PowerTech®

Developing Secure IBM i Applications

Today's Agenda

- **Introductions**
- Design and Documentation
- Application Ownership and Authority
- A Simple Security Model
- Integrity Considerations
- Resources for Security Officers
- Questions & Answers

Your Speaker

ROBIN TATAM, CISM
Director of Security Technologies

952-563-2768
robin.tatam@powertech.com

Certified for IBM | Power Systems

CISM Certified Information Security Manager

About PowerTech

PowerTech
your security expert

- Premier provider of security solutions & services
  - 18 years in the security industry as an established thought leader
  - Customers in over 70 countries, representing every industry
  - Security subject matter expert for COMMON
- Wholly-owned subsidiary of HelpSystems since 2008
- IBM Advanced Business Partner
- Member of PCI Security Standards Council
- Authorized by NASBA to issue CPE credits for security education
- Publisher of the annual "State of IBM i Security Study"

NASBA
National Association of State Boards of Accountancy

IBM Business Partner

PCI Security Standards Council™

Today's Agenda

- Introductions
- **Design and Documentation**
- Application Ownership and Authority
- A Simple Security Model
- Integrity Considerations
- Resources for Security Officers
- Questions & Answers

Design & Documentation

PowerTech
your security expert

An application's security design should be an **integral part** of the normal analysis and planning process.

The architecture should be **documented** for later reference.

Design & Documentation

PowerTech. your security expert

Security design documentation is critical for auditors, system administrators, and the “next” programmer. Include information on:

- Overview of the security architecture
- What profiles need to exist (ownership and runtime)
- Which files contain sensitive data (audited or encrypted?)
- What authorization lists are used
- How data is accessed (application programs, Query, FTP, etc.)
- How users gain access (public authority, private authority, adopted authority)
- Any special object runtime attributes (adoption, etc.)

Design & Documentation

Poor Planning Leads to Failed Execution

(and potentially unsecure applications)

Today's Agenda

- Introductions
- Design and Documentation
- **Application Ownership and Authority**
- A Simple Security Model
- Integrity Considerations
- Resources for Security Officers
- Questions & Answers

Application Ownership

Under IBM i, every object is “owned” by a profile that is initially granted *ALL access to the object.

Object ownership is assigned when the object is first created, and can be changed using the CHGOBJOWN and CHGOWN commands.

Initial ownership is claimed by the user who creates it, or the group that they belong to (depends on their profile settings).

Application Ownership

```
                         Display Object Authority

Object . . . . . . . :   PAYNEW          Owner  . . . . . . . :   PAYOWN
  Library  . . . . . :     PAYROLL       Primary group  . . . :   *NONE
Object type  . . . . :   *FILE           ASP device . . . . . :   *SYSBAS

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

                          Object
User          Group      Authority
*PUBLIC                  *EXCLUDE
PAYOWN                   *ALL
READONLY                 *USE

                                                                   Bottom
Press Enter to continue.

F3=Exit   F11=Display detail object authorities   F12=Cancel   F17=Top
F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 2005.
```

The owner is automatically granted *ALL access

Application Ownership

Consider creating a profile specifically to “own” the related application objects:

- Provides consistency
- Helps simplify save/restore operations

I recommend NOT using IBM-supplied profiles, or allowing programmers to remain the owners.

Application Ownership

The “owning” profile does not need any special authority (unless the application performs system tasks using authority adoption).

```
CRTUSRPRF  USRPRF(PAYOWN)  PASSWORD(*NONE)
SPCAUT(*NONE) INLPGM(*NONE)  INLMNU(*SIGNOFF)
LMTCPB(*YES)
```

An application build process or lifecycle manager (aka change control) can ensure correct object ownership and authority settings.

Application Ownership

It **<u>is</u>** possible to change the owner's authority so that they cannot access an object that they own!

However, ownership provides certain privileges, such as the ability to set authorities for other users—including themselves!

Application Ownership

```
                          Display Object Authority

Object . . . . . . . :   PAYNEW          Owner  . . . . . . . :   PAYOWN
  Library  . . . . . :     PAYROLL       Primary group  . . . :   *NONE
Object type  . . . . :   *FILE           ASP device . . . . . :   *SYSBAS

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

                          Object
User        Group        Authority
*PUBLIC                  *EXCLUDE
PAYOWN                   *EXCLUDE
READONLY                 *USE

                                                                   Bottom
Press Enter to continue.

F3=Exit   F11=Display detail object authorities   F12=Cancel   F17=Top
F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 2005.
```

Application Ownership

The application design should accommodate objects that are created by the users during runtime.

Typically, the application should:

- Create the new object (CRTxxx)
- Set object ownership (CHGOBJOWN)
- Establish the desired authorities (GRTOBJAUT)

Public Authority

IBM i contains a unique concept called Public Authority which is the *default* permission granted to a user who has not been granted any *explicit* authority (including *EXCLUDE).

Public authority is determined by:

- For native objects: public authority is assigned starting from the CRTxxx command
- For IFS objects: public authority is inherited from the parent directory

Public Authority

For native objects, IBM resolves the public authority setting from the command to the library description to the QCRTAUT system value.

