Los programas de salida son esenciales para proteger tu iSeries 400

Por
Wayne O. Evans
Wayne O. Evans Consulting, Inc

Contenidos

Tablas y Gráficos

Los programas de salida (Exit-Programs) son Esenciales

Confiar en la seguridad a través de menú para limitar las acciones del usuario no es adecuado para proteger los datos de producción en el actual AS/400. La seguridad por Menú no previene al usuario de un sistema remoto de ejecutar un FTP (file transfer protocol) o una transferencia de fichero Client Access para cargar o descargar ficheros de producción. Los usuarios remotos pueden ejecutar comandos CL incluso cuando el perfil de usuario restringe la introducción de comandos.

Los programas de salidas son pues esenciales para controlar las acciones de usuarios remotos. Este documento describe cómo los programas de salida pueden constituir una efectiva capa de protección para tus datos de producción. Después de leer este artículo entenderás :

- *El porqué los programas de salida son esenciales para proteger los datos.*
- *Cómo se identifican los programas de salida.*
- *Cómo implementar programas de salida simples para limitar las acciones de usuarios remotos.*
- *Cómo el "PentaSafe Remote Request Management" puede resolver los requisitos de tu programa de salida.*

Introducción

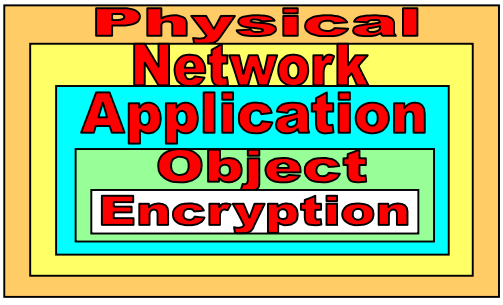
Como está organizado este Documento

Esta introducción te facilita un trasfondo para entender el porqué se necesitan los programas de salida y como éstos te ofrecen una protección adicional. La sección de detalles técnicos dará sugerencias para los gestores del sistema en la aplicación de programas de salida incluyendo un ejemplo de programa. La sección final describe la función específica de Pentasafe llamada "Remote Request Management". La "Remote Request Management" permite al administrador del sistema definir una política (reglas) para aquellas peticiones remotas que serán aceptadas o rechazadas. El "Remote System management" puede fácilmente autorizar a los usuarios a los datos que deben ser accedidos mientras que puede rechazar las peticiones que no se permiten.

Capas de protección

La protección de la información puede entenderse como distintas capas de protección parecidas a las distintas capas de una cebolla. Esta protección en capas dificultaría que un hacker pueda ganar acceso no autorizado a la información. Si un usuario no autorizado (hacker) transige una de las capas, se enfrentará con otra protección en otra capa distinta.

Figura 0 Capas de Protección Información



La Figura 0 muestra las capas de protección utilizadas para proteger los datos. Las distintas aplicaciones de seguridad seleccionarán las siguientes capas de protección .

- La capa física requiere que el usuario tenga algún impedimento físico antes de conseguir el acceso como una llave, control inteligente o biométrico (huella dactilar, scanner de retina).

- La protección de la capa de red utiliza la tecnología de encriptación para proteger las transmisiones de datos. Esta capa evita la revelación de la información transmitida mediante la encriptación y, los códigos de autentificación de mensajes previenen a su vez de la revelación y la falsificación.
- La capa de aplicación incluye programas como seguridad por menú y controles de la aplicación para ocultar la información y restringir funciones basadas en el acceso de usuario. Los programas de salida están incluidos en la capa de aplicación porque restringen las acciones y los objetos a los que un usuario puede acceder. Los programas de salida pueden entenderse como una seguridad por menú extendida a los usuarios remotos del sistema.
- La protección de la capa de objeto son los controles de acceso tradicionales incorporados en el AS/400. Este control es forzado por el sistema en cada petición para acceder a los datos.
- La capa de encriptación protege datos altamente sensibles a través de su almacenamiento en un formato que necesita de la utilización de claves de encriptación para descubrir el contenido de esta información.

La Capa de Aplicación

La protección de la Capa de Aplicación se implementa por controles forzados por los programas y menús. Los programas permiten a los usuarios autorizados acceder a funciones y datos mientras que los no autorizados no disponen de acceso. Esto se ilustra con el menú de seguridad. El Menu security limita las acciones del usuario a las opciones que se le permiten en sus menús de programa. Normalmente en el menú de seguridad los usuarios no disponen de acceso a la línea de comandos. La seguridad por Menú es ampliamente utilizada en las instalaciones de AS/400 para proteger la información y dar a los usuarios un interface simple para acceder a la información. Parecido a la seguridad por menú, los programas de salida pueden limitar a una pantalla las peticiones de usuarios remotos. Los programas de salida pueden aceptar o rechazar las peticiones para una función y pueden usar los siguientes criterios para aceptar o rechazar una petición :

- Operación que se intenta (comando remoto , fichero cargado o descargado)
- Objeto y/o biblioteca que está siendo accedidos,
- Hora del día
- Dirección de Internet del usuario remoto

El uso extendido ha aumentado los riesgos potenciales

Muchas instalaciones dependían de la seguridad por Menú para proteger sus datos cuando el único método de acceso eran terminales dependientes del host (terminales tontos).Cuando el acceso de los usuario al sistema estaba limitado a esos dispositivos con funciones fijas , la seguridad por menú era un método adecuado para las acciones de usuario. El usuario de PC de hoy en día dispone de métodos de acceso que no existen en los terminales tontos.

Los usuarios de Client Access disponen de un icono al que deben hacer click para activar las operaciones de transferencia de ficheros (carga y descarga). El interface de usuario del Client Access a la transferencia de ficheros es amigable, donde el sistema muestra los nombres de usuario para los ficheros autorizados y bibliotecas. Un usuario curioso no necesita conocimientos de nombres de objeto o interfaces de sistema y rápidamente podría cargar o descargar datos de producción. Traspasar la seguridad de menú que intenta limitar qué opciones puede un usuario seleccionar es relativamente sencillo para los usuarios de PCs o otros sistemas remotos

El protocolo preferido para el AS/400 de hoy en día es el TCP/IP. TCP/IP le da al usuario la capacidad de utilizar FTP para cargar y descargar ficheros de producción y/o ejecutar comandos CL en tu AS/400. Estos usuarios remotos no están restringidos por controles de programa y tienen el potencial para modificar y/o borrar los ficheros de datos de producción . De la misma manera que en la transferencia de ficheros de Client Access; el usuario de FTP puede obtener

listas de ficheros que pueden ser transferidos. El interface FTP no es tan amigable como la transferencia de ficheros del Client Access, pero existen interfaces fáciles de usar para la transferencia de ficheros para hacer que el FTP sea tan sencillo de utilizar como la transferencia de ficheros del Client Access .
Cuando se usa la seguridad por menú , la capacidad de limitar (el parámetro LMTCPB) en el perfil de usuario puede restringir la utilización de comandos CL. No obstante, ambos, tanto el usuario de Client Access como el de FTP , pueden emitir comandos CL al AS/400.

