changed.

Chapter 9. I/NET′s Commerce Server/400 Security

This chapter explains what steps should be taken to secure I/NET′s Commerce Server/400 and Webulator/400 on the AS/400 system, and addresses techniques to minimize exposures as they relate to I/NET Commerce Server/400.

Risk analysis is the process of identifying security risks, determining their magnitude, and identifying areas needing safeguards. Risk analysis is a part of risk management and is synonymous with risk assessment according to the *National Computer Security Center (NCSC)*. It is assumed that you have completed your risk assessment, developed a security policy, and have made a decision to implement a Web server on the Internet. It is also assumed that you have designed and implemented network level security and have conducted an OS/400 security review (see Chapter 2, “Start Here by Securing OS/400®” on page 37).

9.1 Potential Exposures versus Benefits

To be able to make an informed business decision on the viability of an Internet solution, you must understand the exposures as well as the benefits to your organization. Since you are reading this book, it is assumed that you understand the benefits of having a Web server and, therefore, we primarily address exposures. The key to understanding the risk of operating a Web Server is understanding the possible exposures. Just who are the bad guys and what can they do to you?

9.1.1 Exposures

- Encryption keys distribution:

  A secure Web server requires encryption keys. The private encryption key is an asset that must be protected at all costs. With a copy of your private key, anyone can pose as you or your company.

- Improper configuration:

  A requirement for proper configuration is that security is implemented on your AS/400 system as outlined in Chapter 2, “Start Here by Securing OS/400®” on page 37. The configuration must support the intent of the policy. Initial testing and ongoing auditing are crucial to compliance. Refer to Section 9.5, “Audit Considerations” on page 239 for further information on auditing.

- Exposure of mixing HTTP and HTTP over SSL (HTTPS) protocols:
  - ID/Passwords:

    The transition from anonymous serving to discretionary access control introduces the exposure of unauthorized access to information through compromised ID/passwords. Policies and procedures that administer these values must be developed and maintained.

  - Mixing HTTP and HTTP over SSL in the same server:

    Utilizing both SSL and HTTP protocols on the same server can expose ID/Passwords to detection. The user authentication function of access control requires an ID/Password pair to be sent to the server. Improperly behaving browsers can, in conjunction with improper server

configuration, transmit ID/Passwords in the clear. This exposure can be reduced by running two instances of Commerce Server/400, one using only HTTP and the other using HTTP over SSL. This requires more system resources, but the design should be less susceptible to configuration errors and less vulnerable to rouge browsers.

Also, using both protocols in the same directory creates an exposure to unauthorized document access. Figure 131 shows the warning display that is shown when both HTTP and HTTP over SSL protocols are configured for the same directory.

```
                          Set Allowed Protocols

                             *** WARNING ***

You have selected to allow the non-secure "HTTP" protocol
in the same directory with SSL (a secure protocol).  While
this is an acceptable practice,it may result in the delivery
of secured document or form information via a non-secured
session.

This warning display is presented to ensure that
this is your desire.



Press Enter to continue setting the new protocol scheme.

Press F12 to cancel - the protocol scheme will not be changed.
```

Figure 131. Mix Protocols Warning Message

- Webulator/400:

  Webulator/400, which is the I/NET workstation gateway product, allows the Web server to present a 5250 display through a Web browser. In a controlled environment, this is no greater exposure than any other non-programmable terminal or workstation running 5250 emulation software. However, connecting the Web server to the Internet removes physical security and relies solely on how you have secured your AS/400 system as discussed in Chapter 2, “Start Here by Securing OS/400®” on page 37.

9.1.2 Benefits of Commerce Server/400

- In addition to the general benefits of a Web server, Commerce Server/400 provides the technological benefit of encrypted transmissions. (See Section 1.9.6, “What is Secured Sockets Layer (SSL)?” on page 31 for information on the SSL protocol.) This is strongly recommended when transmitting ID/Password pairs or conducting commercial transactions over the INTERNET.
- Webulator/400, in conjunction with SSL, allows 5250 applications to be run securely over the Internet.

9.2 Commerce Server/400 Tips and Techniques

This section includes recommendations and examples for designing and maintaining a secure Web server. Section 9.2.1, “Accessing Information through Commerce Server/400” on page 181 and Section 9.2.2, “Webulator/400” on page 184 contain excerpts from I/NET′s online documentation for Commerce

Server/400 and Webulator/400. Refer to this documentation for a complete understanding of functions.

http://www.inetmi.com/pubs/usrguide.htm

9.2.1 Accessing Information through Commerce Server/400

When configuring your Web server, you should think about the kinds of information on your AS/400 system and to whom you want to make it available. There are three basic categories defined by their availability through Commerce Server/400:

- Information unavailable through Commerce Server/400:

  This is information that you do not want anyone to access through the Web server.

  The two methods you have for protecting this information are OS/400 authority and Commerce Server/400 scope control. OS/400 authority is strong but requires some effort to configure and maintain. Web Server/400 scope control should be easier to maintain and is already set up when you install the server.

- Information available to all users that have access to the Web server:

  This is information that you want to be generally available. It may include press releases and other announcements or marketing material. If your Web server is only available through an internal network, you might include other information here such as company-wide announcements.

- Information available to a subset of users that have access to the Web server:

  This is information that should be available to some users who have access to the Web server, but not all. It might include information only available to customers, or information only available internally, even though the Web server itself is available to the public through the Internet.

  The method for protecting this information is *access control*. Using access control, you can specify (on a per directory basis if you want) who can access information, either based on the machine they are using or based on a user name and password they enter, or a combination of the two.

Note that this chapter only describes protecting your AS/400 information in regards to Commerce Server/400. Other software such as Telnet and FTP pose additional concerns that are not addressed here.

9.2.1.1 Commerce Server/400 and OS/400 Authority

Keep in mind that you must always protect your resources with multiple layers of security. If one layer is compromised, there is another layer between your resources and the hacker. Commerce Server/400 has many features that make your AS/400 Web server more secure, but you must always assume that if a hacker can get to the guts of your server, you still have the configured OS/400 security (your main line of defense) to protect it.

The server jobs run under the Commerce Server/400 server user profile, WWWUSER, by default. The server user profile should have the following authorities:

- *USE to any documents you want to make available to browsers
- *USE to any scripts the server should be able to execute

- *USE to the WWWSERVER/WWWDAEMON program and to the QSYS/QSYSNOMAX job queue
- The server user profile (WWWUSER) must be registered in the system directory in order to access Document Library Services (DLS) folders and documents.

To protect information from being accessed through Commerce Server/400, you must ensure that the server user profile only is authorized to the information it is set up to serve. You can do this by either specifically excluding WWWUSER or by using *PUBLIC=*EXCLUDE from all other information.

Webmaster is the title usually given to the one individual in the organization responsible for the Web server. The webmaster uses commands such as CGHWWWCFG, CHGWWWDIR, CHGWWWSEC, and WRKWWWINCL to configure and update the Web server configuration. This person needs enough authority to configure and manage the server. Figure 132 on page 183 is an example of a CL source listing for creating a webmaster user profile. The listing can be edited to reflect your actual directory structure. Using this concept, it is easy to create different webmaster profiles for each server that is running. It may be desirable to further divide authority by excluding the webmaster from some objects such as user lists and the key list file.

```
PGM /* CRTWWWMSTR */
CRTUSRPRF USRPRF(WWWMSTR) PASSWORD(WWWMSTR) PWDEXP(*YES) +
            TEXT('WEBMASTER FOR COMMERCE SERVER/400')

GRTOBJAUT OBJ(QSYS/WWWSERVER) OBJTYPE(*LIB) USER(WWWMSTR) AUT(*CHANGE)

GRTOBJAUT OBJ(WWWSERVER/*ALL) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(WWWSERVER/WWWDAEMON) OBJTYPE(*PGM) USER(WWWMSTR) AUT(*OBJMGT *EXECUTE)

GRTOBJAUT OBJ(QSYS/WWWUSER) OBJTYPE(*USRPRF) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/ADDWBL*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/ADDWWW*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/CHGWBL*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/CHGWWW*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/CRTWWW*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/DLTWWW*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/ENDWWW*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/SETWWW*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/STRWWW*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/WRKWBL*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

GRTOBJAUT OBJ(QSYS/WRKWWW*) OBJTYPE(*ALL) USER(WWWMSTR) AUT(*USE)

CHGAUT OBJ('/WWWSERV') USER(WWWMSTR) DTAAUT(*RWX) OBJAUT(*ALL)

CHGAUT OBJ('/WWWSERV/CFG') USER(WWWMSTR) DTAAUT(*RWX) OBJAUT(*ALL)

CHGAUT OBJ('/WWWSERV/CFG/*') USER(WWWMSTR) DTAAUT(*RWX) OBJAUT(*ALL)

CHGAUT OBJ('/WWWSERV/LOGS') USER(WWWMSTR) DTAAUT(*RWX) OBJAUT(*ALL)

CHGAUT OBJ('/WWWSERV/LOGS/*') USER(WWWMSTR) DTAAUT(*RWX) OBJAUT(*ALL)

CHGAUT OBJ('/WWWSERV/KEY') USER(WWWMSTR) DTAAUT(*RWX) OBJAUT(*ALL)

CHGAUT OBJ('/WWWSERV/KEY/*') USER(WWWMSTR) DTAAUT(*RWX) OBJAUT(*ALL)

CHGAUT OBJ('/WWWSERV/WEBDOCS') USER(WWWMSTR) DTAAUT(*RWX) OBJAUT(*ALL)

CHGAUT OBJ('/WWWSERV/WEBDOCS/*') USER(WWWMSTR) DTAAUT(*RWX) OBJAUT(*ALL)

ENDPGM
```

Figure 132. CL Source for Creating Webmaster Profile

Note: The WBL* objects are for Webulator/400 and are not necessary for Commerce Server/400.

9.2.1.2 Commerce Server/400 Scope Control

Scope control refers to configuration values that affect the scope of the information the server attempts to retrieve.

9.2.1.3 Commerce Server/400 Access Control

While OS/400 authority is strong, it is based on the user profile of the person starting the server and on the server user profile. The server user profile either has access to an object or not. Access control takes into account information about the person requesting the information such as the workstation they are using and the user name and password they enter.

9.2.1.4 IFS/Root File System Security

It is crucial to understand the root file structure and its security characteristics in order to properly configure the SSL server. Refer to Section 2.6, “Integrated File System Security” on page 51.

9.2.2 Webulator/400

A properly configured Webulator/400 should be considered a secure means of delivering access to 5250 applications and data across the Internet. This section is intended to help explain and assist in setting up the security for both the Webulator/400 product and the AS/400 system running Webulator/400. The topics covered should not be considered the only security areas to address nor the only material to consider.

9.2.2.1 Sign-On Methods

Webulator/400 requires a user profile and password to sign on to the AS/400 system. The sign-on process can be configured within the directory based configuration using one of three methods. Each Webulator URL specified within the directory based configuration file allows one of the following methods to be configured:

- Automatic sign on:

  This method keeps all user profile names and passwords hidden from the user. The directory-based configuration file allows the webmaster to configure a user profile name associated with a *WEBULATOR URL. In addition to specifying the user profile name within the directory based configuration file, the person configuring the Webulator/400 product also adds the user profile name and the corresponding password to the Webulator user configuration file (the WRKWBLUSR command or ADDWBLUSR command can be used to add these values).

  Note: Since the password in this file is the AS/400 password, ensure that whenever the password is changed for this user profile, it is also changed within the Webulator/400 user configuration file. If it is not, the URL will fail, the invalid sign-on attempt is logged in the system journals (if enabled), the user profile may be disabled (if the system values are enabled to do so), and the virtual terminal device is varied off (if the system values are enabled to do so).

  Note: The password in this file is stored in plain text and should be protected from the Web Server/400 user profile and other non-authoritative (*PUBLIC) user profiles using AS/400 security. You may allow the user to specify the initial program, menu, or library through a query string if you enable the AllowSignonOverride option of the sign-on method configuration entry.

- User authentication:

  The user ID and password sent on the request from the browser is the AS/400 user profile name and password. The user ID and password are encoded using the base64 encoding of MIME (UUENCODED). Essentially it is the same algorithm used to encode an FTP or Telnet user ID and password. If you are not using Commerce Server/400, this should not be considered a secure encoding algorithm; however, it is better than sending the password across the network in plain text. The Webulator/400 contains a service program (WWWVAUTSRV) that checks the password and user profile passed from the browser to ensure that they are valid for your AS/400 system. This

service program adopts QSYS authority to be able to call AS/400 system security APIs. If you choose to change this service program to no longer adopt authority, you should do the following steps to have the same Webulator/400 functions with regards to the user authentication sign-on method:

1. Give the server user profile specified within the Web Server/400 configuration (default value WWWUSER) *USE authority to the following programs:
   - QSYS/QSYGETPH
   - QSYS/QSYRLSPH
2. Change the owner of the WWWVAUTSRV service program to WWWUSER and remove authority adoption by using the following AS/400 commands:
   - CHGOBJOWN OBJ(WWWSERVER/WWWVAUTSRV) OBJTYPE(*SRVPGM) NEWOWN(WWWUSER)
   - CHGSRVPGM SRVPGM(WWWSERVER/WWWVAUTSRV) USRPRF(*USER)

   In normal circumstances, this method does not show the sign-on display during session initialization. If the user profile is restricted from signing on to a virtual terminal, the sign-on display is shown with an error message explaining that the user profile is not authorized to the workstation. This occurs if the QLMTSECOFR system value is 1 (the default), and user profiles with *ALLOBJ or *SERVICE special authority are not granted private authority to the virtual terminal. You may allow the user to specify the initial program, menu, or library through a query string if you enable the AllowSignonOverride option of the sign-on method configuration entry. Please refer to Query String Options for more information and ramifications of allowing sign on values to be overridden using query string keywords.

- Sign-on display:

  This method initializes the virtual terminal and shows the sign-on display to the user through the browser. The user fills in the user profile name and password. If you are not using Commerce Server/400, these values are sent across the network as plain text.

Access to each of the Webulator URLs can be protected using the access control directives within the directory-based configuration file. However, it is worthy to note that if the access control directives are used in conjunction with the user authentication sign-on method, the user name and password must match a valid AS/400 user profile name and password. Both the access control directives and the Webulator/400 sign on use the same authentication user ID and password passed on the request from the browser.

9.2.2.2 User Profile Considerations

The security of your system may be strengthened by making some changes to the user profiles configured to run from the Webulator/400 product. Depending upon your system′s security requirements, these changes may apply system wide.

Specify ″*NONE″ for the ATNPGM user profile parameter. The ATNPGM user profile parameter specifies the attention key handling program for this user. If not properly configured, this program may allow the user to get outside the realm of the initial sign-on program specified for the user. If *SYSVAL is specified for this parameter, the attention key handling program set up for all

users on the system is available to the user running through Webulator/400. By specifying *NONE for this parameter using the CHGUSRPRF command, the attention key is disabled for this user. Another way to disable this function is within the Webulator/400 button configuration. However, by doing it through the user profile, you are disabling the attention key for the user no matter which URL the user obtains access through.

Set the user profile's limited capability parameter (LMTCPB) to *YES.

Preventing Access to the System Request Menu: Normally, a workstation user can end a request or sign off by requesting the System Request menu and selecting the appropriate option. You can prevent access to the System Request menu. When your system is shipped, public authority to the System Request Menu is *USE. The simplest way to prevent users from accessing this menu is by restricting authority to the panel group QSYS/QGMNSYSR.

To prevent specific users from seeing the System Request Menu, specify *EXCLUDE authority for those users:

```
GRTOBJAUT OBJ(QSYS/QGMNSYSR) +
          OBJTYPE(*PNLGRP)   +
          USER(USERA) AUT(*EXCLUDE)
```

To prevent most users from seeing the System Request Menu, revoke public authority and grant authority to specific users:

```
RVKOBJAUT OBJ(QSYS/QGMNSYSR) +
          OBJTYPE(*PNLGRP) +
          USER(*PUBLIC) AUT(*ALL)
GRTOBJAUT OBJ(QSYS/QGMNSYSR) +
          OBJTYPE(*PNLGRP)   +
          USER(USERA) AUT(*USE)
```

If the user has the ability to invoke the System Request (SYSREQ) menu, they have the ability to carry out requests that, from a security point of view, you may not want them to do. For example, they can sign off, which gives them a sign-on display. If you have configured this user profile name to be an auto sign-on URL, you may not want the user to have a sign-on display, which allows them to guess at user IDs and passwords. Or, they have the ability to view the system operator's messages or send messages to other users on the system (which, if nothing else, may be an annoyance).

The Webulator/400 users who are using automatic sign on or user authentication sign on require manual maintenance when they change their passwords:

- Manually change the password for the user prior to the expiration time warning period.

9.2.2.3 AS/400 Virtual Terminal Considerations

Webulator/400 uses virtual terminals to execute the HTML 5250 session. As a result, some virtual terminals need to be created for this use. Webulator/400 automatically creates these devices if system value QAUTOVRT has not yet been reached. These devices are created starting in the QPACTL01, QPACTL02, or QPACTL03 virtual control units and are named QPADEVnnnn (where nnnn is a number between 0001 and 0250).