Once the *PUBLIC authority is resolved, it's permanent—there is no dynamic link.

Public Authority

There is nothing technically wrong with the *concept* of default public authority.

DSPSYSVAL

QCRTAUT

Problems begin when the QCRTAUT system value remains at its shipped value: *CHANGE

(That's sufficient to read, change, and delete data!)

Public Authority

I recommend controlling the public authority default for each individual library.

CRTxxx
AUT(*LIBCRTAUT)

DSPLIBD
CRTAUT(*EXCLUDE)

DSPSYSVAL
QCRTAUT

This permits granular control; especially when the server contains multiple applications with varying authority requirements.

Public Authority

```
                          Display Object Authority

Object . . . . . . . :   PAYNEW          Owner  . . . . . . . :   PAYOWN
  Library  . . . . . :     PAYROLL       Primary group  . . . :   *NONE
Object type  . . . . :   *FILE           ASP device . . . . . :   *SYSBAS

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

                          Object
User         Group        Authority
*PUBLIC                   *EXCLUDE
PAYOWN                    *ALL
READONLY                  *USE

                                                                  Bottom
Press Enter to continue.

F3=Exit   F11=Display detail object authorities   F12=Cancel   F17=Top
F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 2005.
```

Every object has a default authority (*PUBLIC)

Authority Shortcuts

A user must have the required level of authority to access an object based on the requested action.

Authority is determined in the following (basic) sequence:

1. Individual User
2. Group Profile (consolidated if multiple groups)
3. *PUBLIC

Authority Shortcuts

PowerTech. your security expert

IBM i provides 4 authority templates …

	*OBJOPR	*OBJMGT	*OBJEXIST	*OBJALTER	***OBJREF**	*READ	*ADD	*UPD	*DLT	***EXECUTE**
*ALL	X	X	X	X	X	X	X	X	X	X
*CHANGE	X					X	X	X	X	X
*USE	X					X				X
*EXCLUDE										

Authority Shortcuts

PowerTech. your security expert

… to quickly assign more complex authorities

	*OBJOPR	*OBJMGT	*OBJEXIST	*OBJALTER	***OBJREF**	*READ	*ADD	*UPD	*DLT	***EXECUTE**
*ALL	X	X	X	X	X	X	X	X	X	X
*CHANGE	X					X	X	X	X	X
*USE	X					X				X
*EXCLUDE										

Authority Shortcuts

PowerTech. your security expert

These are the OBJECT authorities.

	*OBJOPR	*OBJMGT	*OBJEXIST	*OBJALTER	*OBJREF	*READ	*ADD	*UPD	*DLT	*EXECUTE
*ALL	X	X	X	X	X	X	X	X	X	X
*CHANGE	X					X	X	X	X	X
*USE	X					X				X
*EXCLUDE										

Authority Shortcuts

PowerTech. your security expert

Although endless combinations are possible, it does not have to be as complex as it might seem.

- *EXCLUDE — Object cannot be accessed.
- *USE — Minimum authority necessary to “use” the object (read it / run it / look at it).
- *CHANGE — Adds the ability to modify the object’s contents.
- *ALL — Can do everything, including deleting the object itself. Do NOT grant lightly.

HOT TIP!

Deploy using IBM i templates whenever possible.

Authority Shortcuts

PowerTech. your security expert

*OBJOPR – Object Operational	Look at the description of an object and use the object as determined by the data authorities the user has.
*OBJMGT – Object Management	Move or rename an object or add members to database files. Superset of *OBJALTER and *OBJREF
*OBJEXIST – Object Existence	Change ownership and delete the object, free storage for the object, perform save and restore operations
*OBJALTER – Object Alter	Add, clear, initialize and reorganize members of database files, after and add attributes of database files, add and remove triggers, change attributes of SQL packages
*OBJREF – Object Reference	Specify database file as the parent in a referential constraint
*AUTLMGT – Authorization List Management	Add and remove users and their authorities from an authorization list.

Authority Shortcuts

PowerTech. your security expert

And these are the DATA authorities.

	*OBJOPR	*OBJMGT	*OBJEXIST	*OBJALTER	***OBJREF**	*READ	*ADD	*UPD	*DLT	***EXECUTE**
*ALL	X	X	X	X	X	X	X	X	X	X
*CHANGE	X					X	X	X	X	X
*USE	X					X				X
*EXCLUDE										