Por si los riegos de seguridad creados por el FTP o Client Access no fueran suficientes, la Gestión de Datos Distribuida (Distributed Data Management (DDM)) representa un tercer método para que los usuarios accedan a los ficheros de datos. La DDM permite un programa en sistemas remotos (PC, S/36, S/38, o AS/400) que lee y escribe los ficheros en un sistema de destino . Los usuarios remotos en el sistema de origen inician peticiones al sistema de destino donde los datos se almacenan. Al igual que Client Access y FTP, DDM también permite la ejecución de comandos desde el acceso local a los ficheros. El comando CL SBMRMTCMD (submit remote command) es un interface CL para la ejecución de comandos desde sistemas remotos.

El AS/400 es una plataforma segura

La arquitectura de seguridad del AS/400 es una de las mejores de los sistemas disponibles comercialmente. La seguridad del AS/400 se aplica por debajo del interface de máquina de forma que el AS/400 no está sometido al potencial de hackers que modifiquen las funciones de control de la seguridad. Pero tú podrías también tener las mejores cerraduras instaladas en tu casa y fallar en utilizarlas o dejar la puerta abierta , de forma que disponer de la mejor protección no te mantiene a salvo de un intruso. De la misma manera, respecto a la seguridad del AS/400, la seguridad forzada por el sistema es excelente pero a menudo las instalaciones fallan en la utilización de la protección disponible o dejan abiertas puertas traseras.

La capa de protección de la seguridad del AS/400 es tan fuerte que una instalación puede usar la seguridad del AS/400 para limitar el acceso a los datos de manera que los programas de salida no serían necesarios. No obstante, la aplicación de la seguridad en la práctica del AS/400 hace que muchos de los controles de seguridad no sean efectivos. Los usuarios habituales dispondrán de acceso a los datos porque los necesitan para ejecutar los programas de producción. Algunos paquetes de aplicaciones muy populares recomiendan asignar a los usuarios perfiles de grupo que sean los propietarios de los datos de producción. Cuando los usuarios son miembros de un perfil de grupo que es dueño de los datos de producción disponen de acceso *ALL a los datos. La pertenencia a este grupo tiene el efecto de eliminar la protección de la capa de objetos. Muchas instalaciones de AS/400 han fallado donde los usuarios gozaban de demasiado acceso a los datos. Esto ocurre cuando la autorización pública para los objetos es demasiado amplia o demasiados usuarios disponen de permiso especial *ALLOBJ .

En vez de corregir el débil diseño de seguridad de los sistemas, muchos administradores de sistema utilizan los programas de salida para limitar las acciones de usuario desde sistemas remotos. Las buenas noticias son que los programas de salida son un método efectivo para controlar las acciones de usuario , incluyendo incluso a aquellos con permiso *ALLOBJ. Este documento describe cómo utilizar estos programas de salida para ejercer ese control.

Bases de los programas de salida

¿Por qué los programas de salida se llaman programas de salida?

Normalmente, el nombre de los programas describe la función que el programa realiza como programa de “nómina”, programa “de entrada de pedidos”. Esta analogía no funciona en el caso de los “Exit Programs”. El propósito de los programas de salida no es la SALIDA sino que permiten a la instalación ejecutar una función cuando se recibe una petición. La función habitual de los programas de salida es suplir la seguridad existente al determinar que peticiones son aceptadas o rechazadas. Un nombre más apropiado habría sido *supplemental security programs* (SSP), algo así como programas de seguridad suplementarios , pero el acrónimo ya estaba siendo utilizado así que se quedó en programas de salida .

Estos programas se llaman de salida porque el sistema (OS/400) **sale** a un programa de instalación ya creado , antes o después que la petición es procesada. La Figura 1 muestra el flujo desde el servidor al programa de salida y su retorno

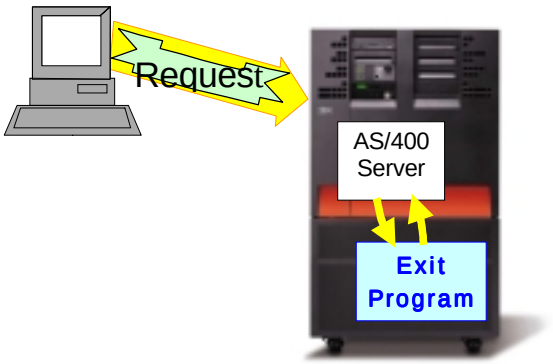


Figura 1 Flujo del Programa de Salida

1. Un usuario del sistema de origen envía una petición al sistema de destino. Esta petición es a menudo una transacción que leerá o escribirá datos o lanzará un comando al sistema destino.
2. El OS/400 reconoce la petición y la dirige al servidor.
3. Si el servidor detecta un programa de salida para esa petición , se llama al programa pasándole los parámetros de describen la petición. Si no existe programa de salida, el paso 4 se omite.
4. El programa de salida analiza los parámetros pasados por el servidor para determinar si la petición debe o ser aceptada. El programa de salida configura un código que devuelve para admitir o rechazar la petición antes de devolverle el control al servidor.
5. El código devuelto desde el programa de salida le dice al servidor que procese o rechace la petición. El programa de salida puede aceptar la petición pero el servidor puede no estar autorizado o no hallar las condiciones y provocar que la petición finalmente se rechace. Los controles del OS/400 a nivel de la seguridad de objetos son forzosamente independientes del programa de salida.

¿Dónde están los programas de salida documentados?