If you have implemented an interactive subsystem policy that involves specific workstation entries for each display device, it may be necessary for you to add

these new virtual devices (or device types) as workstation job entries to be controlled by your interactive subsystem. This is done by using the ADDWSE (Add Workstation Entry) command. Additional information about AS/400 virtual terminals can be found in the *AS/400 Work Management* manual.

9.2.2.4 AS/400 Programming Considerations

First of all, you probably want to restrict users who can sign on to Webulator/400 to the applications that you have selected for their use. Set the LMTCPB parameter in the user profile to *YES. Inquiries and simple data entry that are run through a verification and authentication process are probably the best applications for global Internet access.

Automatic Sign On or Authentication Sign On: If you have chosen to perform automatic sign on or user authentication sign on for the user, you do not want them to get back to a real AS/400 sign-on display. As a result, it is best not to include an option to sign off from your menus (usually option 90) and application displays given to the public. Keep in mind that this also includes the ability to sign off through help displays, whether they were written for your application or are supplied by IBM as generic help for the AS/400 system. The inability to sign off presents us with a reasonable question:

- How do I end the application and the job when the Webulator/400 session is closed by the user?

Interestingly, the answer to that question is to issue the ″SIGNOFF″ command. This is not available to the user, but from within the application when a display error occurs, which identifies that a session has been closed. Basically, you monitor for errors whenever a display is written to the 5250 display from your program. This can be done using MONMSG in CL programs following any SNDRCVF statements. The contents of the EXEC parameter are simply the ″SIGNOFF″ command.

9.2.2.5 Other Security Tips

These are a few general items that do not fall into any other specific category. These items are recommendations that affect operating practices, application objects, and Webulator/400 operational characteristics.

TERMTIME (CHGWBLCFG) must be less than QINACTITV. Since the AS/400 system has the ability to detect and sign off (alternatively disconnect) terminal jobs that have been inactive for a specific period of time, it is possible for a Webulator/400 user to be given a sign-on display if the AS/400 inactivity timeout (system value QINACTITV) expires before the Terminal Timeout (TERMTIME parameter on the CHGWBLCFG command) value. There is also the possibility of a two-minute delay in the timing of the Webulator Terminal timeout value. As a result, it is recommended that if you practice an inactivity timeout policy, the TERMTIME parameter be set to a value at least two minutes less than the QINACTITV system value. Do not change the public authority for the Web Server/400 commands because the Web Server/400 commands change the way Webulator/400 functions and also affects Internet access controls. They are installed with public authority *EXCLUDE and can be accessed by individual users with specific authority granted.

9.3 Implementation Examples

The next few pages show three different examples of Internet and intranet scenarios. They are not necessarily viable business solutions, but are shown here to illustrate the security aspects of a Web application.

At the end of this section, there is a table of all the configuration parameters and values for the respective examples.

9.3.1 Example 1: Unsecured and Secured Transaction on Isolated Server

Our objectives in this simple example are:

- Make the company's product catalog generally available to clients over the Internet.
- Make the purchase transaction secure so that credit card information is kept confidential.
- Make sure that Internet users have access **only** to data and programs from specific directories and libraries on the AS/400 Web server.
- Protect the AS/400 Web server from malicious attacks.

9.3.2 Isolated Server

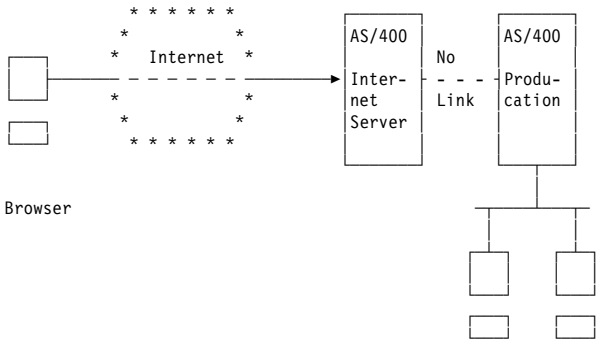


Figure 133. AS/400 System as Isolated Internet Server

This example illustrates a simple secure commercial application. We show a catalog page to the Internet community and provide a secure method for the buyer to fill out a purchase order and send a credit card number. There is no permanent link to the production AS/400 system.

9.3.2.1 Exposures

The main exposure to this solution is the compromising of data integrity and content. In other words, we do not want people to alter the content of the Web pages (for example, changing the price of the space shuttle). Because this is an isolated server, the main exposure is the information that you place on it. Avoid placing any type of sensitive information on any system that is accessed directly from the Internet.

The purchase transaction involves exchange of private data and credit card information so encryption must be used.

9.3.2.2 Sample Application Overview: Browser Windows

The following three windows show how the application looks from the browser. When a broken key is displayed in the lower left-hand corner of the window, it indicates that a document is not encrypted by the server. This ″home page″ is located in the /WWWSERV/WEBDOCS subdirectory that supports both HTTP and SSL protocol. This directive is inherited from the ROOT (/) directory.

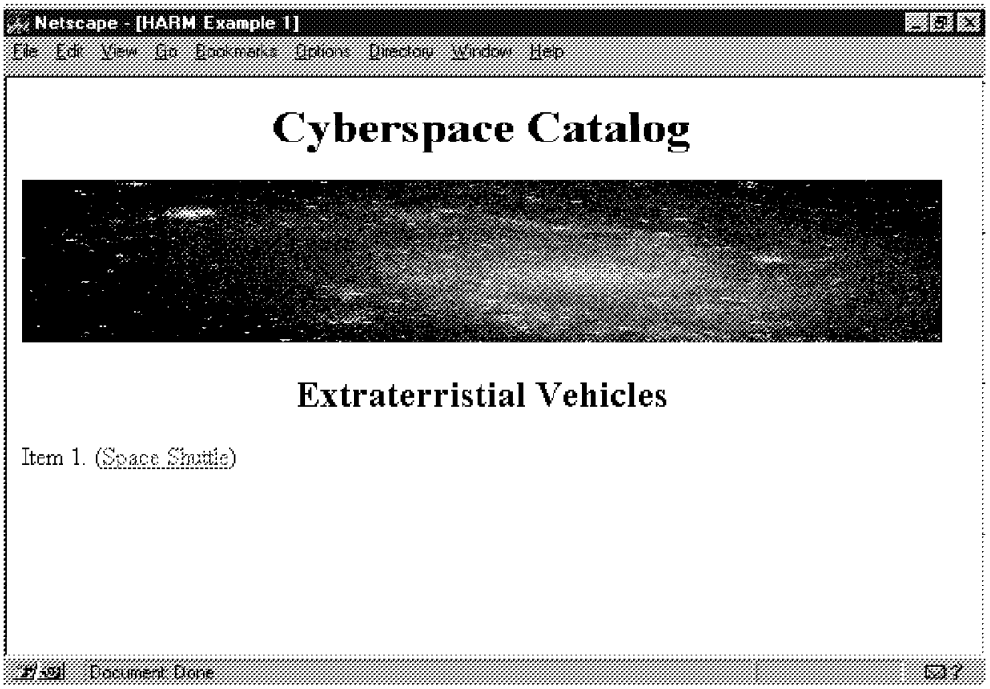


Figure 134. Catalog Window

When the user selects ″Item 1″ , the transaction is forced into SSL protocol because the ″Purchase Order″ form is located in the .../SECDEMO subdirectory, which allows only SSL protocol. The configuration parameters that limit the server to the SSL protocol for that directory are shown in Figure 149 on page 199 and Figure 150 on page 200. Also see Figure 137 on page 192 for a directory structure.

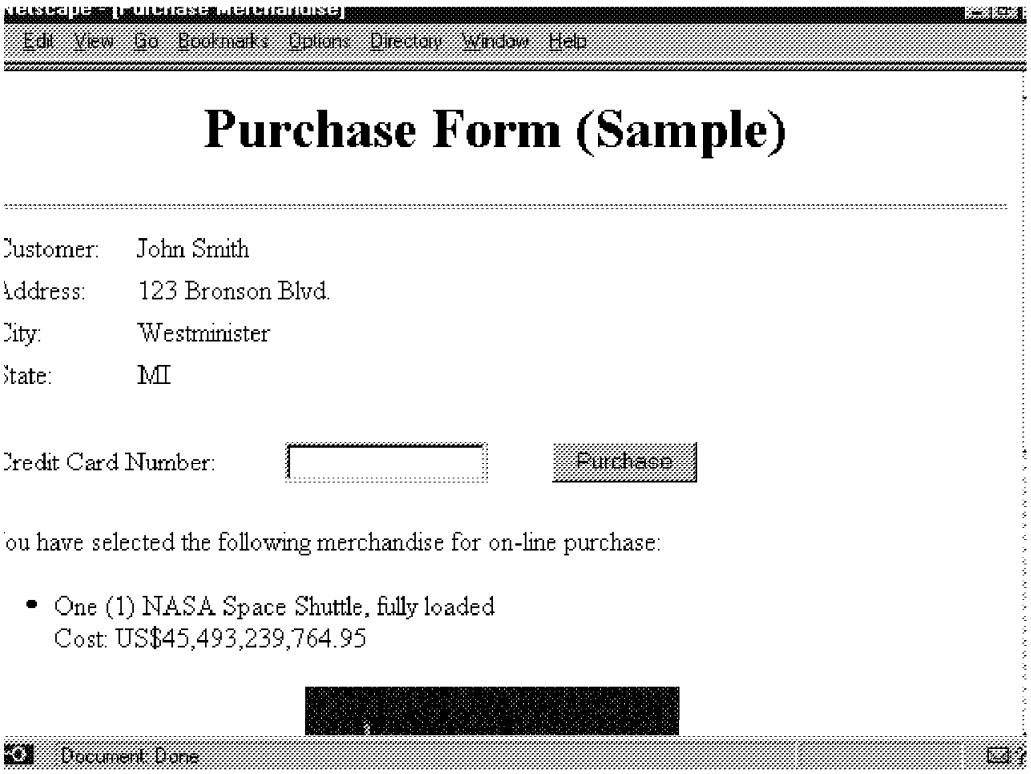


Figure 135. Purchase Order Window

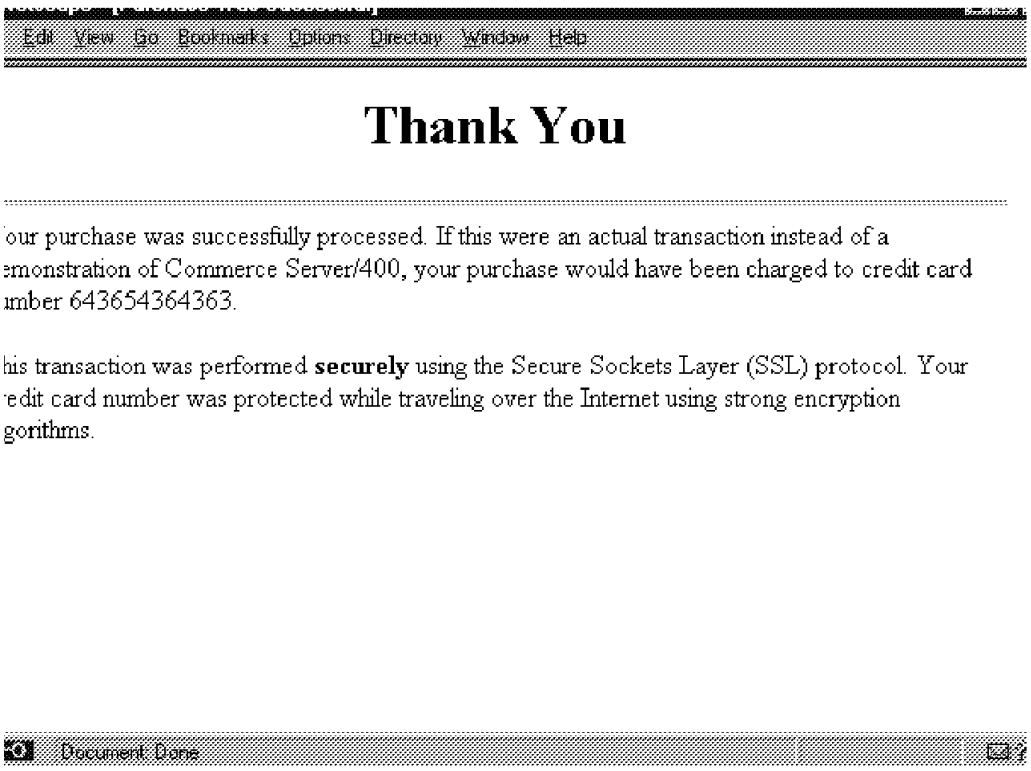


Figure 136. Confirmation Window

This configuration supports security policy in the following areas:

- Allow CGI programs to only run from the WWWCGI library.
- Allow the Web server to serve documents only from the /WWWServ/WebDocs subdirectory and its subdirectories.
- Allow the anonymous application documents to be served to anyone on the Internet with no discretionary access control.
- Protect sensitive data and ensure data integrity by using the SSL protocol support of Commerce Server/400.
- Divide the AS/400 information into two categories:
  - Not available through the Web server (cannot be obtained by direct browser GET request).
  - Available through the Web server (can be accessed by direct browser GET request).
- Disable dynamic indexing in all directories, folders, and libraries.

9.3.2.3 A Security Overview of Example 1

The following layers are utilized to enable and control access:

- At the lowest level, network design provides connectivity to the Internet and limits protocols to HTTP and SSL. The network design does not provide connectivity to the internal network.
- Transaction level security is provided by the SSL protocol.
- TCP/IP configuration provides control at the transport and network layer (for example, port restrictions).
- OS/400 configuration limits IFS access by way of the user profile of the Web server daemon and by object authority of database files and CGI programs.
- Web server configuration adds a higher layer of scope control by limiting the Web server's access to documents in only the ROOT file system and by specifying the entry point into the ROOT file system.

Figure 137 on page 192 gives an overview of the Commerce Server/400 scope control. Commerce Server/400 scope controls refer to the configuration values that affect the scope of the information the server attempts to retrieve.

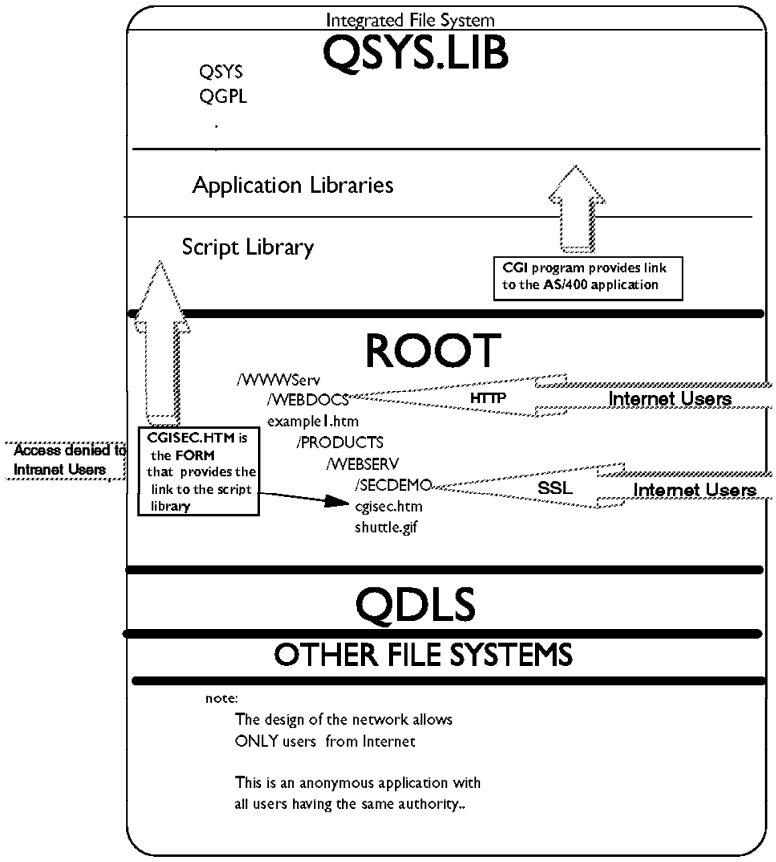


Figure 137. Commerce Server/400 Scope Control Overview

9.3.3 Commerce Server/400 Configuration

Configuration consists of setting OS/400 object authority and configuring Commerce Server/400 scope control and user authentication. See Table 19 on page 236 for the exact values that are used in each of the implementation examples.

The following list contains an overview of the steps that we follow in this example:

__ 1. Verify that the server user profile, WWWUSER, has the necessary authority to serve HTML pages:

- The server user profile must have Read/Execute (*RX) authority to the directory WWWServ/WebDocs and documents in it.

__ 2. Verify that the server user profile, WWWUSER, has the necessary authority to the log files to be used by Commerce Server/400.