Authority Shortcuts

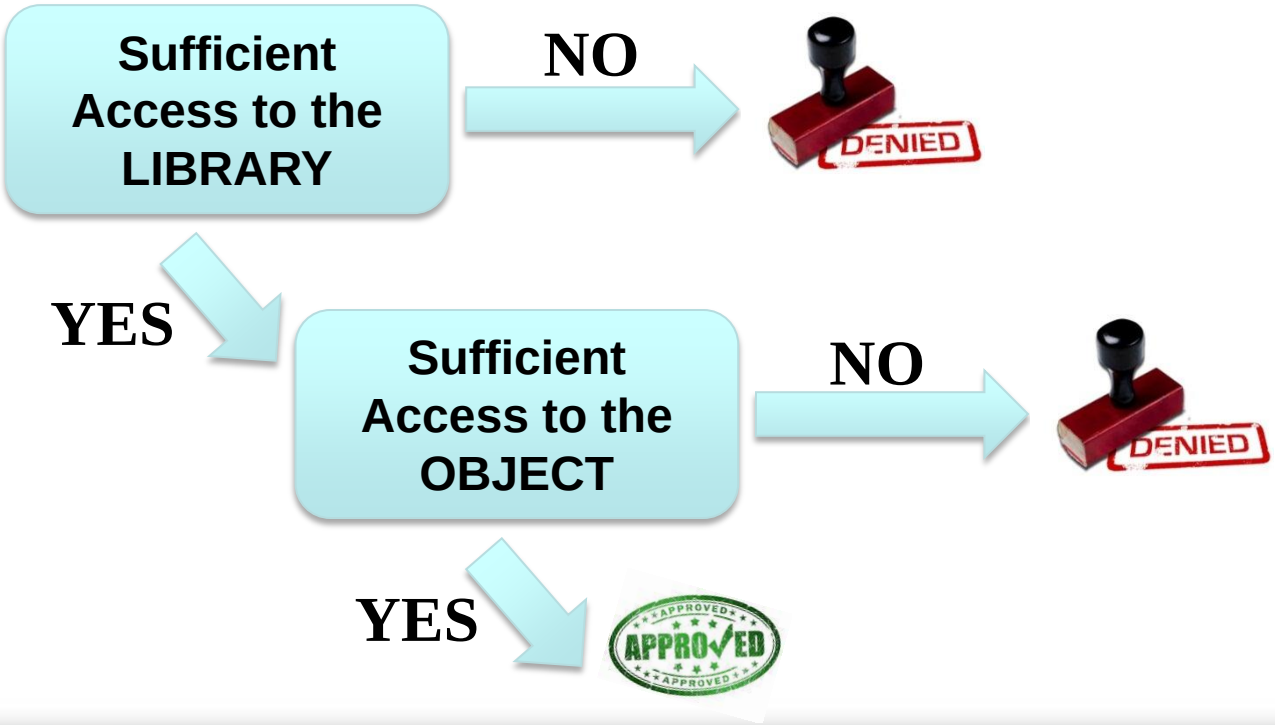
PowerTech. your security expert

*READ	Display the contents of an object, such as viewing the records in a file
*ADD	Add entries to an object, such as adding messages to a message queue, or records to a file
*UPD (Update)	Change entries in an object, such as changing records in a file
*DLT (Delete)	Remove entries from an object, such as removing messages from a message queue or deleting records from a file
*EXECUTE	Run a program or search a library or directory

Authority Considerations

PowerTech. your security expert

IBM i performs TWO evaluations before permitting access to an object.



Authority Considerations

Establishing an application environment that's compliant with object-level security is remarkably quick and easy:

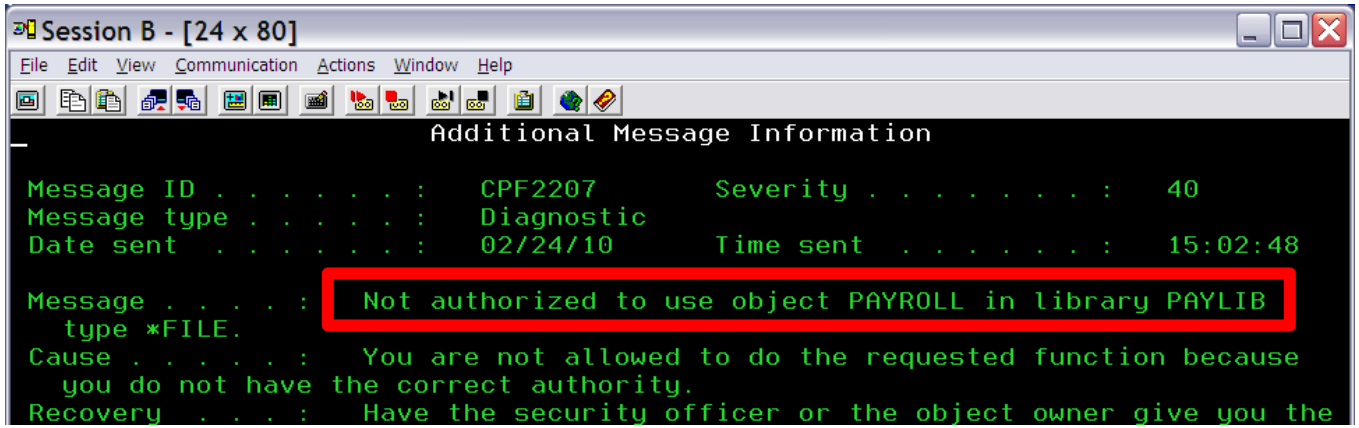
- Place programs in a library and grant *USE access to authorized users
- Place files and data areas in a data library and grant *USE or *CHANGE access to authorized users

HOT TIP!

If you use adopted or swap authority, you can even set public authority to *EXCLUDE (more on this later).

Authority Considerations

If you over-secure an object, or fail to elevate authority at runtime, the user will receive an authority failure.