La documentación de IBM sobre programas de salida se muestra en la Tabla 1. No existe una fuente única donde los programas de salida estén documentados.
Una búsqueda en la biblioteca on-line de IBM por “exit programs” nos devuelve más de 360 referencias. Estas referencias de IBM son muy interesantes en la descripción de parámetros que se

Tabla 1 Documentación IBM sobre Exit Points

Client Access (File transfer, ODBC)	
AS/400 Client Access Host Servers	SC41-5740
Distributed Data Management (DDM, remote commands)	
AS/400 Distributed Data Management	SC41-5307
Internet (Telnet, FTP)	
TCP/IP Configuration and Reference	SC41-5420
Security and Other Application Program Interfaces	
System API Reference Security APIs	SC41-5872
System API Reference	SC41-5801

traspasan y cómo el servidor interpreta los códigos que le son devueltos.

No obstante una de las cuestiones más críticas no está documentada demasiado bien.

¿Como sé yo qué salidas están implicadas en una transacción?

Los manuales de IBM no tiene una referencia tipo “dónde se usan ”y necesitas saber qué salidas se invocan para cada tipo de transacción. Para responder a esta pregunta debes investigar un poco . Una de las técnicas es escribir un programa de salida simple como el que se muestra en la Figura 2 Programa de salida para grabar el uso de salidas . Cambia el tipo de campo en el comando SNDJRNE para identificar las diferentes salidas y grabar el programa en todas las salidas sospechosas.

```
PGM   PARM(&RC &STRU)
      DCL       VAR(&RC) TYPE(*CHAR) LEN(1)
      DCL       VAR(&STRU) TYPE(*CHAR) LEN(80)
      MONMSG    MSGID(CPF0000) EXEC(GOTO CMDLBL(EXIT))
      CHGVAR    &RC    VALUE('1') /* Allow requests */
      SNDJRNE   JRN(QAUDJRN) TYPE(X1) ENTDTA(&STRU)
EXIT:ENDPGM
```

Figura 2 Programa de salida para grabar el uso de salidas

El capítulo 4 de la documentación de PentaSafe PSSecure-400 tiene las mejores referencias cruzadas de funciones para programas de salida que he podido encontrar, pero incluso ese documento es difícil de seguir a menos que estés familiarizado con los detalles de los programas de salida.

¿Cómo grabar un programa de salida?

Existen 2 dos formas de grabar un programa de salida, usando atributos de red o la utilidad de registro. Esta utilidad fue introducida con posterioridad y tiene más flexibilidad que los atributos de red. Sin embargo, los atributos de red son importantes porque son la única forma de especificar programas de salida para DDM.

Atributos de red DDMACC y PCSACC

Cuando en un principio se introdujeron los programas de salida en el S/38, había sólo unas pocas salidas de forma que IBM creó dos atributos de red

- DDMACC – Distributed Data Management Access
- PCSACC – PC Support Access (Ahora llamado Client Access)

Estos atributos de red especifican un nombre de programa cualificado en el comando CHGNETA. Un usuario con permiso especial *ALLOBJ puede especificar los programas de salida utilizando el siguiente comando.

```
CHGNETA      DDMACC(EXITLIB/EXIT2)     PCSACC(EXITLIB/EXIT2)
```

Como el sistema crecía, el número de puntos de salida se incrementó , así que mucha peticiones utilizaban el mismo punto de salida. Esto llevó a que el programa de salida se llamara en ocasiones cuando no era requerido y en otras incrementaba el consumo del sistema. La utilidad de registro es el único lugar donde pueden definirse nuevos puntos de salida. Los atributos de red se mantienen sólo para la compatibilidad de release a release.

La utilidad de registro.

La utilidad de registro se invoca a través del comando WRKREGINF (Work Registration Information).

Existen 56 puntos de salida definidos en la Version 4 Release 4. El número de exit points ha crecido con cada release. La utilidad de registro puede incluir tanto puntos de salida de IBM como de otros software.

A pesar de que el número de puntos de salida ha hecho la utilidad de registro más compleja, la idea es similar a la de los atributos de red. La utilidad de registro, así, puede verse como un depósito de nombres de programas de salida.

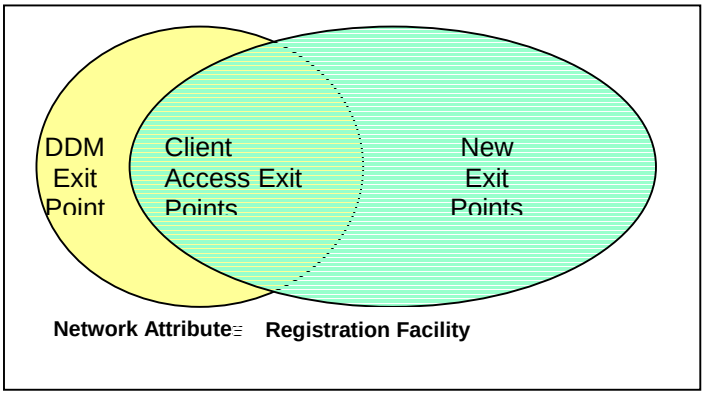


Figura 3 Métodos para Registrar Exit Programs

Como se muestra en la Figura 3 , la función de registro de atributos de red de los programas de salida y la utilidad de registro se superponen. Las salidas del Client Access pueden especificarse utilizando cualquier técnica. Pero la utilidad de registro debe usarse porque existen salidas para operaciones de transferencia de ficheros que sólo pueden especificarse usando la utilidad de registro.

Escribir Programas de Salida no es Sencillo

No es difícil escribir programas de salida simples, pero para escribir un programa de salida que sea flexible y que permite múltiples controles ,se complica mucho. Simplemente el determinar qué salidas invocan diferentes transacciones puede implicar un gran esfuerzo. La próxima sección presenta más información para aquellos lectores a los que les gusten los detalles técnicos.Te recomiendo seriamente que consideres la flexibilidad PentaSafe Remote Request Management en la página 14 antes de invertir tanto esfuerzo como el que se requiere para escribir un programa de salida complejo.

Detalles Técnicos de los Programas de Salida

Esta sección puede ser prescindible para aquellos lectores que quieran un idea general de los programas de salida. El fundamento de esta sección son los detalles de programación para los que deseen escribir sus programas de salida.

Muestra de Programas de Salida

Creo que la mejor forma de programar es copiar otro programa y modificarlo para la función específica. Con esta idea en mente, esta sección presenta varios programas de salida cada uno con más detalles y una mayor complejidad que el lector podrá modificar para ajustarlos a su situación específica.

Parámetros pasados a los programas de salida.

Los parámetros que se pasan a los programas de salida están descritos en la documentación de IBM. Los programas de salida reciben 2 parámetros :

- Parámetro 1 – Código de retorno
  Para aceptar la petición, configura el código de respuesta como '1'. Otros valores rechazarán la petición. Yo acostumbro a utilizar '0' pata rechazar la petición.
- Parámetro 2 - Estructura que describe la petición.