__ 3. Verify that the server user profile has the necessary authority to CGI programs.

__ 4. Enable the use of CGI programs by setting the ENABLESCPT value.

__ 5. Specify which libraries may contain script files by using the WRKWWWSCPL command.

__ 6. Specify (include) the libraries that can contain HTML pages to be served by Commerce Server/400:

- Use the Work WWW Include Library (WRKWWWINCL) command to specify libraries that can be accessed from a URL that points into the QSYS file system.

__ 7. Use the Change WWW Configuration (CHGWWWCFG) command to set the following server characteristics:
- Access configuration file name
- Server root
- Document root for:
  - Root file system
  - QDLS file system
  - QSYS.LIB file system
- Index default view
- Log file locations
- Default source type
- Supported protocols
- Webulator status

__ 8. Use the Change WWW Security (CHGWWWSEC) command to define:
- Key list path and file name
- SSL port number

9.3.3.1 OS/400 Object Authority

To support our security policy, which does not allow all objects to be available to all users, we must set system security level (QSECURITY) to at least 30. We strongly recommend at least level 40. Our configuration was developed and tested on a V3R7 system at level 50.

Before beginning specific object authority configuration, please refer to Chapter 2, “Start Here by Securing OS/400®” on page 37 for information on completing a security review of the system.

The server user profile (WWWUSER) should not be able to sign on to the system and, thus, have a password value of *NONE.

The WWWServ/WebDocs directory and all objects in the directory hierarchy that are accessed through the Web server have OS/400 object authority configured as shown in Figure 138 on page 194.

```
                            Work with Authority

Object . . . . . . . . . . . . :   /WWWServ/WebDocs/example1.htm
Owner  . . . . . . . . . . . . :   WWWUSER
Primary group  . . . . . . . . :   *NONE
Authorization list . . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                   Data     --Object Authorities--
Opt  User        Authority  Exist  Mgt  Alter  Ref

     *PUBLIC     *RX
     WEBMASTER   *RWX         x     x     x     x
     WEBUSER     *RWX         x     x     x     x

Parameters or command
===>
F3=Exit   F4=Prompt   F5=Refresh      F9=Retrieve
F11=Display detail data authorities   F12=Cancel   F24=More keys
(C) COPYRIGHT IBM CORP. 1980, 1996.
```

Figure 138. Authority for AS/400 Objects Available through the Web Server

Authority to the log files subdirectory is configured as shown in Figure 139.

```
                            Work with Authority

Object . . . . . . . . . . . . :   /WWWServ/Logs
Owner  . . . . . . . . . . . . :   WEBMASTER
Primary group  . . . . . . . . :   *NONE
Authorization list . . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                   Data     --Object Authorities--
Opt  User        Authority  Exist  Mgt  Alter  Ref

     *PUBLIC      *EXCLUDE
     WEBMASTER    *RWX         X     X     X     X
     WWWUSER      *RWX         X     X     X     X
```

Figure 139. Authority for Commerce Server/400 Logs

The server user profile, WWWUSER, needs access to the library where the CGI programs reside and to the programs themselves. You have two options:

- Give *PUBLIC the necessary authority to access the library and run the CGI programs.

  In this case, you do not need to give WWWUSER private authorities to the libraries and programs. This approach has the exposure that a malicious local user can locally run the CGI program since *PUBLIC has the necessary authorizations.

- Revoke *PUBLIC authorities and **only** give the WWWUSER the necessary authorities to access and run the CGI programs. This is probably the best solution.

In this example, we chose the first option. We assume that we only have the IBM default user profiles and WEBMASTER in the isolated server so we do not need to worry too much about malicious local users. Figure 140 on page 195 shows how we set up authority for the CGI library and programs.

```
                          Edit Object Authority

Object . . . . . :   WWWCGI          Owner  . . . . . :  WWWUSER
  Library  . . . . :     QSYS          Primary group  . . . :   *NONE
Object type  . . . . :   *LIB

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . .   *NONE

                        Object    ---------------Data---------------
User        Group       Authority  Read  Add  Update  Delete  Execute
*PUBLIC                 *USE        X                            X
WEBMASTER               *ALL        X     X     X       X        X
```

Figure 140. Authority for CGI Library and Programs

Remember

To protect information from being accessed through Commerce Server/400, you must ensure that the server user profile, WWWUSER, does not have access to it. You can do this by either specifically excluding the server user profile or by excluding *PUBLIC.

9.3.3.2 Commerce Server/400 Scope Control

The features of Commerce Server/400 are used to provide another layer of control over the scope of information accessible through the Web server.

The Commerce Server/400 scope control refers to configuration values that affect the scope of the information the server attempts to retrieve.

The *IncludeLibraries* parameter in the master configuration file is used to control the scope of documents that may be accessed by the Web server (Script libraries). Only libraries included in the Include Libraries list can be accessed by a URL that points into the QSYS file system.

The WRKWWWINCL command is used to set this value. For our example, we configure no libraries available for serving Web documents as shown in Figure 141 on page 196.

```
                        Work with WWW Include Library
                                                        System:   SYSTEM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Configuration file path:   /WWWSERV/CFG/WEBSERV.CFG



Type options, press Enter.
  1=Add   2=Change   3=Add same as   4=Remove   5=Display

Opt  Libraries
 _   ____________________________________________________________
```

Figure 141. No HTML Documents Served from AS/400 Libraries

In this example, we use a CGI program to present and process an input form. To enable the server to use CGI programs, use the Change WWW Configuration (CHGWWWCFG) command to set the Enable scripts (ENABLESCPT) value to *INSIDESCRIPLIB. This allows only scripts found inside a library specified by the *ScriptLibraries* parameter to be executed.

```
                    Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Maximum request processors . . . MAXRPS         50
Request wait timeout . . . . . . TIMEOUT        120
Wait threshold . . . . . . . . . THRESHOLD      5
Enable scripts . . . . . . . . . ENABLESCPT     *INSIDESCRIPTLIB
Content CCSID  . . . . . . . . . CNTNTCCSID     819
Default source type  . . . . . . DEFSRCTYPE     *ROOT
Disable server:                  DISABLESVR
                                                *NO
  date available . . . . . . . .                000000
  time available . . . . . . . .                000000
Index name . . . . . . . . . . . IDXNAME        'example1.htm'

Send file content length . . . . SENDFILLEN     *NO
```

Figure 142. Allow CGI Programs to be Accessed

In conjunction with the enable scripts parameter, use the Work with WWW Script Libraries (WRKWWWSCPL) command to specify exactly in which libraries the HTTP server may access CGI programs. For this example, we have only one library, which is called WWWCGI, as shown in Figure 143 on page 197.

```
                          Work with WWW Script Libraries
                                                              System:   HARM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Configuration file path:   /WWWSERV/CFG/WEBSERV.CFG



Type options, press Enter.
 1=Add   2=Change   3=Add same as   4=Remove   5=Display

Opt  Libraries
 _   __________________________________________________
 _   WWWCGI


 F3=Exit   F5=Refresh   F6=Print   F12=Cancel
```

Figure 143. Libraries Containing CGI Programs

Commerce Server/400 uses the NCSA server model directives for controlling the scope of documents that are available to the Web server. Access to each file system in the IFS is allowed and controlled by parameters in the master configuration file. If a value of *EXCLUDE is given for the document root of a file system, the server may not access documents in that file system. If a value is given, it defines the entry point into that file system.

In this example, only documents in the ROOT file system are accessible to the server. Documents in the ROOT file system that are accessible to the Web server are defined by two values in the master configuration file:

1. Server root path (SVRROOT)
2. Document root path (DOCROOT)

No documents that reside above this point in the hierarchy can be accessed.

For this example, we use the CHGWWWCFG command as illustrated in Figure 144.

```
                      Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Connection queue size  . . . . . CONNQUESIZ     64
Initial request processors . . . INITRPS        1
Maximum request processors . . . MAXRPS         50
Request wait timeout . . . . . . TIMEOUT        120
Wait threshold . . . . . . . . . THRESHOLD      5
Enable scripts . . . . . . . . . ENABLESCPT     *INSIDESCRIPTLIB
Content CCSID  . . . . . . . . . CNTNTCCSID     819
Default source type  . . . . . . DEFSRCTYPE     *ROOT
Disable server:                  DISABLESVR
```

Figure 144. Set Default Source File System

Note: Documents are not allowed to be served from either QDLS or QSYS file systems.

```
                    Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.


 Document root path . . . . . . . DOCROOT        'WEBDOCS'

 Document root for QDLS . . . . . DOCROOTQ     > *NONE

 Document root for QSYS . . . . . DOCROOTSYS     *NONE
 Server root path . . . . . . . . SVRROOT        '/WWWSERV'
```

Figure 145. Allow Only Documents in Root File System Served by Commerce Server/400

9.3.3.3 Commerce Server/400 Access Control

The next step is to enable SSL for the server. Please refer to Section 1.9.6, "What is Secured Sockets Layer (SSL)?" on page 31 for more information on SSL protocol.

Two parameters need to be set to enable the SSL protocol.

1. Use the CHGWWWCFG command to set the PROTOCOLS parameter in the master configuration file as shown in Figure 146. In our example, this allows the server to use both protocols.

```
                    Change WWW Configuration (CHGWWWCFG)

 Type choices, press Enter.

 Public user directory  . . . . . PUBUSERDIR     'PUBHTML'

 Temporary directory  . . . . . . TEMPDIR        'TMP'

 Initial library list . . . . . . INLLIBL        *CURRENT
 Password storage . . . . . . . . PSWDSTG        *COMPATIBLE
 Server protocols . . . . . . . . PROTOCOLS      *HTTP
                                                 *SSL
 Webulator user file path . . . . WBLUSRFILE     *NONE

 Maximum Webulator sessions . . . WBLMAXSSN      20
 Disable Webulator:               DISABLEWBL
                                                 *NO
   date available . . . . . . . .                000000
   time available . . . . . . . .                000000
 Update executing RPs . . . . . . UPDATE         *DEFER
```

Figure 146. Change WWW Configuration

2. The ALLOWEDPROTOCOLS parameter in the access configuration file is configured as shown by the following two figures. Figure 147 on page 199 and Figure 148 on page 199 illustrate setting the ROOT directory to allow both SSL and HTTP protocols. This attribute is inherited by all lower subdirectories unless explicitly altered.

```
*STANDARD           Work with WWW Directory Configurations
                                                          System: HARM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory configuration:   /WWWSERV/cfg/access.cfg



Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   6=Work with limits
  8=Work with parsed buttons    9=Work Virtual Keyboard   10=Change Webulator
  14=Change Commerce Server/400

Opt  Directory

14   /
     /WWWSERV/WEBDOCS/PRODUCTS/WEBSERV/SECDEMO/
```

Figure 147. Work with WWW Directory Configuration

```
                            Set Allowed Protocols

 Directory . . . . . :   /



 Select (/) choices and press Enter.  By making no selection, values from
   previous level will be inherited.

 Allowed protocols . . .   /  HTTP
                           /  SSL

 Update executing RPs      *IMMED       *DEFER, *IMMED
```

Figure 148. Set Allowed Protocols

For our example, we next configure the .../secdemo subdirectory to allow only the SSL protocol as illustrated in Figure 149 and Figure 150 on page 200.

```
*STANDARD           Work with WWW Directory Configurations
                                                    System:   SYSTEM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory configuration:   /WWWSERV/cfg/access.cfg



Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   6=Work with limits
  8=Work with parsed buttons    9=Work Virtual Keyboard   10=Change Webulator
  14=Change Commerce Server/400

Opt  Directory

     /
14   /WWWSERV/WEBDOCS/PRODUCTS/WEBSERV/SECDEMO/
```

Figure 149. Work with WWW Directory Configurations

```
                          Set Allowed Protocols

Directory . . . . . :   /WWWSERV/WEBDOCS/PRODUCTS/WEBSERV/SECDEMO/



Select (/) choices and press Enter.  By making no selection, values from
  previous level will be inherited.

Allowed protocols . . .      HTTP
                         /   SSL

Update executing RPs     *IMMED       *DEFER, *IMMED
```

Figure 150. Set Allowed Protocols for SECDEMO Subdirectory

The next value to be set is the state of the Webulator. The shipped default is to not start the Webulator. We leave the value as shown in Figure 151.

```
                    Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Master Configuration File  . . . CFGFILE        '/WWWSERV/CFG/WEBSERV.CF

                                                *SSL
Webulator user file path . . . . WBLUSRFILE     *NONE

Maximum Webulator sessions . . . WBLMAXSSN      20
Disable Webulator:               DISABLEWBL
                                              > *YES
```

Figure 151. Set Webulator to Not Start

9.3.3.4 Additional Configuration Values

The following section and sample composite displays illustrate the setting of several additional configuration values.

- The *Index Default View* value enables or disables the creation of dynamic indexes and sets their default appearance. Set the value to *NONE to prevent browser clients from listing files in the directory.
- The *ACCLOGFILE* value specifies the path and name of the access log that records all attempts to access the server. If this is blank (a command value of *NONE), access logging is turned off. We recommend that you provide a name for the access log. Access logging does create overhead for the CPU so you may want to turn it off if security is less of a concern than performance.
- The *ERRLOGFILE* value specifies the path and name of the error log file that records all server errors. If this is blank (a command value of *NONE), error logging is turned off. We recommend that a name be provided for the error log.
- The *STTLOGFILE* value specifies the path and name of the statistical log. The statistics log is a record of operational events and, from a security point of view, can be analyzed for unusual patterns or trends. If this is blank (a command value of *NONE), statistics logging is turned off. We recommend that a name be provided for the statistical log.
- The *Default Source Type* value indicates the file system to use when any of the following cases occurs:
  - A URL has no alias.

- An alias does not have an explicit source type.
- The server's home page is requested ('/').

This value does not have explicit security implications but must be set to either *ROOT, QSYS, or QDLS. Since we are setting our scope of control to specify only documents in the root file's system, we set this value to *ROOT.

These values are set using the CHGWWWCFG command, which, in this example, is run against the master configuration file, WEBSERV.CFG.

```
                          Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Master Configuration File  . . . CFGFILE        '/WWWSERV/CFG/WEBSERV.CFG

Index default view:              IDXDFTVIEW
                                                 *NONE
                                                 *OFF

Access log file path:            ACCLOGFILE
                                                 'LOGS/ACCESS.LOG'

Error log file path:             ERRLOGFILE
                                                 'LOGS/ERROR.LOG'

Statistics log file path:        STTLOGFILE
                                                 'LOGS/STATS.LOG'

Default source type  . . . . . . DEFSRCTYPE     *ROOT


Update executing RPs . . . . . . UPDATE         *DEFER


F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 152. Set Multiple Configuration Values

The /WWWServ/Cfg/webserv.cfg file is shown in Figure 153 on page 202. Please note that not all master configuration values are included in the file. Some default values are assumed by the server daemon and are not written to this file. The CHGWWWCFG, CHGWWWSEC, WRKWWWSCPL, and WRKWWWINCL commands must be used to verify values.

```
;
; Web Server/400 master configuration file
;
DOCUMENTROOTQDLS

GlobalAdminAccessCfgFile Cfg/AdAccess.cfg

INDEXDEFAULTVIEW  NONE    OFF

INDEXNAME  example1.htm

IndexStyle IncludeAll IncludeHTMLTitles

ScriptLibraries WWWCGI

ServerSideInclude AllowExec
DOMAINNAMELOOKUP  MINIMAL
COMMERCEKEYLISTFILE  /wwwserv/key/keylist.cfg
SERVERPROTOCOLS  HTTP SSL
GLOBALACCESSCFGFILE  cfg/access.cfg
DISABLEWEBULATOR  YES
```

Figure 153. Webserv.cfg

The access configuration file contains directory-based configuration values. Some defaults are assumed if this file is empty or does not exist. Directives are inherited by subdirectories unless explicitly altered. For example, the /WWWSERV/WEBDOCS subdirectory allows both protocols since it inherits that property from the root directory. The .../secdemo allows only SSL because of the explicit directory entry.