An “AF” entry will be logged to QAUDJRN audit journal.

(You’ve activated IBM i auditing right?)

Authority Considerations

Do NOT respond by granting the user *ALLOBJ special authority as this is a system-wide override!!

Determine *why* the failure occurred and correct it.

Private Authority

Private authority is "**named**" access, and granted to an individual user or group profile

(Public authority represents "**anonymous**" access)

Private authority can be more restrictive but is typically *less* restrictive than public authority

Common terms: Deny-by-default &
Least privilege

Private Authority

```
                          Display Object Authority

Object . . . . . . . :   PAYNEW          Owner  . . . . . . . :   PAYOWN
  Library  . . . . . :     PAYROLL       Primary group  . . . :   *NONE
Object type  . . . . :   *FILE           ASP device . . . . . :   *SYSBAS

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

                          Object
User         Group       Authority
*PUBLIC                  *EXCLUDE
PAYOWN                   *ALL
READONLY                 *USE

                                                                  Bottom
Press Enter to continue.

F3=Exit   F11=Display detail object authorities   F12=Cancel   F17=Top
F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 2005.
```

Private authorities are for specific users or groups and are optional

Group Profiles

PowerTech. your security expert

Group Profiles

Group profiles are a mechanism for **role-based access control** (RBAC).

Associate users with similar security requirements using a group and grant application authority to the group.

A user can belong to 1 primary group and up to 15 supplemental groups (don’t go “group crazy”).

Users inherit private and special authorities from ALL of their groups (private authorities are additive).

Group Profiles

Group profiles are for organization and authority inheritance and should never be used to sign on (even for development purposes).

Group profiles are created like any other user, except we recommend:

- PASSWORD(*NONE)
- INLPGM(*NONE)
- INLMNU(*SIGNOFF)
- LMTCPB(*YES)

Group Profiles

A group profile is like any other user profile until it's designated as a group profile for another user.

```
                         Change User Profile (CHGUSRPRF)

Type choices, press Enter.

Keyboard buffering . . . . . . .   *SYSVAL      *SAME,
Maximum allowed storage  . . . .   *NOMAX       Kilobyt
Highest schedule priority  . . .   0            0-9, *S
Job description  . . . . . . . .  QDFTJOBD      Name, *S

Group profile  . . . . . . . . . .   SECADMINS
Group authority  . . . . . . . .   *NONE        *SAME,
Group authority type . . . . . .   *PRIVATE     *PRIVA
Supplemental groups  . . . . . .   *NONE        Name, *S
                + for more values
```

Authorization Lists

Authorization Lists

PowerTech. your security expert

Authorization lists are an **organizational mechanism** for securing objects with similar security requirements:

- All objects secured by an authorization list obtain private authorities (and, optionally, public authorities) from the list
- You can still grant specific authorities to objects to augment (override) the authorities on the authorization list

CRTAUTL AUTL(myautl) AUT(*EXCLUDE)

ADDAUTLE or EDTAUTL to maintain the list entries

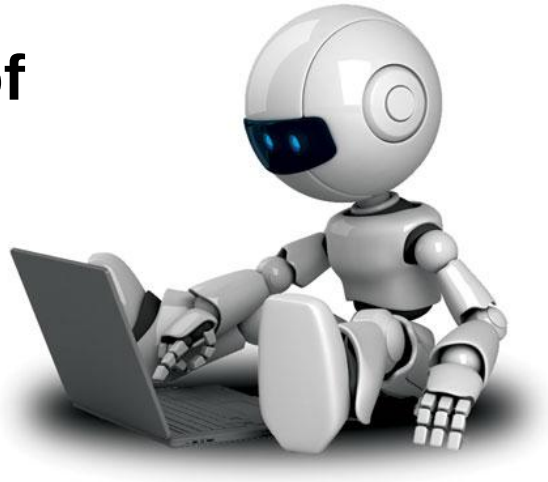
HOT TIP!

24x7 shop? Changing authorities on an authorization list does NOT require a lock on the object.

Authorization Lists

Authorization lists are not required; especially for simple authorization schemes. For example, if using adoption or a profile swap, then everything can simply be set to *EXCLUDE.

Authorization lists may help **future-proof** your application security and also permit access from outside the application (e.g. for file downloading).



Authorization Lists

```
                          Display Object Authority

 Object . . . . . . . :   PAYROLL        Owner  . . . . . . . :   PAYOWN
   Library  . . . . . :     PAYROLL      Primary group  . . . :   *NONE
 Object type  . . . . :   *FILE          ASP device . . . . . :   *SYSBAS

 Object secured by authorization list  . . . . . . . . . . . . :   PAYROLL

                          Object
 User        Group       Authority
 *PUBLIC                 *EXCLUDE
 PAYOWN                  *ALL
 READONLY                *USE

 Press Enter to continue.

 F3=Exit   F11=Display detail object authorities   F12=Cancel
 F14=Display authorization list   F17=Top          F18=Bottom
 (C) COPYRIGHT IBM CORP. 1980, 2005.
```