Nombre	Tipo de dato	Descripción
Perfil de usuario	Carácter 10	Nombre del perfil de usuario que inició la petición.
Identificador de Servidor	Carácter 10	El nombre del servidor *DDM – Distributed Data Management *LMSR – Licencia de gestión*TFRFCL – fichero transferido *FLRSRV -Carpetas compartidas tipo 2 *MSGFCL - mensajes*TRFCTL- fichero transferido *DATAQSRV - Data Queue *VPRT – Impresión vrtual
Función solicitada	Carácter 10	Tipo de función que el usuario ha pedido.
Datos variables	Carácter *	Información que variará según el tipo de servidor.

Prevención de comandos remotos.

El siguiente programa evita del uso de comandos remotos desde DDM y Client Access.
La función requerida por comandos remotos es 'COMMAND' así que este programa rechazará todas las peticiones para la ejecución remota de comandos.

1. Crear el programa llamado EXIT0 desde Figura 4 Prevenir Comandos Remotos .
   Usar el siguiente comando para crear el programa

```
CRTCLPGM  EXITLIB/EXIT0 SRCFILE(   )
```

Figura 4 Prevenir Comandos Remotos

```
PGM         PARM(&RTNCODE &DATA)
   DCL         &DATA          *CHAR  30
   DCL         &RTNCODE *CHAR    1
   DCL         &FUNC          *CHAR  10
   CHGVAR      &FUNC (%SST(&DATA 21 10))
   IF  (&FUNC = 'COMMAND   ') +
    THEN(  CHGVAR   &RTNCODE  '0')
    ELSE     CHGVAR &RTNCODE  '1'

ENDPGM
```

2. Como usuario con permiso *ALLOBJ , grabar el punto de salida al configurar los atributos de red usando :

```
CHGNETA     DDMACC(EXITLIB/EXIT0)
```

Prevenir comandos remotos y la Carga de Ficheros

El siguiente programa es un refinamiento del programa que mostramos en Figura 4. Este programa previene del uso de comandos remotos desde DDM y Client Access. Restringe además la carga de ficheros para todos los usuarios excepto para el perfil de usuario llamado POWERUSER. La última línea del programa ilustra como grabar las acciones del usuario escribiendo las entradas en el journal de auditoría.

1. Crear los comandos del programa llamado EXIT1 desde Figura 5 Prevenir Comandos Remotos y Carga de Ficheros. Cuando el programa se crea se configura para adoptar la autorización del propietario del sistema para evitar violaciones de permisos al escribir en el journal de auditoría.
   ```
   CRTCLPGM  EXITLIB/EXIT1 SRCFILE( )  USRPRF(*OWNER)
   ```
2. El programa escribe en el journal de auditoría de forma que el propietario del programa es cambiado de manera que el programa adopta la autorización que necesita para escribir en el journal.
   ```
   CHGOBJOWN OBJ(EXITLIB/EXIT1)
                              OBJTYPE(*PGM) NEWOWN(QSECOFR)
   ```
3. El programa controla las funciones DDM y Client Access de manera que los atributos de red se modifican para especificar el mismo programa de salida utilizando el siguiente commando.
   ```
   CHGNETA   DDMACC(EXITLIB/EXIT1)  PCSACC(EXITLIB/EXIT1)
   ```

Figura 5 Prevenir Comandos Remotos y Carga de Ficheros.

```
/*****************************************************/
/* Installation instructions                         */
/* 1. Compile program                                */
/*           CRTCLPGM  PGM(EXITLIB/EXIT1)            */
/*                     SRCFILE( ) USRPRF(*OWNER)     */
/* 2. Change owner of the program to user QSECOFR.   */
/*    Adopted authority allows the program sending   */
/*    to the audit journal                           */
/*         CHGOBJOWN OBJ(EXITLIB/EXIT1)              */
/*                   OBJTYPE(*PGM) NEWOWN(QSECOFR)   */
/* 3  Name the exit program in network attributes    */
/*         CHGNETA   DDMACC(EXITLIB/EXIT1)           */
/*                   PCSACC(EXITLIB/EXIT1)           */
/*                                                   */
/* The audit journal QAUDJRN entries created are:    */
/*   'X1' = Requests that are allowed                */
/*   'X0' = Requests that are rejected               */
/*****************************************************/
PGM  (&RC  &STRU )
  DCL        &RC    *CHAR   1  /*Return    1=allow  */
```

```
                                /*          0=prevent*/
  DCL        &STRU  *CHAR 200  /*Request description*/
  DCL        &USER  *CHAR  10  /*User profile name  */
  DCL        &APP1  *CHAR  10  /*Requested function */
  DCL        &APP2  *CHAR  10  /*Sub function       */
  DCL        &TYPE  *CHAR   2  /*Journal entry type */
   MONMSG    CPF0000  EXE(GOTO EXIT)  /*If error exit*/
   CHGVAR   &RC    '1'                /*Allow request*/
   CHGVAR   &USER   %SST(&STRU  1 10) /*Get user     */
   CHGVAR   &APP1   %SST(&STRU 11 10) /*Get appl     */
   CHGVAR   &APP2   %SST(&STRU 21 10) /*Get function */
  /*Do not log IBM request to check license  */
  IF (&APP1  =   '*LMSRV')  GOTO EXIT
           IF  &USER = 'POWERUSER ‘))  GOTO LOG
        /* Prevent use of remote commands          */
  IF  (&APP1 = '*DDM' *AND &APP2 = 'COMMAND') +
         CHGVAR   &RC '0'   /* Prevent the request  */
  ELSE  /*  Prevent file upload from PC users      */
         /*  File download to PC is not prevented   */
    IF   (&APP1 = '*TFRFCTL' *AND &APP2 = 'REPLACE') +
         CHGVAR   &RC '0'   /* Prevent the request */
        /*  Log request in the audit journal       */
LOG:CHGVAR &TYPE  ( 'X' *CAT &RC)
    SNDJRNE  QAUDJRN  TYPE(&TYPE)  &ENTDTA(&STRU)
EXIT:ENDPGM
```

Restringir la transferencia de ficheros a bibliotecas específicas

El siguiente programa restringirá las peticiones de transferencia de ficheros a bibliotecas específicas UP_LIB y DOWN_LIB. Esta es una técnica excelente para limitar a qué bibliotecas se puede acceder con transferencia. El administrador del sistema controla los archivos que se autoriza que sean transferidos a través de la designación de biblioteca, en vez de para cada fichero.