```
; Commerce Server/400 access configuration file
; for Example 1 in the HARM book

<DIRECTORY />
   ALLOWEDPROTOCOLS HTTP SSL
</DIRECTORY>
<DIRECTORY /wwwserv/webdocs/products/webserv/secdemo>
   ALLOWEDPROTOCOLS SSL
</DIRECTORY>
```

Figure 154. Access.cfg

9.3.4 Example 2: Intranet/INTERNET

This example illustrates a Web server solution that allows access to both intranet and Internet users. This network design is referred to as the Integrated Server design shown in Figure 155.

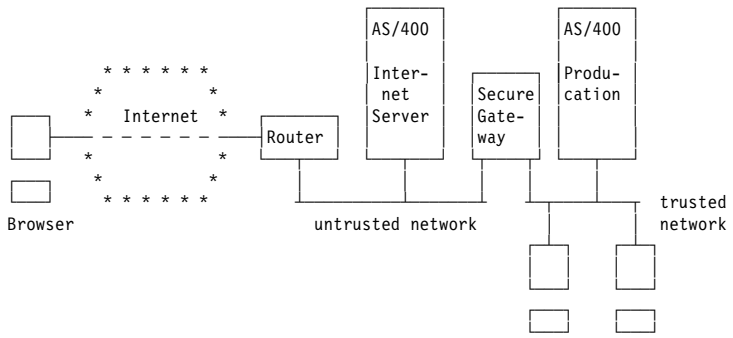


Figure 155. AS/400 System as Integrated Internet Server

9.3.4.1 Exposures

The exposures in this example are:

- Internet users gaining access to the private Web server.
- IDs and passwords on the private network being ″sniffed″.
- Web page content being altered, which damages reputation.

9.3.4.2 Security Policy

In this example, we support our security policy in the following areas:

- Allow the Web server to serve documents from only the directory HARM and its subdirectories.
- Allow documents in the HARM/WEBDOCS/PUBLIC subdirectory to be served anonymously on the Internet.
- Allow documents under the HARM/WEBDOCS/PRIVATE subdirectory to be accessed by all corporate users, but **only** from the intranet.
- Allow documents under the HARM/WEBDOCS/PRIVATE/RESTRICT subdirectory to be accessed by **some** corporate users on the intranet.
- Do not allow User IDs or passwords to be transmitted in the clear across the Internet or intranet.
- Classify the AS/400 information as follows:
  - Not available through the Web Server
  - Available to the public users
  - Available to all private users
  - Available to some private users
- Classify the users as follows:
  - Private users with Web browsers
  - Privileged private users with Web browsers
  - Public users with Web browsers

9.3.4.3 Security Overview of Example 2

The following layers are utilized to provide and control access:

- At the lowest level, network design provides connectivity and limits protocols to HTTP and SSL. The network design provides connectivity to the intranet as well as the Internet.
- Transaction level security is provided by the SSL protocol.
- TCP/IP configuration provides control at the transport and network layer (for example, port restrictions).
- OS/400 configuration limits IFS access by way of the user profile of the Web server daemon and by object authority of database files and CGI programs.
- Web server configuration adds a higher layer of scope control by limiting the Web server′s access to documents in only the ROOT file system and by specifying entry points into the ROOT file system. The Web server configuration also provides discretionary access control by utilizing user authentication. In this example, two instances of Commerce Server/400 are configured:
  - The Public server, which is accessible from the Internet or intranet. This server allows only HTTP protocol. The entry point for browser access is

at the /HARM/WEBDOCS/PUBLIC directory level. No documents above this point in the hierarchy are accessible through the Public Web server.

– The Private server, which is accessible from the intranet. This server allows only SSL protocol. The entry point for browser access is at the /HARM/WEBDOCS/PRIVATE directory level. No documents above this point in the hierarchy are accessible through the Private Web Server.

9.3.4.4 Browser Windows for the Public Web Site

The Public Web server example has only one page. The broken key in the lower left corner indicates that HTTP protocol is being used.

Figure 156. Public Web Site

The following illustration shows the relationship between user sets and data sets in IFS.

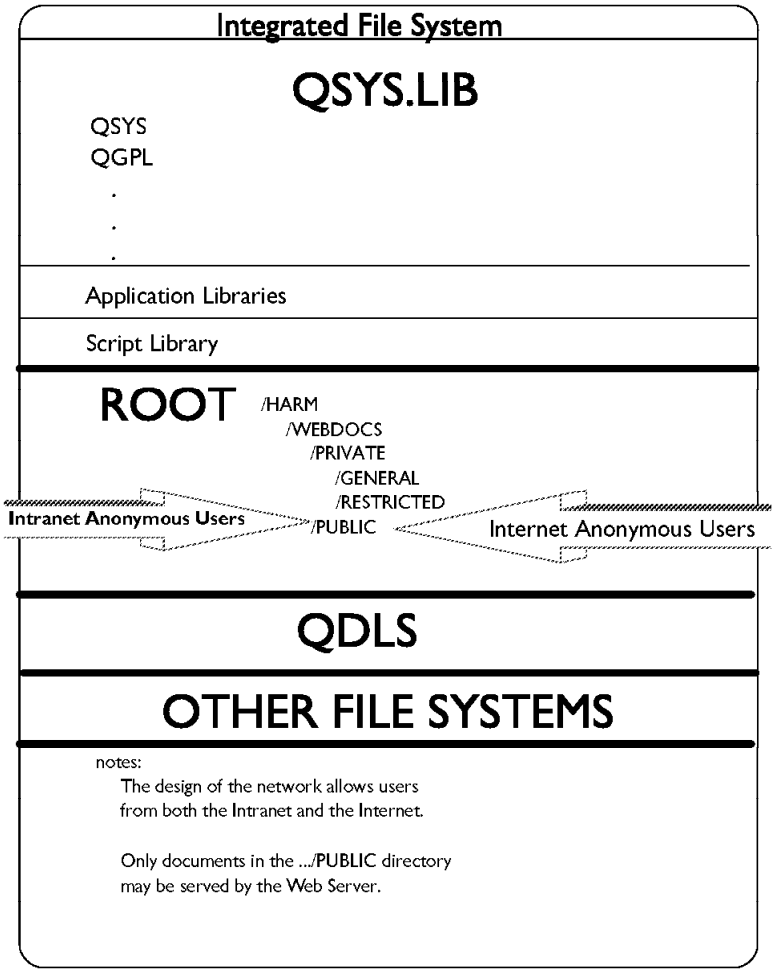


Figure 157. Example 2-2 IFS

9.3.4.5 Public Web Server Configuration

Configuration consists of setting OS/400 object authority, configuring Commerce Server/400 scope control, and user authentication. See Table 19 on page 236 for the exact values that are used in each of the implementation examples.

The following steps are taken for this example:

__ 1. Verify that the server user profile WWWUSER has the necessary authority to serve HTML pages. The server user profile must have Read/Execute (*RX) authority to the directory HARM/WEBDOCS/PUBLIC and the documents in it.
__ 2. Verify that the server user profile, WWWUSER, has the necessary authority to the log files to be used by Commerce Server/400.
__ 3. Disable the use of CGI scripts by setting the *ENABLESCPT* parameter to *None.
__ 4. Specify (include) the libraries that can contain HTML pages to be served by Commerce Server/400. Use the WRKWWWINCL command to specify libraries that may contain Web documents.
__ 5. Use the CHGWWWCFG command to set the following server characteristics:

- Server root
- Document root for:
  - Root file system
  - QDLS file system
  - QSYS.LIB file system
- Index default view
- Log file locations
- Default source type
- Webulator status

In this example, the portion of the directory structure in the ROOT file system that contains our Web pages is illustrated in Figure 157 on page 205

OS/400 Object Authority: To support our security policy, which does not allow all objects to be available to all users, we must set system security level (QSECURITY) to at least 30. We strongly recommend at least level 40. Our configuration was developed and tested on a V3R7 system at level 50.

Before beginning specific object authority configuration, please refer to Chapter 2, “Start Here by Securing OS/400®” on page 37 for information on completing a security review of the system. The server user profile should not be able to sign on to the system and, thus, have a password value of *NONE.

The HARM/WEBDOCS/PUBLIC directory and all objects in the directory hierarchy that are accessed through the Web server have OS/400 object authority configured as illustrated in Figure 158.

```
                              Work with Authority

Object . . . . . . . . . . . . :   /HARM/WEBDOCS/PUBLIC/pubhome.htm
Owner  . . . . . . . . . . . . :   WWWUSER
Primary group  . . . . . . . . :   *NONE
Authorization list . . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                    Data     --Object Authorities--
Opt  User        Authority  Exist  Mgt  Alter  Ref

     *PUBLIC     *RX
     WEBMASTER   *RWX         x      x     x     x

Parameters or command
===>
F3=Exit   F4=Prompt   F5=Refresh      F9=Retrieve
F11=Display detail data authorities   F12=Cancel   F24=More keys
(C) COPYRIGHT IBM CORP. 1980, 1996.
```

Figure 158. Authority for AS/400 Objects Available to Web Server

Authority to the log files subdirectory is configured as shown in Figure 159 on page 207.

```
                              Work with Authority

Object . . . . . . . . . . . . :   /HARM/LOGS
Owner  . . . . . . . . . . . . :   WEBMASTER
Primary group  . . . . . . . . :   *NONE
Authorization list . . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                   Data     --Object Authorities--
Opt  User        Authority  Exist  Mgt  Alter  Ref

     *PUBLIC      *EXCLUDE
     WEBMASTER    *RWX         X     X     X     X
     WWWUSER      *RWX         X     X     X     X
```

Figure 159. Authority to Commerce Server/400 Server Logs

9.3.5 Commerce Server/400 Scope Control

The features of Commerce Server/400 are used to provide another layer of control over the scope of information accessible through the Web server.

In this example, we do not use any CGI programs. To prevent the server from using CGI programs, use the Change WWW Configuration (CHGWWWCFG) command to set the Enable scripts (ENABLESCPT) value to *None. This prevents scripts from being run even if a request is made to a valid program.

```
                     Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Maximum request processors . . . MAXRPS         50
Request wait timeout . . . . . . TIMEOUT        120
Wait threshold . . . . . . . . . THRESHOLD      5
Enable scripts . . . . . . . . . ENABLESCPT     *None
Content CCSID  . . . . . . . . . CNTNTCCSID     819
Default source type  . . . . . . DEFSRCTYPE     *ROOT
Disable server:                  DISABLESVR
                                                *NO
  date available . . . . . . . .                000000
  time available . . . . . . . .                000000
Index name . . . . . . . . . . . IDXNAME        'example1.htm'

Send file content length . . . . SENDFILLEN     *NO
```

Figure 160. Prevent Execution of CGI Programs

The *IncludeLibraries* parameter in the master configuration file is used to control the scope of documents that may be accessed by the Web server. Only libraries included in the Include Libraries list can be accessed by a URL that points into the QSYS file system.

The WRKWWWINCL command is used to set this value. For our example, we configure no libraries available for serving Web documents as shown in Figure 161 on page 208.

```
                        Work with WWW Include Library
                                                    System:   SYSTEM
  Update executing RPs . .   *DEFER     *DEFER, *IMMED
  Configuration file path:   /HARM/CFG/PUBSERV.CFG



  Type options, press Enter.
    1=Add   2=Change   3=Add same as   4=Remove   5=Display

  Opt  Libraries
   _   ____________________________________________________________
```

Figure 161. Allow No Libraries to Contain Web Documents

Commerce Server/400 uses the NCSA server model directives for controlling the scope of documents that are available to the Web server. Access to each file system in the IFS is allowed and controlled by parameters in the master configuration. If a value of *NONE is given for the document root of a file system, the server may not access documents in that file system. If a value is given, it defines the entry point into that file system. In this example, only documents in the ROOT file system are accessible to the server. Documents in the ROOT file system that are accessible to the Web server are defined by two values in the master configuration file:

1. Server root path (SVRROOT)
2. Document root path (DOCROOT)

No documents that reside above this point in the hierarchy can be accessed. For this example, use the CHGWWWCFG command as illustrated in Figure 162.

```
                      Change WWW Configuration (CHGWWWCFG)

  Type choices, press Enter.

  Connection queue size  . . . . . CONNQUESIZ     64
  Initial request processors . . . INITRPS        1
  Maximum request processors . . . MAXRPS         50
  Request wait timeout . . . . . . TIMEOUT        120
  Wait threshold . . . . . . . . . THRESHOLD      5
  Enable scripts . . . . . . . . . ENABLESCPT     *INSIDESCRIPTLIB
  Content CCSID  . . . . . . . . . CNTNTCCSID     819
  Default source type  . . . . . . DEFSRCTYPE     *ROOT
  Disable server:                  DISABLESVR
```

Figure 162. Set Default Source File System

```
                      Change WWW Configuration (CHGWWWCFG)

 Type choices, press Enter.


  Document root path . . . . . . . DOCROOT        'WEBDOCS/PUBLIC'

  Document root for QDLS . . . . . DOCROOTQ     > *NONE

  Document root for QSYS . . . . . DOCROOTSYS     *NONE
  Server root path . . . . . . . . SVRROOT        '/HARM'
```

Figure 163. Allow Only ROOT Web Documents

Note that documents are not allowed to be served from QDLS or QSYS.

9.3.5.1 Additional Configuration Values

The following section and sample composite displays illustrate setting several additional configuration values.

- The *Index Default View* value enables or disables the creation of dynamic indexes and sets their default appearance. Set the value to *NONE to prevent browser clients from listing files in the directory.
- The *ACCLOGFILE* value specifies the path and name of the access log that records all attempts to access the server. If this is blank (a command value of *NONE), access logging is turned off. We recommend that you provide a name for the access log. Access logging does create overhead for the CPU so you may want to turn it off if security is less of a concern than performance.
- The *ERRLOGFILE* value specifies the path and name of the error log file that records all server errors. If this is blank (a command value of *NONE), error logging is turned off. We recommend that a name be provided for the error log.
- The *STTLOGFILE* value specifies the path and name of the statistical log. The statistics log is a record of operational events and, from a security point of view, can be analyzed for unusual patterns or trends. If this is blank (a command value of *NONE), statistics logging is turned off. We recommend that a name be provided for the statistical log.
- The *Default Source Type* value indicates the file system to use when any of the following cases occurs:
  - A URL has no alias.
  - An alias does not have an explicit source type.
  - The server's home page is requested ('/').

  This value does not have explicit security implications, but must be set to either *ROOT, QSYS, or QDLS. Since we are setting our scope of control to specify only documents in the root files system that are accessed, we set this value to *ROOT.

These values are set using the CHGWWWCFG command, which, in this example, is run against the master configuration file, WEBSERV.CFG.

```
                    Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Master Configuration File  . . . CFGFILE        '/WWWSERV/CFG/WEBSERV.CFG

Index default view:              IDXDFTVIEW
                                              *NONE
                                              *OFF

Access log file path:            ACCLOGFILE
                                              'LOGS/ACCESS.LOG'

Error log file path:             ERRLOGFILE
                                              'LOGS/ERROR.LOG'

Statistics log file path:        STTLOGFILE
                                             'LOGS/STATS.LOG'

Default source type  . . . . . . DEFSRCTYPE     *ROOT


Update executing RPs . . . . . . UPDATE         *DEFER


F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 164. Set Multiple Configuration Values

The final value to check is the state of the Webulator. The shipped default is to not start the Webulator. We leave the value shown in Figure 165.

```
                    Change WWW Configuration (CHGWWWCFG)

 Type choices, press Enter.

 Master Configuration File  . . . CFGFILE        '/HARM/CFG/PUBSERV.CFG

 Webulator user file path . . . . WBLUSRFILE     *NONE

 Maximum Webulator sessions . . . WBLMAXSSN      20
 Disable Webulator:               DISABLEWBL
                                               > *YES
```

Figure 165. Set Webulator to Not Start

The /HARM/CFG/PUBSERV.CFG file is shown in Figure 166 on page 211. Please notice that not all master configuration values are included in the file. Some default values are assumed by the server daemon and are not written to this file. The CHGWWWCFG, CHGWWWSEC, and WRKWWWINCL commands must be used to verify values.

```
;
; Public Commerce Server/400 master configuration file
; for Example 2 in the HARM book
; PUBSERV.CFG

DOCUMENTROOTQDLS

GlobalAdminAccessCfgFile Cfg/AdAccess.cfg


INDEXDEFAULTVIEW  NONE    OFF

INDEXNAME  PUBHOME.HTM

IndexStyle IncludeAll IncludeHTMLTitles


ServerSideInclude AllowExec
DOMAINNAMELOOKUP  MINIMAL
SERVERPROTOCOLS  HTTP
GLOBALACCESSCFGFILE  cfg/pubacc.cfg
DISABLEWEBULATOR  YES
DOCUMENTROOT  WEBDOCS/PUBLIC
SERVERROOT  /HARM
ACCESSLOG  LOGS/PUBACC.LOG  850
ERRORLOG  LOGS/PUBERR.LOG  850
PORT  82
STATISTICSLOG  LOGS/PUBSTAT.LOG  850
```

Figure 166. Public Web Server Master Configuration File

The access configuration file contains directory-based configuration values. Some defaults are assumed if this file is empty or does not exist. Directives are inherited by subdirectories unless explicitly altered.

```
; Public Commerce Server/400 access configuration file
; for Example 2 in the HARM book

<DIRECTORY />
   ALLOWEDPROTOCOLS HTTP
</DIRECTORY>
```

Figure 167. Public Web Server Access Configuration File

9.3.5.2 Browser Windows for the Private Web Site

This example Web site consists of four pages:

1. The Home Page (Figure 168 on page 212):

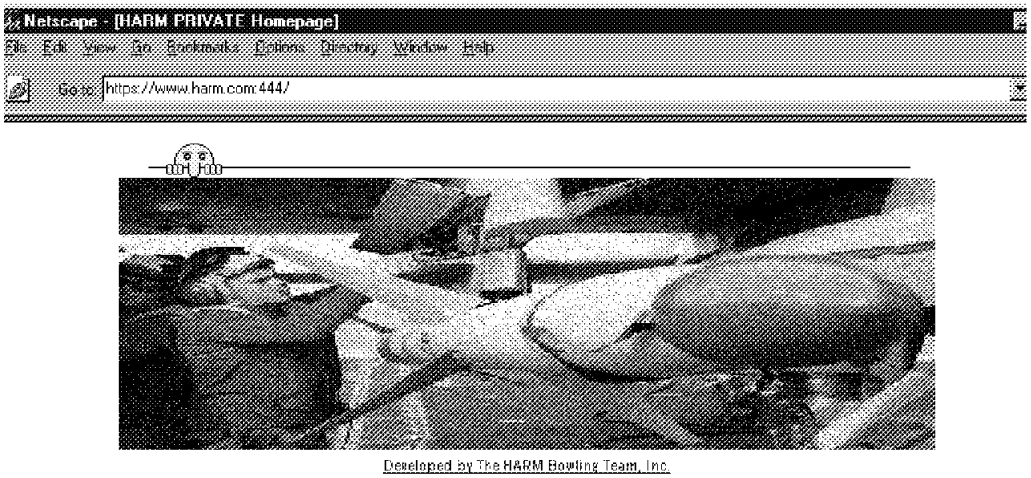


Figure 168. Home Page

2. General Information Area Home Page (Figure 169):

Figure 169. General Information Area Home Page

3. Password window to the restricted page (Figure 170 on page 213):

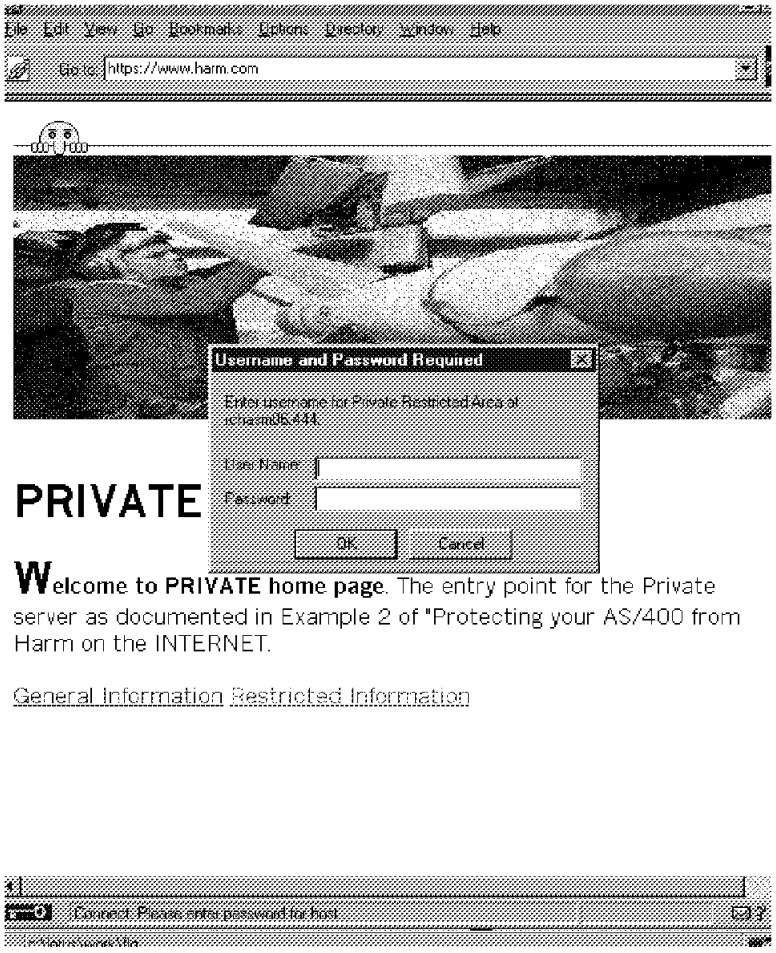


Figure 170. Password Window

4. The Private Restricted Home Page (Figure 171 on page 214):

Figure 171. Private Restricted Home Page

Figure 172 on page 215 and Figure 173 on page 216 show the relationship between user sets and data sets in the IFS.

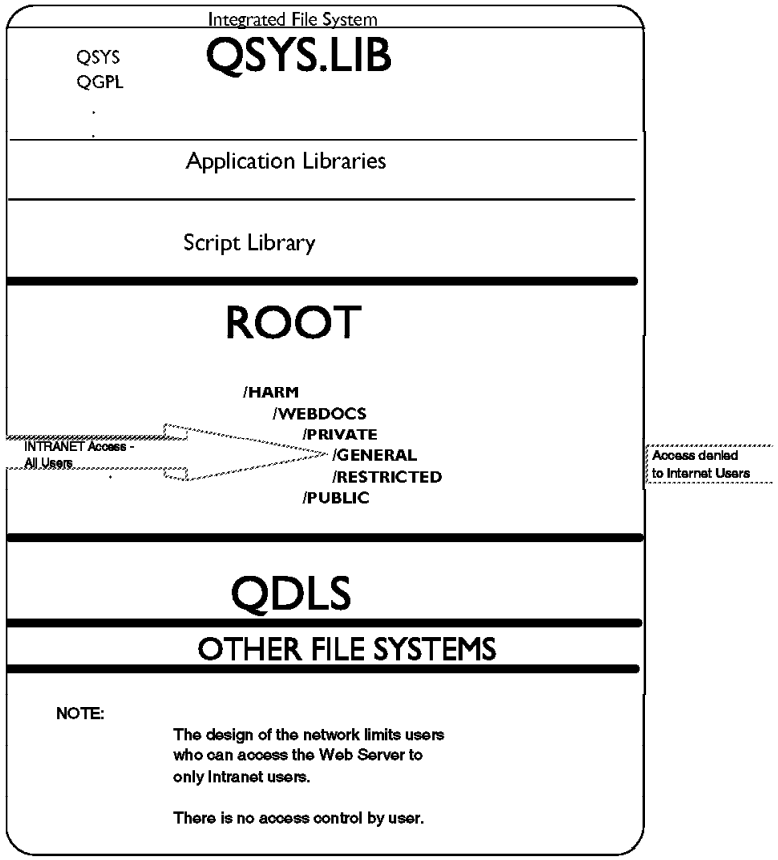


Figure 172. Private Web Site Access for General Users

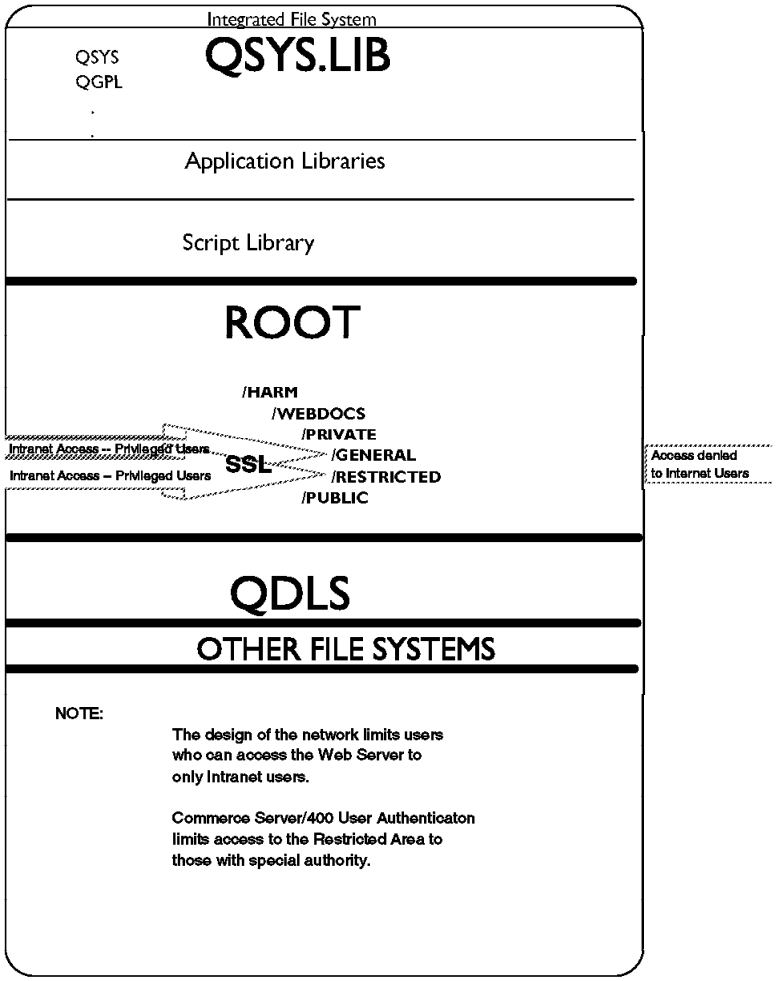


Figure 173. Private Web Site Access for Privileged Users

9.3.5.3 Private Web Server Configuration

Configuration consists of setting OS/400 object authority, configuring Commerce Server/400 scope control, and user authentication.

See Table 19 on page 236 for the exact values that are used in each of the implementation examples.

The following steps are taken for this example:

__ 1. Verify that the server user profile WWWUSER has the necessary authority to serve HTML pages. The server user profile must have Read/Execute (*RX) authority to the directory HARM/WEBDOCS/PRIVATE and the documents in it.
__ 2. Verify that the server user profile, WWWUSER, has the necessary authority to the log files to be used by Commerce Server/400.
__ 3. Disable the use of CGI scripts by setting the *ENABLESCPT* parameter to *None.
__ 4. Specify (include) the libraries that can contain HTML pages to be served by Commerce Server/400. Use the WRKWWWINCL command to specify libraries that may contain Web documents.

__ 5. Use the CHGWWWCFG command to set the following server characteristics:
   - Server root
   - Document root for:
     - Root file system
     - QDLS file system
     - QSYS.LIB file system
   - Index default view
   - Log file locations
   - Default source type
   - Supported protocols
   - Webulator status

__ 6. Use the CHGWWWSEC command to define:
   - Key list path and file name
   - SSL port number

__ 7. Use the CHGWWWDIR command to define:
   - Authorization user file path and file name
   - Authorization type
   - Authorization name

9.3.5.4 OS/400 Object Authority

To support our security policy, which does not allow all objects to be available to all users, we must set system security level (QSECURITY) to at least 30. We strongly recommend at least level 40. Our configuration was developed and tested on a V3R7 system at level 50.

Before beginning specific object authority configuration, please refer to Chapter 2, “Start Here by Securing OS/400®” on page 37 for information on completing a security review of the system. The server user profile should not be able to sign on to the system and, thus, have a password value of *NONE. The HARM/WEBDOCS/PRIVATE directory and all objects in the directory hierarchy that are accessed through the Web server have OS/400 object authority configured as illustrated in Figure 174.

```
                            Work with Authority

Object . . . . . :   /HARM/WEBDOCS/PRIVATE/pvthome.htm
Owner  . . . . . :   WEBMASTER
Primary group  . . . . . :   *NONE
Authorization list . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                   Data     --Object Authorities--
Opt  User        Authority  Exist  Mgt  Alter  Ref

     *PUBLIC     *EXCLUDE
     WWWUSER     *RX
     WEBMASTER   *RWX         X      X     X     X

Parameters or command
===>
F3=Exit   F4=Prompt   F5=Refresh      F9=Retrieve
F11=Display detail data authorities   F12=Cancel   F24=More keys
(C) COPYRIGHT IBM CORP. 1980, 1996.
```

Figure 174. Authority for Objects Available to Web Server

Authority to the log files subdirectory is configured as shown in Figure 175 on page 218.

```
                              Work with Authority

Object . . . . . . . . . . . . :   /HARM/LOGS
Owner  . . . . . . . . . . . . :   WEBMASTER
Primary group  . . . . . . . . :   *NONE
Authorization list . . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                   Data     --Object Authorities--
Opt  User        Authority  Exist  Mgt  Alter  Ref

     *PUBLIC      *EXCLUDE
     WEBMASTER    *RWX         X     X     X     X
     WWWUSER      *RWX
```

Figure 175. Work with Authority

9.3.5.5 Commerce Server/400 Scope Control

The features of Commerce Server/400 are used to provide another layer of control over the scope of information accessible through the Web server.

In this example, we do not use any CGI programs.

To prevent the server from using CGI programs, use the Change WWW Configuration (CHGWWWCFG) command to set the Enable scripts (ENABLESCPT) value to *None. This prevents scripts from being run even if a request is made to a valid program.

```
                     Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Maximum request processors . . . MAXRPS         50
Request wait timeout . . . . . . TIMEOUT        120
Wait threshold . . . . . . . . . THRESHOLD      5
Enable scripts . . . . . . . . . ENABLESCPT     *None
Content CCSID  . . . . . . . . . CNTNTCCSID     819
Default source type  . . . . . . DEFSRCTYPE     *ROOT
Disable server:                  DISABLESVR
                                                *NO
  date available . . . . . . . .                000000
  time available . . . . . . . .                000000
Index name . . . . . . . . . . . IDXNAME        'example1.htm'

Send file content length . . . . SENDFILLEN     *NO
```

Figure 176. Prevent Execution of CGI Programs

The *IncludeLibraries* parameter in the master configuration file is used to control the scope of documents that may be accessed by the Web server. Only libraries included in the Include Libraries list can be accessed by a URL that points into the QSYS file system.

The WRKWWWINCL command is used to set this value. For our example, we configure no libraries available for serving Web documents as shown in Figure 177 on page 219.

```
                          Work with WWW Include Library
                                                  System:   SYSTEM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Configuration file path:   /HARM/CFG/PVTSERV.CFG



Type options, press Enter.
  1=Add   2=Change   3=Add same as   4=Remove   5=Display

Opt  Libraries
 _   ____________________________________________________________
```

Figure 177. Allow No Libraries to Contain Web Documents

Commerce Server/400 uses the NCSA server model directives for controlling the scope of documents that are available to the Web server. Access to each file system in the IFS is allowed and controlled by parameters in the master configuration. If a value of *NONE is given for the document root of a file system, the server may not access documents in that file system. If a value is given, it defines the entry point into that file system. In this example, only documents in the ROOT file system are accessible to the server. Documents in the ROOT file system that are accessible to the Web server are defined by two values in the master configuration file:

1. Server root path (SVRROOT)
2. Document root path (DOCROOT)

No documents that reside above this point in the hierarchy can be accessed. For this example, use the CHGWWWCFG command as shown in Figure 178.

```
                      Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Connection queue size  . . . . . CONNQUESIZ     64
Initial request processors . . . INITRPS        1
Maximum request processors . . . MAXRPS         50
Request wait timeout . . . . . . TIMEOUT        120
Wait threshold . . . . . . . . . THRESHOLD      5
Enable scripts . . . . . . . . . ENABLESCPT     *INSIDESCRIPTLIB
Content CCSID  . . . . . . . . . CNTNTCCSID     819
Default source type  . . . . . . DEFSRCTYPE     *ROOT
Disable server:                  DISABLESVR
```

Figure 178. Set Default Source File System

```
                    Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.


 Document root path . . . . . . . DOCROOT        'WEBDOCS/PRIVATE'

 Document root for QDLS . . . . . DOCROOTQ     > *NONE

 Document root for QSYS . . . . . DOCROOTSYS     *NONE
 Server root path . . . . . . . . SVRROOT        '/HARM'
```

Figure 179. Allow Only ROOT Web Documents

9.3.5.6 Commerce Server/400 Access Control

The following functions are configured to control access:

- SSL
- Host filtering
- User authentication

SSL: Two parameters need to be set to enable the SSL protocol.