This object is secured by the PAYROLL authorization list

Authorization Lists

```
                          Display Object Authority

 Object . . . . . . . :   PAYROLL        Owner  . . . . . . . :   PAYOWN
   Library  . . . . . :     PAYROLL      Primary group  . . . :   *NONE
 Object type  . . . . :   *FILE          ASP device . . . . . :   *SYSBAS

 Object secured by authorization list  . . . . . . . . . . . . :   PAYROLL

                              Object
 User        Group           Authority
 *PUBLIC                     *EXCLUDE
 PAYOWN                      *ALL
 READONLY                    *USE

 Press Enter to continue.

 F3=Exit   F11=Display detail object authorities   F12=Cancel
 F14=Display authorization list   F17=Top          F18=Bottom
 (C) COPYRIGHT IBM CORP. 1980, 2005.
```

These authorities take precedence over those on the authorization list

Authorization Lists

```
                          Display Object Authority

Object . . . . . . . :   PAYROLL        Owner  . . . . . . . :   PAYOWN
  Library  . . . . . :     PAYROLL      Primary group  . . . :   *NONE
Object type  . . . . :   *FILE          ASP device . . . . . :   *SYSBAS

Object secured by authorization list  . . . . . . . . . . . . :   PAYROLL

                          Object
User          Group       Authority
*PUBLIC                   *EXCLUDE
PAYOWN                    *ALL
READONLY                  *USE

Press Enter to continue.

F3=Exit   F11=Display detail object authorities   F12=Cancel
F14=Display authorization list   F17=Top          F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 2005.
```

You must manually set *PUBLIC to *AUTL to defer to the authorization list

Group or Authorization List?

PowerTech. your security expert

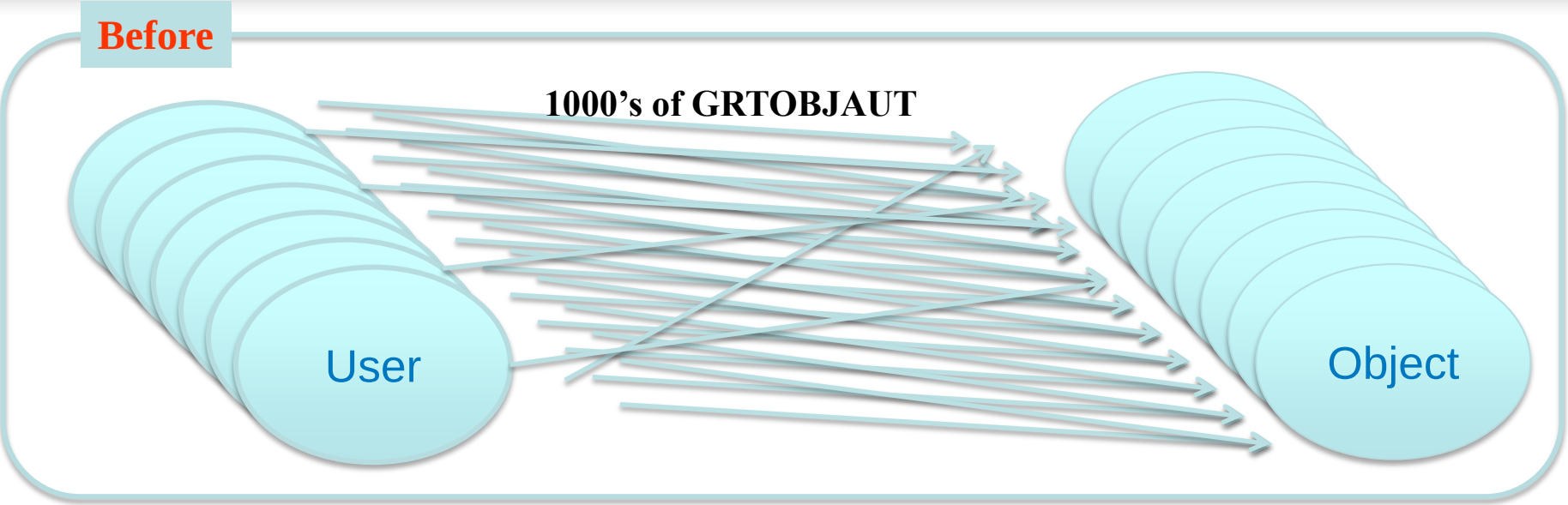
This is a very popular question. It's typically not a decision of *which* one you should use; consider using them both.