1. Crear el programa llamado EXIT2 desde Figura 6 Restringir Transferencia de ficheros a Bibliot. específicas .
   ```
   CRTCLPGM    EXITLIB/EXIT2 SRCFILE(  )  USRPRF(*OWNER)
   ```
2. El programa escribe en el journal de manera que se cambia el propietario del programa para que adopte la autorización que necesita para escribir en el journal.
   ```
   CHGOBJOWN   OBJ(EXITLIB/EXIT2)
               OBJTYPE(*PGM) NEWOWN(QSECOFR)
   ```
3. El programa controla las funciones de transferencia de archivos de Client Access. Utilizar la opción de registro para registrar el programa de salida.
   ```
   ADDEXITPGM  EXITPNT(QIBM_QTF_TRANSFER)
               FORMAT(TRAN0100)  PGMNBR(1)
               PGM(EXITLIB/EXIT2)
               TEXT('Limit transfer to specific libraries')
   ```
4. El atributo de red debe configurarse para indicar que la utilidad de registro está siendo utilizada ,con el siguiente comando.
   ```
   CHGNETA     PCSACC(*REGFAC)
   ```

Figura 6 Restringir Transferencia de ficheros a Bibliot. específicas

```
/*============================================================*/
/* To compile:                                                */
/*           CRTCLPGM   PGM(EXITLIB/EXIT2) +                  */
/*                      USRPRF(*OWNER) SRCFILE( )             */
```

```
/* installation instructions:                                    */
/*   1. Compile program                                          */
/*   2. Change owner of the program to user QSECOFR.             */
/*      Adopted authority allows the program sending             */
/*      to the audit journal                                     */
/*           CHGOBJOWN  OBJ(EXITLIB/EXIT2) OBJTYPE(*PGM) +      */
/*                      NEWOWN(QSECOFR)                         */
/*   3. Name the exit program in registration facility          */
/*           ADDEXITPGM EXITPNT(QIBM_QTF_TRANSFER) +            */
/*                      FORMAT(TRAN0100) PGMNBR(1)+             */
/*                      PGM(EXITLIB/EXIT2) +                    */
/*                      text('limit to specific libraries') */
/*   4. Set registration facility in the network attribute     */
/*           CHGNETA    PCSACC(*REGFAC)                         */
/* The request is recorded in the audit journal                 */
/* The audit journal QAUDJRN entries created are:               */
/*   'X1' = requests that are allowed                           */
/*   'X0' = requests that are rejected                          */
/*============================================================*/
PGM        PARM(&RC &STRU)
  DCL      VAR(&RC) TYPE(*CHAR) LEN(1)   /*1=allow 0=prevent*/
  DCL      VAR(&STRU) TYPE(*CHAR) LEN(80) /* request  description */
DCL  VAR(&USER)  TYPE(*CHAR) LEN(10) /* user profile  */
DCL  VAR(&APP1)  TYPE(*CHAR) LEN(10) /* function      */
DCL  VAR(&APP2)  TYPE(*CHAR) LEN(10) /* sub function  */
DCL  VAR(&TFOBJ) TYPE(*CHAR) LEN(10) /* file  name    */
DCL  VAR(&TFLIB) TYPE(*CHAR) LEN(10) /*library        */
DCL  VAR(&TFMBR) TYPE(*CHAR) LEN(10) /* member        */
DCL  VAR(&TFFMT) TYPE(*CHAR) LEN(10) /* format        */
DCL  VAR(&TYPE)  TYPE(*CHAR) LEN(2)  /* journal  type */
MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(EXIT))
CHGVAR     VAR(&RC) VALUE('1') /* set return code to +
             allow request unless rejected by program */
CHGVAR     VAR(&USER) VALUE(%SST(&STRU 1 10)) /*  user */
CHGVAR    VAR(&APP2) VALUE(%SST(&STRU 21 10)) /*function */
CHGVAR     VAR(&TFOBJ) VALUE(%SST(&STRU 31 10)) /* file */
CHGVAR     VAR(&TFLIB) VALUE(%SST(&STRU 41 10)) /* libr */
CHGVAR     VAR(&TFMBR) VALUE(%SST(&STRU 51 10)) /* memb */
CHGVAR     VAR(&TFFMT) VALUE(%SST(&STRU 61 10)) /* Format */
   /*****************************************/
/*  Prevent file upload from PC users      */
/*       except in the UP_LIB library      */
/*  prevent  download to PC                */
/*       except in the DOWN_LIB library    */
/*****************************************/
IF   COND(&APP2 *EQ 'REPLACE') THEN(DO)
 IF   COND(&TFLIB *NE 'UP_LIB    ') THEN( +
       CHGVAR &RC) '0') /*prevent request*/
ENDDO
IF   COND(&APP2 *EQ 'SELECT') THEN(DO)
 IF   COND(&TFLIB *NE 'DOWN_LIB  ') THEN( +
      CHGVAR &RC) '0') /*prevent request*/
ENDDO
 /*****************************************/
/* Log request in the audit journal       */
 /*****************************************/
LOG:  CHGVAR VAR(&TYPE) VALUE('X' *CAT &RC)
  SNDJRNE JRN(QAUDJRN) TYPE(&TYPE)  ENTDTA(&STRU)
Exit:ENDPGM
```

Salida de Cliente Optimizada

Entonces descubrí que el programa de salida EXIT2 sólo restringía a usuarios que estaban utilizando las versiones originales (Windows 3.1 and DOS) de Client access. Versiones posteriores de Client access usaban un servidor distinto y no llamaban a los programas de salida existentes. Después de un poco de investigación, escribí un programa EXIT2A para el cliente optimizado. El nombre del archivo que es transferido no se pasa en la estructura. En vez de eso, el programa comprueba en la petición de archivo para buscar el nombre del archivo que está siendo transferido.
Los parámetros de la transferencia pasan la sentencia SQL que representa la petición de transferencia. Sería posible analizando sintácticamente la sentencia SQL conocer más sobre la transacción, pero la complejidad de esto se escapa del enfoque de este documento.