1. Use the CHGWWWCFG command to set the PROTOCOLS parameter in the master configuration file as shown in Figure 180. In our example, this allows the server to use both protocols.

```
                    Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Public user directory  . . . . . PUBUSERDIR     'PUBHTML'

Temporary directory  . . . . . . TEMPDIR        'TMP'

Initial library list . . . . . . INLLIBL        *CURRENT
Password storage . . . . . . . . PSWDSTG        *COMPATIBLE
Server protocols . . . . . . . . PROTOCOLS
                                        *SSL
Webulator user file path . . . . WBLUSRFILE     *NONE

Update executing RPs . . . . . . UPDATE         *DEFER
```

Figure 180. Change WWW Configuration

2. The ALLOWEDPROTOCOLS parameter in the access configuration file is configured as shown by the following two figures. Figure 181 on page 221 and Figure 182 on page 221 illustrate setting the ROOT directory to allow **only** SSL protocol. This attribute is inherited by all lower subdirectories unless explicitly altered.

```
*STANDARD           Work with WWW Directory Configurations
                                                          System: HARM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory configuration:   /HARM/CFG/PVTACC.CFG



Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   6=Work with limits
  8=Work with parsed buttons    9=Work Virtual Keyboard   10=Change Webu
  14=Change Commerce Server/400

Opt  Directory

14   /
     /HARM/PRIVATE
```

Figure 181. Work with WWW Directory Configuration

```
                            Set Allowed Protocols

 Directory . . . . . :   /



 Select (/) choices and press Enter.  By making no selection, values from
   previous level will be inherited.

 Allowed protocols . . .   /  SSL

 Update executing RPs      *IMMED       *DEFER, *IMMED
```

Figure 182. Set Allowed Protocols

Host Filtering: Host filtering is configured to limit which clients can access the server. Set the configuration so that only hosts from **ibm.com** are allowed access. This configuration value works in conjunction with the network design that uses packet filtering to not allow incoming packets from the untrusted network to have a source name ending with **ibm.com**. The first step is to use the WRKWWWDIR command to add access methods. Take option 6 for the /directory and take option 1 to add all methods.

```
 *STANDARD           Work with WWW Directory Configurations
                                                          System:   RC
 Update executing RPs . .   *DEFER     *DEFER, *IMMED
 Directory configuration:   /harm/cfg/PVTACC.cfg



 Type options, press Enter.
   1=Add   2=Change   4=Remove   5=Display   6=Work with limits
   8=Work with parsed buttons    9=Work Virtual Keyboard   10=Change Webu
   14=Change Commerce Server/400

 Opt  Directory

 6    /
      /HARM/WEBDOCS/PRIVATE/RESTRICT/

F3=Exit   F5=Refresh   F6=Print   F12=Cancel
```

Figure 183. Allow Only Hosts from IBM

```
*STANDARD                   Work with WWW Limits
                                                        System:  SYSTEM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory . . . . . :   /
Type options, press Enter.
  1=Add   2=Change order   4=Remove   5=Work with allow/deny/require

Opt     Access methods

 1      GET PUT POST DELETE HEAD
```

Figure 184. Allow Only Hosts From IBM

Next, take Option 2 to change the order.

```
                            Change Filter Order

Directory . . . . . :   /



Access methods  . . . :   GET PUT POST DELETE HEAD



Order . . . . . .   DENY,ALLOW         allow,deny, deny,allow,
                                         mutual-failure, *DEFAULT, *
Update executing RPs    *DEFER         *DEFER, *IMMED
```

Figure 185. Change Order

Return to Figure 184 and take Option 5 to set the host filtering as shown in Figure 186. Add "Deny all" and "Allow .ibm.com."

```
*STANDARD                   Work with WWW Limits
                                                          System:   RC
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory . . . . . :   /



Type options, press Enter.
  1=Add   2=Change order   4=Remove   5=Work with allow/deny/require

Opt     Access methods

 5      GET PUT POST DELETE HEAD
```

Figure 186. Allow Only Hosts From IBM

```
*STANDARD              Work with WWW Allow/Deny/Require
                                                              System:   RC
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory . . . . . :   /



Access methods  . . . :   GET PUT POST DELETE HEAD



Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display

Opt  Keyword  Value

     ALLOW    .ibm.com
     DENY     ALL
```

Figure 187. Allow Only Hosts from IBM

User Authentication: User authentication specifies which users are allowed to access documents in a particular directory. In this example, we have a directory named /HARM/WEBDOCS/PRIVATE/RESTRICT that can be accessed only by the user named Claus. These are the steps to configure the access control:

1. Add user Claus to the user file.
2. Add the require limit to the /HARM/WEBDOCS/PRIVATE/RESTRICT directory.

```
                            Select WWW User File
                                                              System:   RC
Directory configuration:   /harm/cfg/PVTACC.cfg
Type option, press Enter.
  1=Select

Opt  User file path
     /harm/cfg/user.cfg
```

Figure 188. Select WWW User File

```
                            Work with WWW Users
                                                          System:  HARM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
User file path . . . :   /harm/cfg/user.cfg



Type options, press Enter.
  1=Add   2=Change   3=Add same as   4=Remove   5=Display

Opt  User name

 _   CLAUS
```

Figure 189. Work with WWW Users

```
*STANDARD                    Work with WWW Limits
                                                              System:   RC
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory . . . . . :   /HARM/WEBDOCS/PRIVATE/RESTRICT/



Type options, press Enter.
  1=Add   2=Change order   4=Remove   5=Work with allow/deny/require

Opt     Access methods

        GET PUT POST DELETE HEAD
```

Figure 190. Work with WWW Limits

```
*STANDARD              Work with WWW Allow/Deny/Require
                                                      System: HARM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory . . . . . :   /HARM/WEBDOCS/PRIVATE/RESTRICT/



Access methods  . . . :   GET PUT POST DELETE HEAD



Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display

Opt  Keyword  Value

     REQUIRE  USER claus
```

Figure 191. Work with WWW Allow/Deny/Require

9.3.5.7 Commerce Server/400 Additional Configuration Values

The following section and sample composite displays illustrate setting several additional configuration values.

- The *Index Default View* value enables or disables the creation of dynamic indexes and sets their default appearance. Set the value to *NONE to prevent browser clients from listing files in the directory.
- The *ACCLOGFILE* value specifies the path and name of the access log that records all attempts to access the server. If this is blank (a command value of *NONE), access logging is turned off. We recommend that you provide a name for the access log. Access logging does create overhead for the CPU so you may want to turn it off if security is less of a concern than performance.
- The *ERRLOGFILE* value specifies the path and name of the error log file that records all server errors. If this is blank (a command value of *NONE), error logging is turned off. We recommend that a name be provided for the error log.
- The *STTLOGFILE* value specifies the path and name of the statistical log. The statistics log is a record of operational events and, from a security point of view, can be analyzed for unusual patterns or trends. If this is blank (a command value of *NONE), statistics logging is turned off. We recommend that a name be provided for the statistical log.

- The *Default Source Type* value indicates the file system to use when any of the following cases occurs:
  - A URL has no alias.
  - An alias does not have an explicit source type.
  - The server's home page is requested ('/').

  This value does not have explicit security implications, but must be set to either *ROOT, QSYS, or QDLS. Since we are setting our scope of control to specify only documents in the root files system that is accessed, we set this value to *ROOT.

These values are set using the CHGWWWCFG command, which, in this example, is run against the master configuration file, WEBSERV.CFG.

```
                    Change WWW Configuration (CHGWWWCF)

Type choices, press Enter.

Master Configuration File  . . . CFGFILE        '/WWWSERV/CFG/WEBSERV.CFG

Index default view:              IDXDFTVIEW
                                                 *NONE
                                                 *OFF

Access log file path:            ACCLOGFILE
                                                 'LOGS/ACCESS.LOG'

Error log file path:             ERRLOGFILE
                                                 'LOGS/ERROR.LOG'

Statistics log file path:        STTLOGFILE
                                                'LOGS/STATS.LOG'

Default source type  . . . . . . DEFSRCTYPE     *ROOT


Update executing RPs . . . . . . UPDATE         *DEFER


F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 192. Set Multiple Configuration Values

The next step is to enable SSL for the server. Please refer to Section 1.9.6, "What is Secured Sockets Layer (SSL)?" on page 31 for more information on SSL protocol.

The final value to be set is the state of Webulator. The shipped default is to not start the Webulator. We leave the value shown in Figure 193.

```
                    Change WWW Configuration (CHGWWWCFG)

 Type choices, press Enter.

 Master Configuration File  . . . CFGFILE        '/HARM/CFG/PVTSERV.CFG

 Webulator user file path . . . . WBLUSRFILE     *NONE

 Maximum Webulator sessions . . . WBLMAXSSN      20
 Disable Webulator:               DISABLEWBL
                                                > *YES
```

Figure 193. Set Webulator to Not Start

The /HARM/Cfg/pvtserv.cfg file is shown in Figure 194 on page 226. Please notice that not all master configuration values are included in the file. Some default values are assumed by the server daemon and are not written to this file. The CHGWWWCFG, CHGWWWSEC, and WRKWWWINCL commands must be used to verify values.

```
;
; Private Commerce Server/400 master configuration file
; for Example 2 in the HARM book
;

DOCUMENTROOTQDLS

GlobalAdminAccessCfgFile Cfg/AdAccess.cfg


INDEXDEFAULTVIEW  NONE    OFF

INDEXNAME  PVTHOME.HTM

IndexStyle IncludeAll IncludeHTMLTitles

ScriptLibraries WWWCGI

ServerSideInclude AllowExec
DOMAINNAMELOOKUP  MINIMAL
COMMERCEKEYLISTFILE  /wwwserv/key/keylist.cfg
SERVERPROTOCOLS  SSL
GLOBALACCESSCFGFILE  cfg/pvtacc.cfg
DISABLEWEBULATOR  YES
DOCUMENTROOT  WEBDOCS/PRIVATE
SERVERROOT  /HARM
ACCESSLOG  LOGS/PVTACC.LOG  850
ERRORLOG  LOGS/PVTERR.LOG  850
PORT  83
SERVERROOT  /harm
COMMERCESSLPORT  444
STATISTICSLOG  LOGS/PVTSTAT.LOG  850
```

Figure 194. Private Commerce Server Master Configuration File

The access configuration file contains directory-based configuration values. Some defaults are assumed if this file is empty or does not exist. Directives are inherited by subdirectories unless explicitly altered. For example, the /HARM/WEBDOCS/PRIVATE subdirectory allows only SSL since it inherits that property from the root directory.

```
<DIRECTORY /HARM/WEBDOCS/PRIVATE/RESTRICT>
   AUTHUSERFILE /harm/cfg/user.cfg STREAM
   AUTHTYPE BASIC
   AUTHNAME Private Restricted Area
   <LIMIT GET PUT POST DELETE HEAD>
      REQUIRE user claus
   </LIMIT>
</DIRECTORY>
<DIRECTORY />
   <LIMIT GET PUT POST DELETE HEAD>
      DENY FROM all
      ORDER DENY,ALLOW
      ALLOW FROM .ibm.com
   </LIMIT>
   ALLOWEDPROTOCOLS SSL
</DIRECTORY>

;
; Private Commerce Server/400 access configuration file
; Example 2 of the HARM book


;
; Private Commerce Server/400 access configuration file
;
```

Figure 195. Private Web Server Access Configuration File

9.3.6 Example 3 - WEBULATOR

This example illustrates a Webulator solution that allows access to a limited set of INTERNET users. This network design is referred to as the Integrated Server design shown in Figure 196.

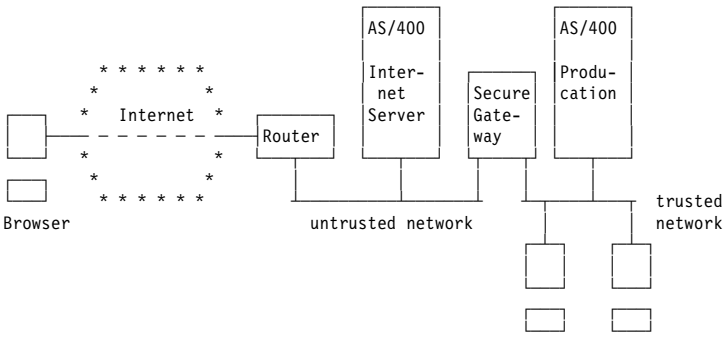


Figure 196. AS/400 System as Integrated Internet Server

9.3.6.1 Exposures

The exposures in this example can be of greater consequence than those associated with HTTP servers. While a breakdown in security in our first two examples might compromise the privacy of an HTML document, a breech of security in this scenario can result in the user gaining access to an AS/400 command line. Fortunately, as in our first two examples, more than one layer of security is implemented, which does provide a secure solution.

9.3.6.2 Security Policy

In this example, we support our security policy in the following areas:

- Restrict use to only one host name.
- Restrict use to only one user profile.
- Ensure transaction security by using SSL.

9.3.6.3 Security Overview for Example 3

The following layers are utilized to provide and control access.

- At the lowest level, network design provides connectivity to the Internet and limits protocols to SSL. The network design does provide connectivity to the internal network.
- Transaction level security is provided by the SSL protocol.
- TCP/IP configuration provides control at the transport and network layer (for example, port restrictions).
- OS/400 configuration limits IFS access by way of the user profile of the Web server daemon and by object authority of database files and CGI programs.
- Web server configuration adds a higher layer of scope control by limiting the Web server's access to documents in only the ROOT file system and by specifying entry points into the ROOT file system. The Web server configuration also provides discretionary access control by utilizing user authentication.

9.3.6.4 Browser Windows

Two windows are used to illustrate this example:

- Webulator Sign On Window (Figure 210 on page 234)
- Webulator Main Menu (Figure 211 on page 235)

9.3.6.5 Configure the Commerce Server/400 for Webulator

Configuration consists of setting OS/400 object authority and configuring Commerce Server/400 scope control and user authentication.

See Table 19 on page 236 for the exact values that are used in each of the implementation examples.

The following steps are taken for this example:

__ 1. Use the WRKAUT command to give the WWWUSER profile access to the log files in the HARM/LOGS directory.
__ 2. Use the GRTOBJAUT or EDTOBJAUT command to give the WWWUSER profile access to:
   - The WWWSERVER library and all contained objects
   - The QSYS/QSYSNOMAX job queue
__ 3. Use the WRKWWWINCL command to specify libraries that may contain Web documents.
__ 4. Use the CHGWWWCFG command to set the following server characteristics:
   - Access configuration file name
   - Server root
   - Document root for:
     - Root file system
     - QDLS file system

        - QSYS.LIB file system
      - Index default view
      - Log file locations
      - Default source type
      - Index name
      - HTTP port number
      - Supported protocols
      - Webulator status

__ 5. Use the CHGWWWSEC command to define:
      - Key list path and file name
      - SSL port number

__ 6. Use the CHGWWWDIR command to define:
      - Authorization user file path and file name
      - Authorization type
      - Authorization name

__ 7. Use the CHGWBLCFG command to set the sign-on method.

__ 8. Use the WRKWWWLIM command to set:
      - Access methods
      - Limit users by setting host filtering.

__ 9. Prevent sign-on window.

OS/400 Object Authority: To support our security policy, which does not allow all objects to be available to all users, we must set system security level (QSECURITY) to at least 30. We strongly recommend at least level 40. Our configuration was developed and tested on a V3R7 system at level 50.

Before beginning specific object authority configuration, please refer to Chapter 2, "Start Here by Securing OS/400®" on page 37 for information on completing a security review of the system. The server user profile should not be able to sign on to the system and, thus, have a password value of *NONE. Authority to the log files subdirectory is configured as shown in Figure 197.

```
                              Work with Authority

Object . . . . . :   /HARM/LOGS
Owner  . . . . . :   WEBMASTER
Primary group  . . . . . :   *NONE
Authorization list . . . . . :   *NONE


Type options, press Enter.
  1=Add user   2=Change user authority   4=Remove user

                   Data     --Object Authorities--
Opt  User        Authority  Exist  Mgt  Alter  Ref

     *PUBLIC     *EXCLUDE
     WEBMASTER   *RWX         X     X     X     X
     WWWUSER     *RWX         X     X     X     X
```

Figure 197. Work with Authority

Jobs that serve content (the server daemons) run under the configured server profile (WWWUSER by default) and should have object authority configured as shown in Figure 198 on page 230.

```
                          Edit Object Authority

Object . . . . . :   WWWSERVER       Owner  . . . . . :  WWWUSER
  Library  . . . . :     QSYS          Primary group  . . . :  *NONE
Object type  . . . . :   *LIB

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . .   *NONE

                         Object
User        Group       Authority
WWWUSER                 *CHANGE
WEBMASTER               *ALL
*PUBLIC                 *EXCLUDE
```

Figure 198. Give WWWUSER Authority To Server Program Objects

Use the GRTOBJAUT or EDTOBJAUT command to grant WWWUSER authority to the QSYS/QSYSNOMAX job queue.

```
                          Edit Object Authority

Object . . . . . :   QSYSNOMAX       Owner  . . . . . :   QSYS
  Library  . . . . :     QSYS          Primary group  . . . :   *NONE
Object type  . . . . :   *JOBQ

Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . .   *NONE

                         Object
User        Group       Authority
QSYS                    *ALL
  .                       .
  .                       .
  .                       .
WWWUSER                 *USE
QSYSOPR                 *USE
QSPL                    *CHANGE
QPGMR                   *USE
*PUBLIC                 *USE
```

Figure 199. Give WWWUSER Authority to the Job Queue

As well as granting authority to objects available to the server, make sure that authority is revoked where it is not needed. For example, authority is given to /QDLS/WebDocs folder when Commerce Server/400 is installed. Use the WRKFLR command and option 14 to remove access by the server as illustrated in Figure 200 on page 231.

```
                          Change Authorized Users

Folder . . . . . . . . . . . . :   WEBDOCS
  In folder  . . . . . . . . . :     *NONE


Type changes, press Enter.
Authority:  *ALL, *CHANGE, *USE, *EXCLUDE

User        Group       Authority        User        Group     Authority
WWWUSER                 *EXCLUDE
```

Figure 200. Remove Web Server Authority

9.3.6.6 Commerce Server/400 Scope Control

The features of Commerce Server/400 are used to provide another layer of control over the scope of information accessible through the Web server.

The *IncludeLibraries* parameter in the master configuration file is used to control the scope of documents that may be accessed by the Web server. Only libraries included in the Include Libraries list can be accessed by a URL that points into the QSYS file system.

The WRKWWWINCL command is used to set this value. For our example, we configure no libraries available for serving Web documents as shown in Figure 201.