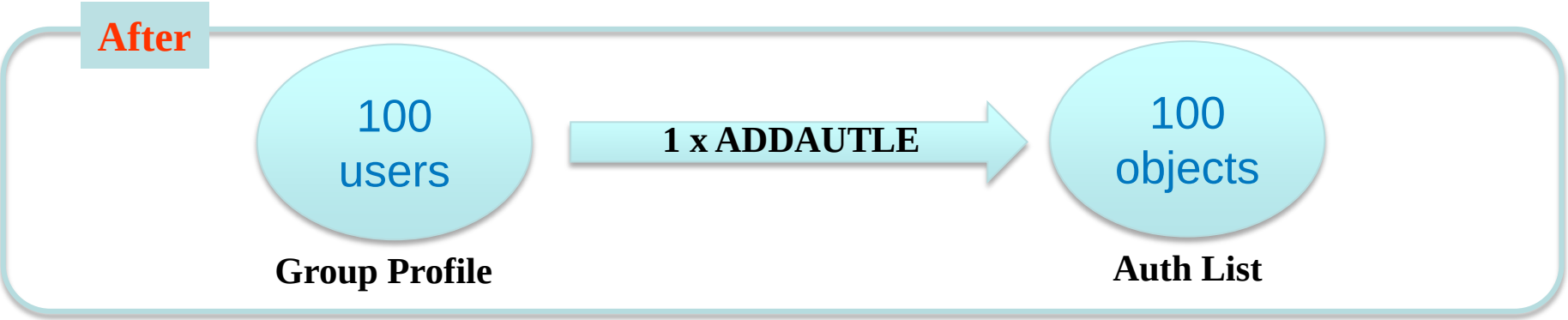
Remember:

- Groups associates <u>users</u> with similar access requirements.
- Authorization List secures <u>objects</u> with similar security requirements.

Group or Authorization List?

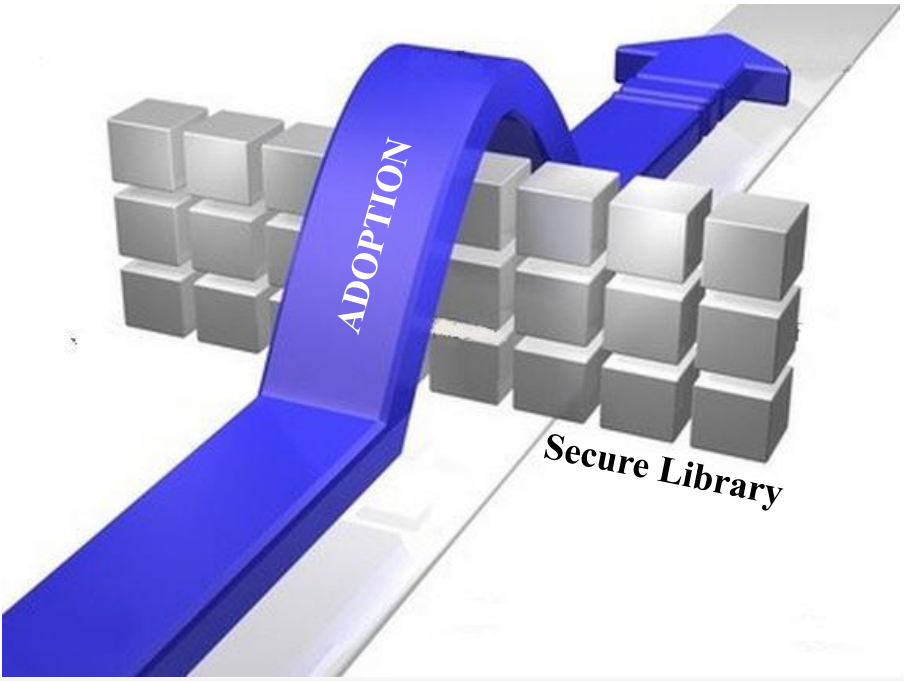
PowerTech. your security expert





Adopted Authority

Adoption permits a user to access objects that are normally restricted.



It works by inheriting the authority of the application program's owner profile to supplement the user's own authority.

It's only used if the user's own authority fails the authorization tests.

Adopted Authority

Normally, a program executes with the authority of the user running it.

Adoption comes into play if the IBM i authority check determines that the user does not have sufficient permission.

Adoption adds the authority of the programs' object owner which can (potentially) increase the effective authority while the program runs.

Adopted Authority

In addition, authority can be inherited from programs higher in the call stack; however, that is a separate setting.

Some nuances:

- Authority is additive (adoption cannot reduce a user's authority)
- Adoption is not observed in the Integrated File System (use a profile swap instead)
- Authority can be adopted from multiple owning profiles

HOT TIP!

If a program uses adopted authority, it should never present a command line to the user!

Adopted Authority

```
                          Display Program Information

Program  . . . . . . . :   CFG001C        Library                  :   UTLCNFIG
Owner  . . . . . . . . :   QSECOFR
Program attribute  . . :   CLP

Program creation information:
  Program creation date/time . . . . . .                            :06:42
  Type of program  . . . . . . . . . . .
  Source file  . . . . . . . . . . . . . . . . . . . . . . . . :   QCL
    Library  . . . . . . . . . . . . . . . . . . . . . . . . . :     U   CNFIGS
  Source member  . . . . . . . . . . . . . . . . . . . . . . . :   CFG   1C
  Source file change date/time . . . . . . . . . . . . . . . . :   01    /13  15:56:05
  Observable information . . . . . . . . . . . . . . . . . . . :   *ALL
  User profile . . . . . . . . . . . . . . . . . . . . . . . . :   *OWNER
  Use adopted authority . . . . . . . . . . . . . . . . . . . :   *YES
  Log commands (CL program)  . . . . . . . . . . . . . . . . :   *JOB
  Allow RTVCLSRC (CL program)  . . . . . . . . . . . . . . . :   *YES
  Fix decimal data . . . . . . . . . . . . . . . . . . . . . :   *NO
                                                                      More...
Press Enter to continue.

F3=Exit   F12=Cancel
(C) COPYRIGHT IBM CORP. 1980, 2005.
```