Los pasos para añadir el programa de salida son :

1. Crear el programa llamado EXIT2A desde Figura 7 Restringir Transferencia Archivos a Bibliot. específicas para clientes optimizados.
   Utilizar el siguiente commando para crear el programa
   ```
   CRTCLPGM  EXITLIB/EXIT2A SRCFILE(  )  USRPRF(*OWNER)
   ```
2. El programa escribe en el journal de manera que se cambia el propietario del programa para que adopte el permiso necesario para escribir en el journal de auditoría.
   ```
   CHGOBJOWN   OBJ(EXITLIB/EXIT2A)
               OBJTYPE(*PGM) NEWOWN(QSECOFR)
   ```
3. El programa controla las funciones de transferencia de ficheros Client Access .
   Utilizar la utilidad de registro para registrar el programa de salida.
   ```
   ADDEXITPGM  EXITPNT(QIBM_QZDA_NBR1)
               FORMAT(ZDAD0100)  PGMNBR(1)
               PGM(EXITLIB/EXIT2A)  REPLACE(*NO)
               TEXT('Limit transfer to specific libraries')
   ```

Figura 7 Restringir Transferencia Archivos a Bibliot. específicas para clientes optimizados.

```
/*================================================*/
/* 1. Compile program                             */
/* CRTCLPGM  PGM(EXITLIB/EXIT2A) USRPRF(*OWNER)   */
/* 2. Change owner of the program to user QSECOFR.*/
/*    Adopted authority allows the program sending*/
/*    to the audit journal                        */
/*    CHGOBJOWN  OBJ(EXITLIB/EXIT2A) OBJTYPE(*PGM) +*/
/*               NEWOWN(QSECOFR)                  */
/* 3. Name the exit program in registration facility*/
/*    ADDEXITPGM EXITPNT(QIBM_QZDA_NBR1   ) +     */
/*               FORMAT(ZDAD0100) PGMNBR(1)+      */
/*               PGM(EXITLIB/EXIT2A) REPLACE(*NO) +*/
/*               TEXT('limit to specific libraries')*/
/* The request is recorded in the audit journal   */
/* The audit journal QAUDJRN entries created are: */
/* 'Z1'=requests allowed   Z0' = requests rejected*/
/*================================================*/
 PGM        PARM(&RC &REQUEST)
  DCL        VAR(&RC) TYPE(*CHAR) LEN(1)
  DCL        VAR(&REQUEST) TYPE(*CHAR) LEN(700)
  DCL        VAR(&TYPE) TYPE(*CHAR) LEN(2)
  DCL &X1800 *CHAR 4 VALUE(X'00001800') /*create database  */
  DCL &X1801 *CHAR 4 VALUE(X'00001801') /*create src file  */
  DCL &X1802 *CHAR 4 VALUE(X'00001802') /*add    member    */
  DCL &X1803 *CHAR 4 VALUE(X'00001803') /*clear  member    */
  DCL &X1804 *CHAR 4 VALUE(X'00001804') /*delete member    */
  DCL &X1805 *CHAR 4 VALUE(X'00001805') /*file override    */
  DCL &X1806 *CHAR 4 VALUE(X'00001806') /*delete  override */
  DCL &X1807 *CHAR 4 VALUE(X'00001807') /*create savefile  */
```

```
  DCL &X1808 *CHAR 4 VALUE(X'00001808') /*clear  savefile  */
  DCL &X1809 *CHAR 4 VALUE(X'00001809') /*delete file      */
  /* OPTIMIZED DATABASE SERVER DECLARES */
  DCL &DBFMT) *CHAR)  8  /* format name            */
  DCL &DBFID) *CHAR)  4  /* function identifier    */
  /* FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAD0100 */
  DCL &DBFILE *CHAR 128  /* file name              */
  DCL &DBLIB  *CHAR  10  /* library name           */
  DCL &DBMBR  *CHAR  10  /* member name            */
  DCL &DBAUT  *CHAR  10  /* authority to file      */
  DCL &DBBFIL *CHAR 128  /* based on file name     */
  DCL &DBBLIB *CHAR  10  /* based on library name  */
  DCL &DBOFIL *CHAR  10  /* override file name     */
  DCL &DBOLIB *CHAR  10  /* override library name  */
  DCL &DBOMBR *CHAR  10  /* override member name   */
```

```
  MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(EXIT))
 /* allow request unless rejected by program */
  CHGVAR     VAR(&RC) VALUE('1')
  /* set variables from request description */
  CHGVAR VAR(&DBFMT)  VALUE(%SST(&REQUEST 21 8))
  CHGVAR VAR(&DBFID)  VALUE(%SST(&REQUEST 29 4))  CHGVAR
VAR(&DBFILE) VALUE(%SST(&REQUEST 33 128))
  CHGVAR VAR(&DBLIB)  VALUE(%SST(&REQUEST 161 10))
  CHGVAR VAR(&DBMBR)  VALUE(%SST(&REQUEST 171 10))
  CHGVAR VAR(&DBOFIL) VALUE(%SST(&REQUEST 329 10))
  CHGVAR VAR(&DBOLIB) VALUE(%SST(&REQUEST 339 10))
  CHGVAR VAR(&DBOMBR) VALUE(%SST(&REQUEST 349 10))
  IF COND((&DBFID = &X1805)) THEN(DO) /* OVERRIDE */
    IF COND(&DBOLIB = 'UP_LIB')   THEN(GOTO LOG)
    IF COND(&DBOLIB = 'DOWN_LIB') THEN(GOTO LOG)
    CHGVAR VAR(&RC) VALUE('0')
  ENDDO
  /* log request in the audit journal */
LOG:
  CHGVAR   VAR(&TYPE) VALUE('Z' *CAT &RC)
  SNDJRNE  JRN(QAUDJRN) TYPE(&TYPE) ENTDTA(&REQUEST)
EXIT: ENDPGM
```

Programas de salida FTP y TELNET

De manera similar a los ejemplos anteriores, pueden utilizarse otros puntos de salida para escribir programas de salida que controlen el acceso FTP y TELNET. El Manual de Configuración TCP/IP y Referencia (SC41-5420) facilita detalles sobre los parámetros que se traspasan y los códigos de retorno. Estos parámetros y códigos son un poco más complicados para estos puntos de salida.

La técnica para escribir Programas de Salida.

Pueden evaluarse diseños de programas de salida alternativos antes de intentar escribir un programa de salida.

Comparar con constante

En el programa de ejemplo EXIT1 el código compara el perfil de usuario con una constante fija (‘POWERUSER’). La lógica del programa es muy simple y fácil de seguir , pero la técnica de comparación contra una constante tiene una limitación importante. La comparación con constantes es muy eficiente en la ejecución pero implica que el programa de salida necesitará ser recompilado cuando los usuarios o los niveles de acceso cambien.