```
                        Work with WWW Include Library
                                                    System:   SYSTEM
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Configuration file path:   /WWWSERV/CFG/WBLSERV.CFG



Type options, press Enter.
  1=Add   2=Change   3=Add same as   4=Remove   5=Display

Opt  Libraries
 _   ____________________________________________________________
```

Figure 201. Allow No Libraries to Contain Web Documents

```
                    Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Master Configuration File  . . . > '/harm/CFG/wblSERV.CFG'

Directory based configuration:
                              'cfg/wblacc.cfg'

  file CCSID . . . . . . . . . .   *SAME         1-65534, *SAME, *DEFAULT
Administrator access file path     'Cfg/AdAccess.cfg'

Alias file path  . . . . . . . .   'CFG/ALIAS.CFG'

Content type file path . . . . .   'CFG/CONTENT.CFG'

Default content type . . . . . .   'TEXT/PLAIN'
```

Figure 202. Set Access Configuration File Name

In this example, no serving of documents is allowed. Therefore, we configure each file system to have a value of *NONE for the document root parameter. This action, plus the WRKWWWINCL command use as stated in Figure 201 on page 231, prevents the Web server from serving any documents through a GET request.

```
                    Change WWW Configuration (CHGWWWCFG)
Type choices, press Enter.

Document root path . . . . . . . > *NONE

Document root for QDLS . . . . .   *NONE

Document root for QSYS . . . . .   *NONE         Name, *SAME, *NONE
Server root path . . . . . . . .   '/HARM'
```

Figure 203. Disable Access to Web Documents in All File Systems

Even though this implementation does not use all of the log files, each of them must be defined with a unique name as shown in Figure 204.

```
                    Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Access log file path:            ACCLOGFILE
                                              'LOGS/WBLACC.LOG'

Error log file path:             ERRLOGFILE
                                              'LOGS/WBLERR.LOG'

Statistics log file path:        STTLOGFILE
                                             'LOGS/WBLSTAT.LOG'


Socket Port  . . . . . . . . . . PORT           84

Update executing RPs . . . . . . UPDATE         *DEFER


F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 204. Set Log File Paths and Names

```
                    Change WWW Configuration (CHGWWWCFG)

Type choices, press Enter.

Public user directory  . . . . .   'PUBHTML'

Temporary directory  . . . . . .   'TMP'

Initial library list . . . . . .   *CURRENT      *SAME, *DEFAULT, *CURRENT
Password storage . . . . . . . .   *COMPATIBLE   *SAME, *DEFAULT...
Server protocols . . . . . . . .   *SSL          *SAME, *DEFAULT, *HTTP,

Webulator user file path . . . .   *NONE

Maximum Webulator sessions . . .   20           1-9999, *SAME, *DEFAULT
Disable Webulator:
                                   *NO          *SAME, *DEFAULT, *YES,
  date available . . . . . . . .   000000       Date, 000000
  time available . . . . . . . .   000000       Time, 000000
Update executing RPs . . . . . .   *DEFER       *IMMED, *DEFER
```

Figure 205. Enable Webulator

Use the CHGWWWDIR command to set the authentication realm, type, and sign-on method.

```
                    Change WWW Directory (CHGWWWDIR)

Type choices, press Enter.

Master Configuration File  . . . CFGFILE      > '/harm/CFG/wblSERV.CFG'
Standard or Administrative . . . STDORADM     > *STANDARD
Directory  . . . . . . . . . . . DIRECTORY    > '/*META/WEBULATOR/'
Authentication realm . . . . . . AUTHNAME       'WEBULATOR'


Authentication type  . . . . . . AUTHTYPE       *BASIC


User file path:                  USRFILE
                                                *INHERIT

  file type  . . . . . . . . . .                *SAME
Group file path  . . . . . . . . GRPFILE        *INHERIT

File CCSID . . . . . . . . . . . FILECCSID      *INHERIT
```

Figure 206. Set Authname Authtype

```
                  Change Webulator/400 Config (CHGWBLCFG)

Type choices, press Enter.

Master Configuration File  . . . > '/harm/CFG/wblSERV.CFG'
Directory  . . . . . . . . . . . > '/*META/WEBULATOR/'
Sign-on method:
                            *USEAUTHENTICATION
  User name  . . . . . . . . . .   *SAME         Character value, *SAME.
  Allow signon overrides . . . .   *INHERIT      *SAME, *INHERIT, *YES,
Extended input field . . . . . .   *INHERIT      *SAME, *INHERIT, *TEXTA
Background color ID  . . . . . .   *INHERIT      000000-FFFFFF, *SAME, *
Background image URL . . . . . .   *INHERIT
```

*Figure 207. Set Sign-On Method to *USEAUTHENTICATION*

```
*STANDARD                    Work with WWW Limits
                                                              System:   RCH
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory . . . . . :   /*META/WEBULATOR/



Type options, press Enter.
  1=Add   2=Change order   4=Remove   5=Work with allow/deny/require

Opt     Access methods

        GET PUT POST DELETE HEAD
```

Figure 208. Set Access Methods

```
*STANDARD              Work with WWW Allow/Deny/Require
                                                              System:   SYS
Update executing RPs . .   *DEFER     *DEFER, *IMMED
Directory . . . . . :   /*META/WEBULATOR/



Access methods  . . . :   GET PUT POST DELETE HEAD



Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display

Opt  Keyword  Value

     REQUIRE  USER wwwmstr
```

Figure 209. Limit Webulator Users to WWWMSTR

As shown by the following displays, the user must sign on at the pop-up window rather than an AS/400 sign-on display. The initial menu or initial program is controlled by the user profile.

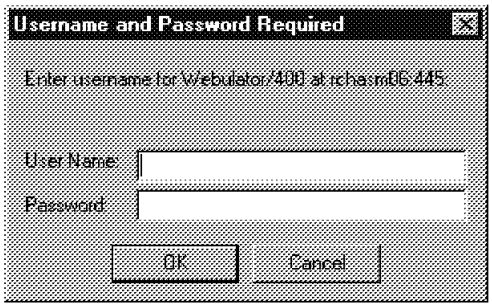


Figure 210. Webulator Sign-On Window

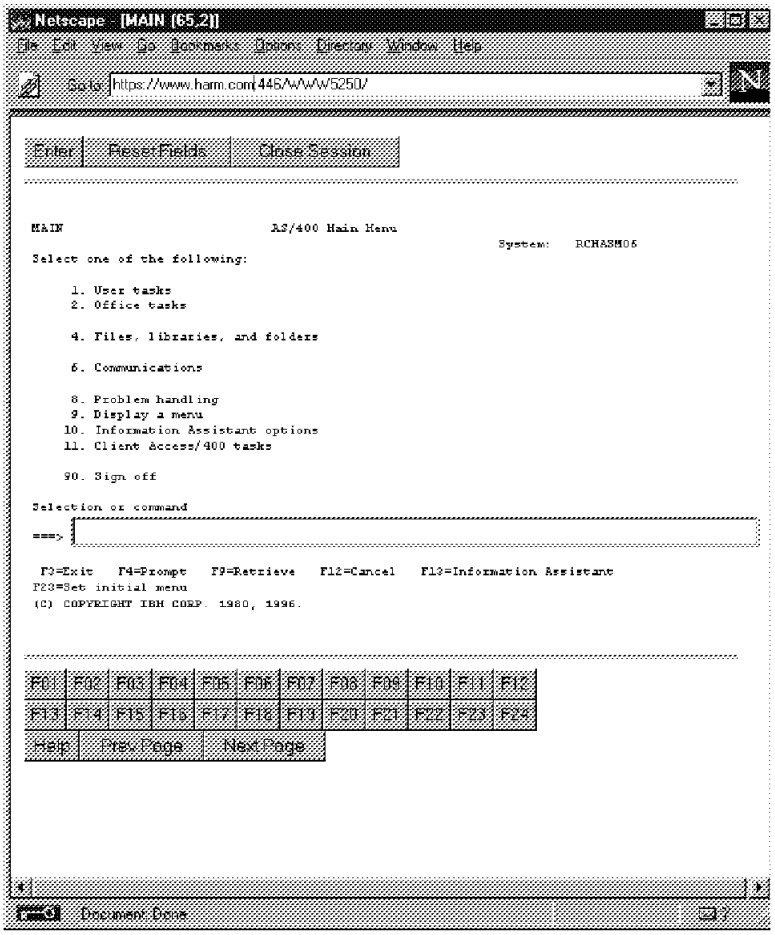


Figure 211. Webulator Main Menu

Table 19. Commerce Server/400 Configuration Values

Configuration Parameter	Configuration Command	SECDEMO Values	Public Web Server	Private Web Server	Webulator Values	Contained in File	Description/Comments
INCLUDELIBRARIES	WRKWWWINCL	(blank)	(blank)	(blank)	(blank)	CFGFILE	Allow Web Documents in Library
CFGFILE	CHGWWWCFG	/CFG /WEBSERV.CFG	/CFG /PUBSERV.CFG	/CFG /PVTSERV.CFG	/CFG /WBLSERV.CFG	CFGFILE	Master Configuration File
ACCGBLFILE	CHGWWWCFG	/CFG ACCESS.CFG	/CFG PUBACC.CFG	/CFG PVTACC.CFG	/CFG WBLACC.CFG	CFGFILE	Global Access Configuration File
DOCROOT	CHGWWWCFG	/WEBDOCS	/WEBDOCS /PUBLIC	/WEBDOCS /PRIVATE	*NONE	CFGFILE	Root File System Document Root
DOCROOTQ	CHGWWWCFG	*NONE	*NONE	*NONE	*NONE	CFGFILE	QDLS Document Root
DOCROOTSYS	CHGWWWCFG	*NONE	*NONE	*NONE	*NONE	CFGFILE	QSYS.LIB Document Root
SVRROOT	CHGWWWCFG	/WWWSERV	/HARM	/HARM	/HARM	CFGFILE	Server Root
IDXDFTVIEW	CHGWWWCFG	*NONE *OFF	*NONE *OFF	*NONE *OFF	*NONE *OFF	CFGFILE	Index Default View
ACCLOGFILE	CHGWWWCFG	/LOGS /ACCESS.LOG	/LOGS /PUBACC.LOG	/LOGS /PVTACC.LOG	/LOGS /WBLACC.LOG	FGFILE	Access Log File
ERRLOGFILE	CHGWWWCFG	/LOGS /ERRLOG.LOG	/LOGS /PUBERR.LOG	/LOGS /PVTERR.LOG	/LOGS /WBLERR.LOG	FGFILE	Error Log File
STTLOGFILE	CHGWWWCFG	/LOGS /STATS.LOG	/LOGS /PUBSTAT.LOG	/LOGS /PVTSTAT.LOG	/LOGS /WBLSTAT.LOG	CFGFILE	Statistics Log File
DEFSRCTYPE	CHGWWWCFG	*ROOT	*ROOT	*ROOT	*ROOT	CFGFILE	Default Source Type
IDXNAME	CHGWWWCFG	example1.htm	pubhome.htm	pvthome.htm	index.htm	CFGFILE	Index Name
PORT	CHGWWWCFG	81	82	83	84	CFGFILE	HTTP Port Number
PROTOCOLS	CHGWWWCFG	*HTTP *SSL	*HTTP	*SSL	*SSL	CFGFILE	Supported Protocols
DISABLEWBL	CHGWWWCFG	*YES	*YES	*YES	*NO	CFGFILE	Disable Webulator
KEYFILE	CHGWWWSEC	/WWWSERV /KEY /KEYLIST2.CFG	N/A	/WWWSERV /KEY /KEYLIST2.CFG	/WWWSERV /KEY /KEYLIST2.CFG	CFGFILE	Commerce Key List File
SSLPORT	CHGWWWSEC	443	N/A	444	445	CFGFILE	SSL Port Number
AUTHUSERFILE	CHGWWWDIR	/WWWSERV /KEY /KEYLIST2.CFG	N/A	/WWWSERV /KEY /KEYLIST2.CFG	/WWWSERV /KEY /KEYLIST2.CFG	ACCGBLFILE	Access Configuration File
AUTHTYPE	CHGWWWDIR	*BASIC	*BASIC	*BASIC	*BASIC	ACCGBLFILE	Access Configuration File
AUTHNAME	CHGWWWDIR	SECDEMO	N/A	Private Restricted Area	N/A	ACCGBLFILE	Access Configuration File

9.4 Logging

Three different log files are maintained by each Commerce Server/400 daemon that is running.

- Access log
- Statistics Log
- Error Log

Although the primary use of these logs is for performance monitoring and creating demographic reports, they can also be useful for security auditing tools. Rather than looking for specific information, the best technique is to look for anomalies. The first step in the process is to establish a baseline for each of the logs. Second, automate a process to compare daily log activity to the baseline and continuously update the baseline. The following list contains some patterns you might try to detect:

- Access log:
  - Excessive peaks of activity, especially at night
  - Disproportional hits from one address
  - Top 10 document hit patterns vastly different than the baseline
- Statistics log:
  - Requests per second should be base-lined and monitored for radical changes.
- Error log:
  - Scan the log for illegal URLs. For example, scan for the word QSYS or other IBM library names that do not contain Web documents. The following message is from an error log. This indicates that someone is experimenting with URLs in an attempt to access information other than what you intended.

    ```
    RP1:  Error:  The library QSYS is not in the list of valid QSYS libraries.
    ```

By default, the Commerce Server/400 logs are created in the ROOT file system. In this format, they are directly usable to the many tools available for analyzing logs. For a sample of analysis tools, link to the following URLs:

```
http://serverwatch.iworld.com/tools/usage.html
http://serverwatch.iworld.com/tools/monitor.html
```

Figure 212 and Figure 213 on page 238 are two example charts that can be easily produced from the access log.

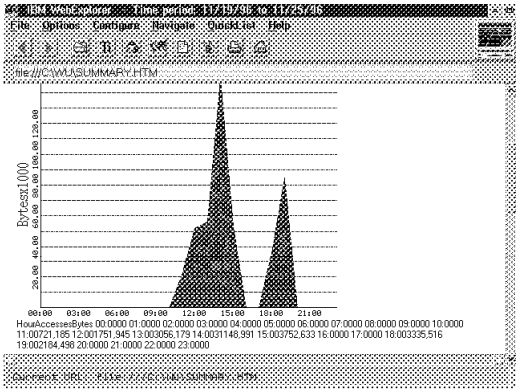


Figure 212. Log Analysis by Time Slice

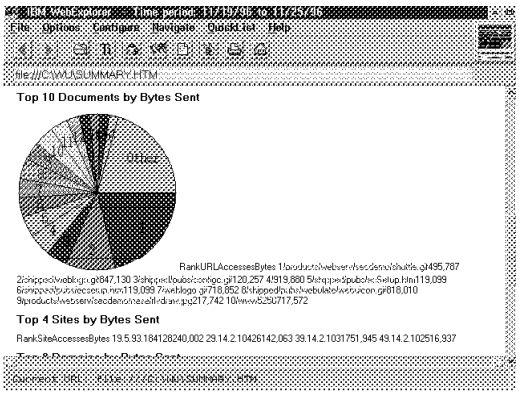


Figure 213. Log Analysis by Document Accessed

Log files can also be written in the QSYS file system as database files. These logs can be analyzed and audited with AS/400 query tools.

9.4.1 Status Codes for Access Log

The following example shows status codes as specified in the HTTP 1.0 specification.

```
2xx Success
       The requested action was successfully received and understood
              200 OK
              201 Created
              202 Accepted
              203 Provisional Information
              204 No Response
              205 Deleted
              206 Modified
3xx Redirection
       Further action must be taken in order to complete the request
              301 Moved Permanently
              302 Moved Temporarily
              303 Method
              304 Not Modified
4xx Client Error
       The request contains bad syntax or is inherently impossible to ful
              400 Bad Request
              401 Unauthorized
              402 Payment Required
              403 Forbidden
              404 Not Found
              405 Method Not Allowed
              406 None Acceptable
              407 Proxy Authentication Required
              408 Request Timeout
5xx Server Error
       The server could not fulfill the request
              500 Internal Server Error
              501 Not Implemented
              502 Bad Gateway
              503 Service Unavailable
              504 Gateway Timeout
```

9.5 Audit Considerations

The objects that need to be audited include user profiles, special authorities, object authorities, configuration files, and CGI programs.