Activate Adoption by setting this parameter to *OWNER

Adopted Authority

```
                          Display Program Information

 Program . . . . . . . . . :   CFG001C        Library . . . . . . . . . :   UTLCNFIG
 Owner . . . . . . . . . . :   QSECOFR
 Program attribute . . . :   CLP

 Program creation information:
   Program creation date/time . . . . . . . . . . . . . . . . . . . . . :06:42
   Type of program . . . . . . . . . . . . . . . . . . . . . . . . . . .
   Source file . . . . . . . . . . . . . . . . . . . . . . . . . . . . :   QCLSRC
     Library . . . . . . . . . . . . . . . . . . . . . . . . . . . . . :     UTLCNFIGS
   Source member . . . . . . . . . . . . . . . . . . . . . . . . . . . :   CFG001C
   Source file change date/time . . . . . . . . . . . . . . . . . . . :   01/08/13  15:56:05
   Observable information . . . . . . . . . . . . . . . . . . . . . . :   *ALL
   User profile . . . . . . . . . . . . . . . . . . . . . . . . . . . :   *OWNER
   Use adopted authority . . . . . . . . . . . . . . . . . . . . . . . :   *YES
   Log commands (CL program) . . . . . . . . . . . . . . . . . . . . :   *JOB
   Allow RTVCLSRC (CL program) . . . . . . . . . . . . . . . . . . . :   *YES
   Fix decimal data . . . . . . . . . . . . . . . . . . . . . . . . . :   *NO
                                                                                  More...
 Press Enter to continue.

 F3=Exit   F12=Cancel
 (C) COPYRIGHT IBM CORP. 1980, 2005.
```

And the program will also run with the authority of this profile

Adopted Authority

```
                          Display Program Information

 Program  . . . . . . . :   CFG001C      Library  . . . . . . . :   UTLCNFIG
 Owner  . . . . . . . . :
 Program attribute  . . :

 Program creation informati
   Program creation date/ti                                  01/08/13  16:06:42
   Type of program  . . . .                                  OPM
   Source file  . . . . . .                                  QCLSRC
     Library  . . . . . . . . . . . . . . . . . . . . :        UTLCNFIGS
   Source member  . . . . . . . . . . . . . . . . . . :      CFG001C
   Source file change date/time . . . . . . . . . . . :      01/08/13  15:56:05
   Observable information . . . . . . . . . . . . . . :      *ALL
   User profile . . . . . . . . . . . . . . . . . . . :      *OWNER
   Use adopted authority  . . . . . . . . . . . . . . :      *YES
   Log commands (CL program)  . . . . . . . . . . . . :      *JOB
   Allow RTVCLSRC (CL program)  . . . . . . . . . . . :      *YES
   Fix decimal data . . . . . . . . . . . . . . . . . :      *NO
                                                                          More...
 Press Enter to continue.

 F3=Exit   F12=Cancel
 (C) COPYRIGHT IBM CORP. 1980, 2005.
```

*YES directs IBM i to utilize authorities adopted from the prior programs in the call stack

Adopted Authority

A built-in function called *MODINVAU* controls whether the adoption is passed to a called program by turning it on and off *inside* the calling program.

This ensures that the correct setting is always active, even if the programmer forgets to set the program attribute correctly.

On: **CallPrc PRC('_MODINVAU') Parm(x'00')**

Off: **CallPrc PRC('_MODINVAU') Parm(x'01')**

Adopted Authority

PowerTech. your security expert

Program Call Stack

PGM_A
Owner: APPOWNER
User Profile: *OWNER

PGM_B
Owner: QSECOFR
Use Adopted Authority: *YES
User Profile: *OWNER

PGM_C
Owner: APPOWNER
Use Adopted Authority: *NO
User Profile: *USER

Users Checked

user then APPOWNER

user then APPOWNER (from PGM_A) then QSECOFR

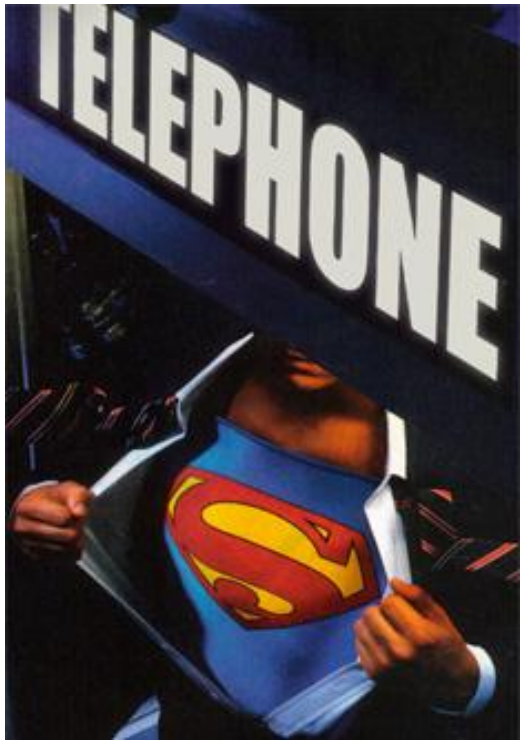
Only *user* because USEADPAUT(*NO) and USRPRF(*USER)

Swap Profiles

A more modern approach to altering authority is to swap to a more powerful profile.