Comparar con usuarios del archivo.
Algunos programadores de programas de salida almacenarán los nombres de los usuarios y de las transacciones autorizadas en un archivo . Esta es una técnica muy útil en vez de recompilar el programa cada vez que se realiza un cambio. La limitación es que debe habilitarse algún método para actualizar los archivos de forma segura. El archivo representa funciones de control de acceso adicionales que no son obvias al fijarnos en la seguridad que buscamos.

Comparar con lista de autorizaciones.
La técnica que prefiero en vez de comprobar el acceso del usuario en un archivo es utilizar una lista de autorizaciones para almacenar los nombres de los perfiles de usuario y sus accesos. La utilización de una lista de permisos tiene estas ventajas :

El interface standard de seguridad del OS/400 se usa para mantener la lista de usuarios. Esto elimina la necesidad de escribir programas de mantenimiento de la información de acceso.

- Los accesos serán reportados usando el interface de seguridad del OS/400 y por tanto los auditores están familiarizados con la información y con los que pueden modificar esta información de acceso.
- La lista de autorizaciones puede incluir nombres de perfiles de grupo de manera que no es necesario mantenerla para usuarios individuales.
- La comprobación del acceso es muy eficiente y no necesita abrir archivos adicionales.

Figura 8 Lógica para comprobar Lista de Autorizaciones

```
IF  COND(……. ) THEN(DO)
   CHKOBJ     OBJ(QSYS/FILEREAD) +
                    OBJTYPE(*AUTL) AUT(*USE)
   MONMSG   MSGID(CPF9800) +
                       EXEC(CHGVAR &RC  '0')
   GOTO       LOG
```

PentaSafe Remote Request Management

Visión general

El Remote Request Management (RRM) de PentaSafe facilita la seguridad de la instalación para peticiones remotas. El RMM aplica el interface de programa (programas de salida y programas de gestión) para controlar el FTP, la transferencia de ficheros , y comandos de usuarios remotos. RRM se implementa usando programas de salida pero los detalles de programación están ocultos a los administradores del sistema ,lo que les permite a éstos centrarse en las reglas del negocio necesarias para asegurara lo datos del AS/400 de accesos desde sistemas remotos, en vez de preocuparse de la programación.

El Remote Request Management permite al administrador del sistema asegurar las peticiones remotas críticas , incluyendo las siguientes :

- Peticiones de Transferencia de Archivos (DDM, DRDA[1]), Client Access, FTP)
- Comandos Remotos (DDM, Client Access, FTP)
- Sign-on (Telnet y FTP)

Para las peticiones aceptadas, el administrador puede elegir cambiar (swap) el perfil de usuario cuando el servidor procesa la transacción.

Las funciones de informe del RRM incluyen la capacidad de grabar las peticiones remotas para que el manager del sistema pueda revisar la actividad de los usuarios , en caso de que fuera

[1] DRDA (Distributed Remote Data Access) supports remote system add to data. DRDA used APPC, APPN or TCP/IP communication protocols.

necesaria. Cuando la petición se rechaza, se puede enviar un mensaje de alerta . Las peticiones rechazadas se graban en un log de rechazos .

Usando Remote Request Management

RRM oculta los detalles de implementación gracias a que facilita al administrador unas opciones a través de menú para manejar las siguientes tareas :

- Instalar o Eliminar Remote Request Management
- Ejecutar el modo de recolección de datos.

Un archivo de control de las peticiones aceptadas determina que peticiones serán permitidas. La selección de peticiones es muy flexible y dependerá de los requerimientos de seguridad de acceso. Las peticiones remotas pueden aceptarse o rechazarse en función de uno o más de los criterios siguientes :

- Tipo de petición
  - Carga o Descarga
  - Comandos específicos
- Usuarios
  - Todos los usuarios
  - Nombre del usuario remoto.
  - Perfil de Grupo del Usuario Remoto.
  - Nombre de Perfil Genérico.
- Nombre del Objeto
  - Todos los objetos de una biblioteca específica.
  - Nombre Objeto Genérico
  - Nombre Objeto Específico
- Hora del día del acceso.
- Dirección IP para TELNET

Las opciones de arriba sobrepasan los requerimientos de muchas instalaciones, pero en algunos casos raros son necesarios criterios especiales. El RRM dispone de la capacidad para invocar programas de usuario para manejar necesidades adicionales.

Configuración

El soporte RRM se instala como parte del producto PSSecure de PentaSafe. Una vez en el programa, el administrador puede instalar todos los programas de salida simplemente seleccionando una opción de menú. La instalación de programas de salida específicos puede habilitarse o deshabilitarse en función de que la instalación requiera una mayor personalización. Los programas de salida se instalan con la opción de “sin restricción de transacciones” activada.

Recolección de peticiones

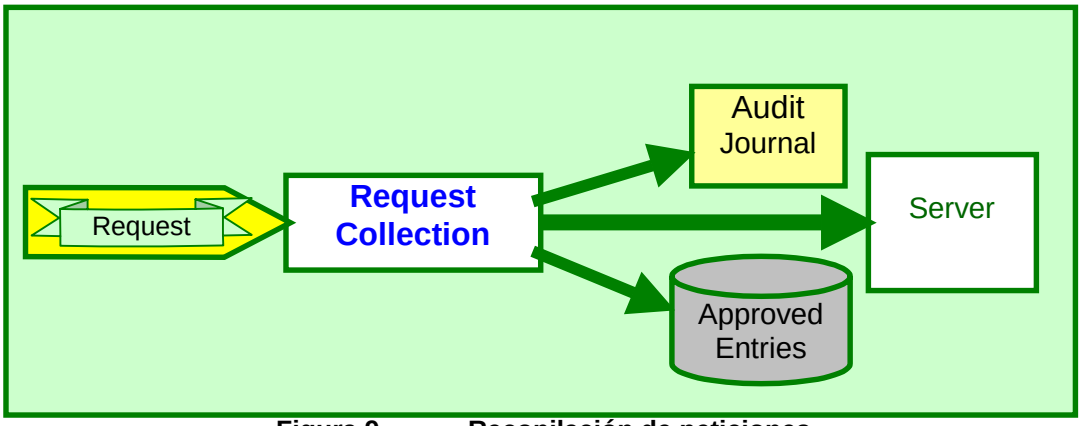


Figura 9 Recopilación de peticiones.