9.5.1 User Profiles

Verify that WWWUSER has Password *NONE:

```
PRTUSRPRF TYPE(*PWDINFO) SELECT(*USRCLS)
```

```
                                    User Profile Information                                                Page     1
5716SS1 V3R7M0  961108                                                                                HARM      12/13/96  13:11:32
 Report type  . . . . . . . . . :   *PWDINFO
 Select by  . . . . . . . . . . :   *USRCLS
 User class . . . . . . . . . . :   *ALL
 QPWDEXPITV system value  . . . :       31
 User                        Not Valid         No        Previous     Password     Expiration     Password
 Profile        Status       Sign-ons       Password     Sign-on      Changed       Interval      Expired
 ANONYMOUS      *ENABLED            0          X          /  /       12/11/96      *SYSVAL        *NO
 LABUSER        *ENABLED            0                     07/10/96     07/10/96      *SYSVAL        *NO
 MRK            *ENABLED            0                     12/09/96     12/02/96      *SYSVAL        *NO
 M36DFT         *ENABLED            0                     04/11/96     04/11/96      *SYSVAL        *NO
 PAYROLLUSR     *ENABLED            0                       /  /       12/06/96      *SYSVAL        *NO
 WEBMASTER      *ENABLED            0                     11/23/96     11/16/95      *SYSVAL        *NO
 QTCP           *ENABLED            0          X            /  /       10/31/95      *SYSVAL        *NO
 QTMHHTP1       *ENABLED            0          X            /  /       10/23/96      *NOMAX         *NO
 QTMHHTTP       *ENABLED            0          X            /  /       10/23/96      *NOMAX         *NO
 QTMPLPD        *ENABLED            0          X            /  /       10/31/95      *NOMAX         *NO
 WEBUSER        *ENABLED            0                     12/03/96     12/03/96      *SYSVAL        *NO
 WWWUSER        *ENABLED            0          X          12/09/96     12/10/96      *SYSVAL        *NO
                                  * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 214. PRTSYSSECA Report - Check that WWWUSER has No Password

9.5.2 Special Authorities

Verify that *only* the intended user profiles have *ALLOBJ or *IOSYSCFG special authorities:

```
PRTUSRPRF TYPE(*AUTINFO) SELECT(*SPCAUT) SPCAUT(*ALLOBJ *IOSYSCFG) +
USRCLS(*ALL)
```

```
                                    User Profile Information                                                Page     1
5716SS1 V3R7M0  961108                                                                                HARM      12/13/96  13:34:37
 Report type  . . . . . . . . . :   *AUTINFO
 Select by  . . . . . . . . . . :   *SPCAUT
 Special authorities  . . . . . :   *ALLOBJ     *IOSYSCFG
                                   -------------Special Authorities-------------
                                                *IO                                                                    Group
 User        Group       *ALL  *AUD  SYS  *JOB  *SAV  *SEC  *SER  *SPL  User                 Owner       Group       Authority   Limited
 Profile     Profiles     OBJ   IT   CFG   CTL   SYS   ADM  VICE   CTL  Class                Owner       Authority   Type        Capability
 BOSS        *NONE              X     X     X     X           X     X    *SECOFR              *USRPRF     *NONE       *PRIVATE    *NO
 QLPAUTO     *NONE        X           X     X     X     X                  *SYSOPR              *USRPRF     *NONE       *PRIVATE    *NO
 QLPINSTALL  *NONE        X           X     X     X     X                  *SYSOPR              *USRPRF     *NONE       *PRIVATE    *NO
 QSECOFR     *NONE        X     X     X     X     X     X     X     X    *SECOFR              *USRPRF     *NONE       *PRIVATE    *NO
 QSYS        *NONE        X     X     X     X     X     X     X     X    *SECOFR              *USRPRF     *NONE       *PRIVATE    *NO
 WEBMASTER   *NONE                    X                                  *USER                *USRPRF     *NONE       *PRIVATE    *NO
                                  * * * * *  E N D  O F  L I S T I N G  * * * * *
```

*Figure 215. PRTSYSSECA Report - Users with Special Authority *IOSYSCFG and *ALLOBJ*

9.5.3 Object Authority

For the libraries, directories, and subdirectories that the Commerce Server/400 serves HTML pages from, verify that:

- WWWUSER has Read and eXecute rights.
- *PUBLIC is *EXCLUDE where access control is implemented.

```
DSPAUT OBJ('/HARM/WEBDOCS/PRIVATE/RESTRICT')
```

```
                              Display Authority                              Page    1
5716SS1 V3R7M0  961108                                                HARM     01/20/97  07:05:08
 Object . . . . . . . . . . . :  /harm/webdocs/public
 Owner  . . . . . . . . . . . :  WEBMASTER
 Primary group  . . . . . . . :  *NONE
 Authorization list . . . . . :  *NONE
            Data      --------------------Data Authorities---------------------    ------Object Authorities-------
 User       Authority   Objopr    Read    Add    Update    Delete    Execute    Exist    Mgt    Alter    Ref
 *PUBLIC     *RX          X       X                          X
 WEBMASTER   *RWX         X       X      X       X         X         X        X      X      X       X
 WWWUSER     *RWX         X       X      X       X         X         X        X      X      X       X
                        ***** E N D  O F  L I S T I N G *****
```

Figure 216. WWWUSER and PUBLIC Authority to /HARM/WEBDOCS/PUBLIC Directory

```
DSPAUT OBJ('/HARM')
```

```
                              Display Authority                              Page    1
5716SS1 V3R7M0  961108                                                HARM     01/20/97  06:22:31
 Object . . . . . . . . . . . :  /harm/webdocs/private/restrict/reshome.htm
 Owner  . . . . . . . . . . . :  WEBMASTER
 Primary group  . . . . . . . :  *NONE
 Authorization list . . . . . :  *NONE
            Data      --------------------Data Authorities---------------------    ------Object Authorities-------
 User       Authority   Objopr    Read    Add    Update    Delete    Execute    Exist    Mgt    Alter    Ref
 *PUBLIC     *EXCLUDE
 WEBMASTER   *RWX         X       X      X       X         X         X        X      X      X       X
 WWWUSER     *RX          X       X                                  X
                        ***** E N D  O F  L I S T I N G *****
```

Figure 217. DSPAUT Report - WWWUSER and PUBLIC Authority to /HARM/WEBDOCS/PRIVATE/RESTRICT/reshome.htm

Verify that the WWWUSER user profile has public authority set to *EXCLUDE:

```
DSPOBJAUT OBJ(WWWUSER) OBJTYPE(*USRPRF)
```

```
                              Display Object Authority                                          Page     1
5716SS1 V3R7M0  961108                                                                  HARM      12/13/96  17:09:34
 Object . . . . . . . :   WWWUSER                          Owner  . . . . . . . :   QSYS
   Library  . . . . . :     QSYS                           Primary group  . . . :   *NONE
 Object type  . . . . :   *USRPRF

                            Object      -------------Object---------------    --------------------Data--------------------
 User          Group        Authority    Opr    Mgt    Exist    Alter    Ref    Read    Add    Update    Delete    Execute
 WWWUSER                    *ALL          X      X       X        X       X      X       X       X         X          X
 *PUBLIC                    *EXCLUDE
                                    * * * * *  E N D  O F  L I S T I N G  * * * * *
```

Figure 218. DSPOBJAUT Report - Public Authority to WWWUSER User Profile

```
                                        Display Authorized Objects                                          Page     1
5716SS1 V3R7M0  961108                                                                          SYSTEM06  01/19/97  09:29:12
 User Profile . . . . . . . . . . . . . . . :   WWWUSER
                                 -----------Object-------------   -------------Data--------------             List
 Object       Library      Type       Opr   Mgt   Exist   Alter   Ref   Read   Add   Upd   Dlt   Execute   Exclude   Mgt
 QSYSNOMAX    QSYS         *JOBQ       X                                 X                          X
 CGISEC       WWWCGI       *PGM        X                                 X                          X
 WWU444       WWWSERVER    *USRSPC     X     X       X       X     X     X     X     X     X       X
 WWU446       WWWSERVER    *USRSPC     X     X       X       X     X     X     X     X     X       X
 WWU81        WWWSERVER    *USRSPC     X     X       X       X     X     X     X     X     X       X
 WWU82        WWWSERVER    *USRSPC     X     X       X       X     X     X     X     X     X       X
*CPI220D - 90 objects were not included in this list.
                                  * * * * *   E N D   O F   L I S T I N G   * * * * *
DSPUSRPRF USRPRF(WWWUSER) TYPE(*OBJAUT)
```

Figure 219. DSPUSRPRF Display Private Authorities for WWWUSER

Verify that the WWWUSER user profile does not have access to other important libraries such as PAYROLL.

Figure 220. PAYROLL Library Authority

Verify the WWWUSER user profile authority to the CGI library. In our example, this is WWWCGI.

You need to check if the following user profiles have the correct authority to the CGI library:

- WWWUSER - *USE
- WEBMASTER - *ALL
- PUBLIC - *EXCLUDE

```
DSPOBJAUT OBJ(WWWCGI) OBJTYPE(*LIB)
```

```
                                        Display Object Authority                                            Page     1
5716SS1 V3R7M0  961108                                                                                HARM      12/14/96  15:21:31
 Object . . . . . . . :   HARM                                    Owner  . . . . . . . :   QPGMR
   Library  . . . . . :     QSYS                                  Primary group  . . . :   *NONE
 Object type  . . . . :   *LIB
 Object secured by authorization list  . . . . . . . . . . . . :   *NONE
                          Object      -------------Object---------------    --------------------Data--------------------
 User          Group      Authority    Opr    Mgt    Exist    Alter    Ref    Read    Add    Update    Delete    Execute
 WWWUSER                  *USE          X                                      X                                    X
 WEBMASTER                *ALL          X      X       X        X       X      X       X       X         X          X
 *PUBLIC                  *EXCLUDE
                                  * * * * *   E N D   O F   L I S T I N G   * * * * *
```

Figure 221. DSPOBJAUT Report - CGI Library Authorities

9.5.4 Web Server Configuration Files

The three methods used to audit the QATMHTTP do not translate well to Commerce Server configuration files, which are stream files:

1. File level audit can be turned on but no file name is logged in the journal receiver.
2. There is no equivalent command to DSPOBJD for a stream file. Therefore, the WRKLNK ('/filename') and an option 8 must be used to obtain the last change date.
3. The configuration values cannot be easily printed out since default values are not written to the steam file.

To audit unwanted changes to the configuration file, the auditing level must be set (see Figure 222 on page 242).

```
CHGAUD OBJ('/WWWSERV/CFG/WEBSERV.CFG') OBJAUD(*CHANGE)
```

```
                          Change Auditing Value (CHGAUD)

Type choices, press Enter.

Object . . . . . . . . . . . . . > '/WWWSERV/CFG/WEBSERV.CFG'

Object auditing value  . . . . . > *CHANGE       *NONE, *USRPRF, *CHANGE
```

Figure 222. Change Auditing Level for Commerce Server/400 Configuration File

If you start the QAUDJRN journal and select *CHANGE as the object auditing value in the QAUDLVL system value, you can search for a ZC journal entry (an object was changed). You can find the user profile that changed the file, user and device name, job name, and the date and time a change was made. You can use the following command:

```
DSPAUDJRNE ENTTYP(ZC)
```

```
                              QUERY NAME . . . . . QSECZC
                              LIBRARY NAME . . . . QSYS
                              FILE         LIBRARY      MEMBER       FORMAT
                              QASYZCJE     QTEMP        QASYZCJE     QASYZCJE
                              DATE . . . . . 01/20/97
                              TIME . . . . . 14:40:14


01/20/97  14:40:14                                                        PAGE    1
   USER       OBJECT     LIBRARY    OBJECT   JOB        JOB        JOB    DATE       TIME
   PROFILE    NAME       NAME       TYPE     NAME       USER       NUMBER
ZC ADAN       QATMHTTPC  QUSRSYS    *FILE    QPADEV0003 ADAN       013941 010797 13:28:22
ZC BRSMITH    QATMHTTPC  QUSRSYS    *FILE    QPADEV0003 BRSMITH    013966 011097 15:09:26
ZC BRSMITH    QATMHTTPC  QUSRSYS    *FILE    QPADEV0003 BRSMITH    013966 011097 15:26:27
ZC A960321WM  *N         *N         *STMF    QPADEV0003 A960321WM  014236 012097 11:11:43
* * *  E N D  O F  R E P O R T  * * *
```

Figure 223. DSPAUDJRNE Report - Display Audit Journal Entries -- Type ZC

Since there is no equivalent command to DSPOBJD for stream files, you must issue a WRKLNK command and use option 8 to check the last change date for a file.

```
WRKLNK OBJ('/harm/cfg/pubserv.cfg')
       DETAIL(*EXTENDED)
       DSPOPT(*ALL)
```

```
                              Work with Object Links

Directory  . . . . :   /harm/cfg

Type options, press Enter.
  3=Copy   4=Remove   5=Next level   7=Rename   8=Display attributes
  11=Change current directory ...

Opt   Object link            Type             Attribute    Text
 8    pubserv.cfg            STMF
```

Figure 224. Work with Object Links

```
                              Display Attributes


Object . . . . . :   /harm/cfg/pubserv.cfg

Type . . . . . :   STMF

Owner  . . . . . :   A960321WM
System object is on  . . . . . :   Local
Auxiliary storage pool . . . . . :   1
  Object overflowed  . . . . . :   No

Code page  . . . . . :   437
Hidden file  . . . . . :   No
PC system file . . . . . :   No
Read only  . . . . . :   No

Need to archive (PC) . . . . . :   Yes
Need to archive (AS/400) . . . . . :   Yes

Last access date/time  . . . . . :   01/17/97  18:18:52
Data change date/time  . . . . . :   12/13/96  13:06:33
Attribute change date/time . . . . . :   01/20/97  11:10:44

Size of object data in bytes . . . . . :   600
Allocated size of object . . . . . :   4096
Size of extended attributes  . . . . . :   0

Auditing value . . . . . :   *CHANGE

Object domain  . . . . . :   *SYSTEM

Number of hard links . . . . . :   1
```

Figure 225. Display Attributes

9.5.5 TCP/IP Configuration

Verify that unwanted TCP/IP servers are not started. You can disable any TCP/IP server application from being automatically started, by using the following commands:

```
CHGSNMPA AUTOSTART(*NO)
CHGTELA AUTOSTART(*NO)
CHGFTPA AUTOSTART(*NO)
CHGSMTPA AUTOSTART(*NO)
CHGLPDA AUTOSTART(*NO)
CHGHTTPA AUTOSTART(*NO)
CHGWSGA AUTOSTART(*NO)
CHGPOPA AUTOSTART(*NO)
```

You can also use the following command to end unwanted servers that are already started:

```
ENDTCPSVR SERVER(server-name)
```

9.5.6 CGI Programs

Verify that your CGI programs were not changed.

If you start the QAUDJRN journal and select *CHANGE as the object auditing value in the QAUDLVL system value, you can search for a ZC journal entry (an object was changed). You can find the user profile that changed the file, user and device name, job name, and the date and time a change was made. You can use the following command:

```
DSPAUDJRNE ENTTPY(ZC)
```

```
                                  QUERY NAME . . . . . QSECZC
                                  LIBRARY NAME . . . . QSYS
                                  FILE         LIBRARY      MEMBER       FORMAT
                                  QASYZCJE     QTEMP        QASYZCJE     QASYZCJE
                                  DATE . . . . . . . . 01/20/97
                                  TIME . . . . . . . . 14:40:14


01/20/97  14:40:14                                                          PAGE    1
   USER       OBJECT     LIBRARY    OBJECT   JOB        JOB        JOB    DATE       TIME
   PROFILE    NAME       NAME       TYPE     NAME       USER       NUMBER
ZC ADAN       QATMHTTPC  QUSRSYS    *FILE    QPADEV0003 ADAN       013941 010797 13:28:22
ZC BRSMITH    QATMHTTPC  QUSRSYS    *FILE    QPADEV0003 BRSMITH    013966 011097 15:09:26
ZC BRSMITH    QATMHTTPC  QUSRSYS    *FILE    QPADEV0003 BRSMITH    013966 011097 15:26:27
ZC A960321WM  *N         *N         *STMF    QPADEV0003 A960321WM  014236 012097 11:11:43
* * *  E N D  O F  R E P O R T  * * *
```

Figure 226. DSPAUDJRNE Report - Display Audit Journal Entries -- Type ZC

Appendix A. Special Notices

This publication is intended to help webmasters, system administrators, security administrators and other personnel involved in planning, configuring, or administering services on AS/400 systems connected to the Internet.

The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM TCP/IP Connectivity Utilities for AS/400 and IBM Operating System/400. See the PUBLICATIONS section of the IBM Programming Announcement for IBM TCP/IP Connectivity Utilities for AS/400 and IBM Operating System/400 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific

information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	AS/400
Client Access	DB2/400
DRDA	IBM
OfficeVision	OfficeVision/400
OS/2	OS/400
SQL/400	400

The following terms are trademarks of other companies:

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix B. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 249.

- *WWW.Security*, SG24-4564-00
- *Cool Title About the AS/400 and Internet*, SG24-4815-01

B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

B.3 Other Publications

These publications are also relevant as further information sources:

- *TCP/IP Configuration and Reference*, SC41-3420-04
- *Tips and Tools for Securing Your AS/400*, SC41-3300-01
- *AS/400 Security - Reference*, SC41-4302-01
- *SECUREWAY: AS/400 and the Internet*, G325-6321-00
- *AS/400 Crytographic Support/400 User*, SC41-8080-00
- *Common Cryptographic Architecture Services/400 Installation and Operator*, SC41-0102-02
- *AS/400 System API Reference*, SC41-4801-01
- *OS/400 Security - Reference V3R7*, SC41-4301-01
- *Backup and Recovery - Advanced*, SC41-4305-01

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 AllerOS/400d Denmark

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Index

Special Characters

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

ITSO Redbook Evaluation

AS/400 Internet Security: Securing Your AS/400 from HARM in the Internet
SG24-4929-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redeval@vnet.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

_____

_____

_____

_____

_____

This soft copy for use by IBM employees only.

Printed in U.S.A.