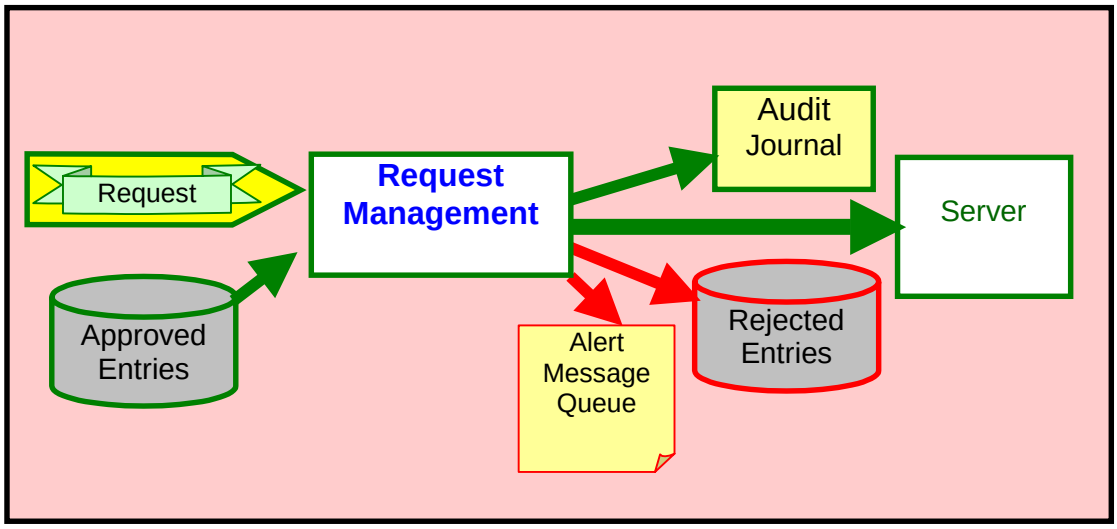
El primer paso que se recomienda es activar la Request Collection (recolección de peticiones) para que el soporte RRM monitorice las transacciones remotas. Esto le permite al manager del sistema observar las peticiones remotas que ocurren en el sistema. La Request Collection se activa con una simple opción desde la pantalla de Mantenimiento de RRM defaults como la Figura 10 Mantenimiento RRM Defaults . Mientras se recolectan estas peticiones, ninguna transacción se rechaza, lo que elimina interrupciones de usuario posibles mientras el sistema utiliza el Request Collection para observar que tipo de peticiones remotas se realizan. En la Figura 9 , se permiten todas las transacciones . El archivo de Entradas Aprobadas contiene un modelo de los tipos de transacciones que generan los usuarios del sistema. La recolección de peticiones se activa normalmente de 1 a 2 semanas de manera que la actividad normal de producción se graba en el fichero de entradas aprobadas. Ahora el sistema ya está preparado para el siguiente paso y cambiará la opción de Request Collection a N para finalizar este proceso.

Figura 10 Mantenimiento RRM Defaults

```
  TS1000C                Remote Request Management
                  M a i n t a i n   R R M   D e f a u l t s          SYSNAME

Check Group Profile Authority? . . . . . . . . . . . . . . . Y (Y/N)
Check Supplemental Group Profile Authority? . . . . . . . . . N (Y/N)
Audit remote requests? . . . . . . . . . . . . . . . . . . . . _ (Y/N/Blank)
Secure all exit points? . . . . . . . . . . . . . . . . . . . . _ (Y/N/Blank)
Number of days to retain rejected entries (0 - 999) . . . . . 045
Debug exit program ? . . . . . . . . . . . . . . . . . . . . . N (Y/N)
Collect RRM requests? (security is bypassed) . . . . . . . . . Y (Y/N)
Library for exit program shells . . . . . . . . . . . . . . . QGPL
Collection Repository? . . . . . . . . . . . . . . . . . . . . A (A/R)

Message Queues to notify if exit program fails . . . . . . . .
                                                               _____

Message Queues to notify of rejected requests  . . . . . . . .
          JIM        QSECOFR    PAC                            _____

Approve/Reject request in case exit program fails . . .        A (A/R)
Submit end Pre-start job to job queue. .   QSYSNOMAX  QSYS
TELNET . . . . . . . . Log Message . . .   Y      Log File . . Y (Y/N)
```

Gestión de Peticiones

Figura 11 Gestión de Peticiones.



El administrador del sistema puede entonces revisar los modelos de transacciones recopilados en el archivo de Entradas Aprobadas. Cualquier modelo de transacción que no debería estar permitido ,será eliminado de ese archivo. De forma que los restantes modelos de transacciones serán la actividad que se aprueba para el sistema. Se debe activar entonces la Request Management , y así solo serán aceptadas aquellas peticiones que coinciden con las del fichero de Entradas Aprobadas .Todas las otras peticiones se rechazarán y quedarán grabadas en un archivo de Entradas Rechazadas. En caso de que algunas peticiones que deberían haberse permitido se rechazaran , el administrador del sistema puede moverlas fácilmente al archivo de autorizadas de forma que las peticiones serán aceptadas. Puede diseñarse un mensaje de alerta para que se avise al administrador cuando se detecten peticiones no autorizadas. La recolección de datos en el journal es opcional y puede usarse para grabar tanto las transacciones remotas autorizadas como las rechazadas.

Resumen

El Remote Request Management ofrece la protección que necesitan los datos del AS/400.
Las transacciones (FTP, Transferencia de Archivos y comandos remotos) que traspasan la seguridad de menú pueden ser fácilmente permitidas o rechazadas por el RRM.
El RRM le permite al administrador decidir qué tipos de transacciones se van a aprobar y cuales no a partir de los datos recopilados del uso del sistema actual. Esta enfoque de "aprobación a través de ejemplos " simplifica enormemente la especificación de reglas. Además, el criterio de aprobación es flexible y cubre de sobras las necesidades de la mayoría de instalaciones AS/400
Las opciones más interesantes son que el soporte puede instalarse y desinstalarse del sistema de producción sin interrupción de la producción normal. Sólo aquellas peticiones que no sean autorizadas se rechazarán.

Conclusión

Es esencial disponer de varias capas de protección para la protección de la información. Los controles de seguridad de AS/400 son demasiado a menudo ineficaces y deben ser complementados por programas de salida para gestionar las peticiones remotas.
La implementación de programas de salida no es una tarea sencilla y eso hace recomendable la evaluación de programas de terceros como el PentaSafe Remote Request Management