Swapping relinquishes your current profile attributes and inherits multiple attributes of the target profile, such as:

- Special Authorities
- Private Authorities
- Group membership
- Command Line Permission



Swap Profiles

```
                          Display Job Status Attributes
                                                                System:   ROBINSON
 Job:   RTATAMA1       User:   RTATAM        Number:   947752

 Status of job . . . . . . . . . . . . . . . . . . . . :   ACTIVE
 Current user profile . . . . . . . . . . . . . . . . :   QSECOFR
 Job user identity . . . . . . . . . . . . . . . . . . :   QSECOFR
   Set by . . . . . . . . . . . . . . . . . . . . . . :     *DEFAULT
 Entered system:
   Date . . . . . . . . . . . . . . . . . . . . . . . :   02/11/13
   Time . . . . . . . . . . . . . . . . . . . . . . . :   12:54:45
 Started:
   Date . . . . . . . . . . . . . . . . . . . . . . . :   02/11/13
   Time . . . . . . . . . . . . . . . . . . . . . . . :   12:54:45
 Subsystem . . . . . . . . . . . . . . . . . . . . . . :   QINTER
   Subsystem pool ID . . . . . . . . . . . . . . . . . :     2
 Type of job . . . . . . . . . . . . . . . . . . . . . :   INTER
 Special environment . . . . . . . . . . . . . . . . . :   *NONE
 Program return code . . . . . . . . . . . . . . . . . :   1

 Press Enter to continue.

 F3=Exit   F5=Refresh   F12=Cancel   F16=Job menu
```

During an active swap, IBM i reacts as if you are signed on as this profile

Swap Profiles

```
                          Display Job Status Attributes
                                                                System:   ROBINSON
Job:   RTATAMA1       User:   RTATAM        Number:   947752

Status of job . . . . . . . . . . . . . . . . . :   ACTIVE
Current user profile . . . . . . . . . . . . . :   QSECOFR
Job user identity . . . . . . . . . . . . . . . :   QSECOFR
  Set by . . . . . . . . . . . . . . . . . . . . :     *DEFAULT
Entered system:
  Date . . . . . . . . . . . . . . . . . . . . . :   02/11/13
  Time . . . . . . . . . . . . . . . . . . . . . :   12:54:45
Started:
  Date . . . . . . . . . . . . . . . . . . . . . :   02/11/13
  Time . . . . . . . . . . . . . . . . . . . . . :   12:54:45
Subsystem . . . . . . . . . . . . . . . . . . . :   QINTER
  Subsystem pool ID . . . . . . . . . . . . . . :     2
Type of job . . . . . . . . . . . . . . . . . . :   INTER
Special environment . . . . . . . . . . . . . . :   *NONE
Program return code . . . . . . . . . . . . . . :   1
                                                                          More...
Press Enter to continue.

F3=Exit   F5=Refresh   F12=Cancel   F16=Job menu
```

Auditing is tied to the original job (so concurrent swapping is okay)

Swap Profiles

PowerTech. your security expert

Swapping is performed via security APIs:

- QSYGETPH — Get profile handle
- QWTSETP — Swap profile using profile handle
- QSYRLSPH — Release profile handle

Programs may need to use adoption to satisfy API rules:

- Users must have at least *USE access to the target profile
- If the target profile has an expired password, user must also have *ALLOBJ and *SECADM
- If the target profile is disabled, profile handle may be denied or user must also have *ALLOBJ and *SECADM (depends on API parameters)

Swap Profiles

```
                    Retrieve Job Attributes (RTVJOBA)

Type choices, press Enter.

CL var for JOB          (10) . .   ____________   Character value
CL var for USER         (10) . .   ____________   Character value
CL var for NBR           (8) . .   ____________   Character value
CL var for CURUSER      (10) . .   ____________   Character value
CL var for TYPE          (1) . .   ____________   Character value
CL var for SUBTYPE       (1) . .   ____________   Character value
CL var for SYSLIBL     (165) . .   ____________   Character
CL var for CURLIB       (10) . .   ____________   Character
CL var for USRLIBL    (2750) . .   ____________   Character
CL var for ASPGRP       (10) . .   ____________   Character
CL var for LOGLVL        (1) . .   ____________   Character
CL var for LOGSEV      (2 0) . .   ____________   Numeric
CL var for LOGTYPE      (10) . .   ____________   Character
CL var for LOGCLPGM     (10) . .   ____________   Character
CL var for LOGOUTPUT    (10) . .   ____________   Character
CL var for JOBMSGQMX   (2 0) . .   ____________   Number
                                                              More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

When retrieving the active user, always use *CURUSER*

Swap Profiles

Swapping resolves three key challenges with adoption:

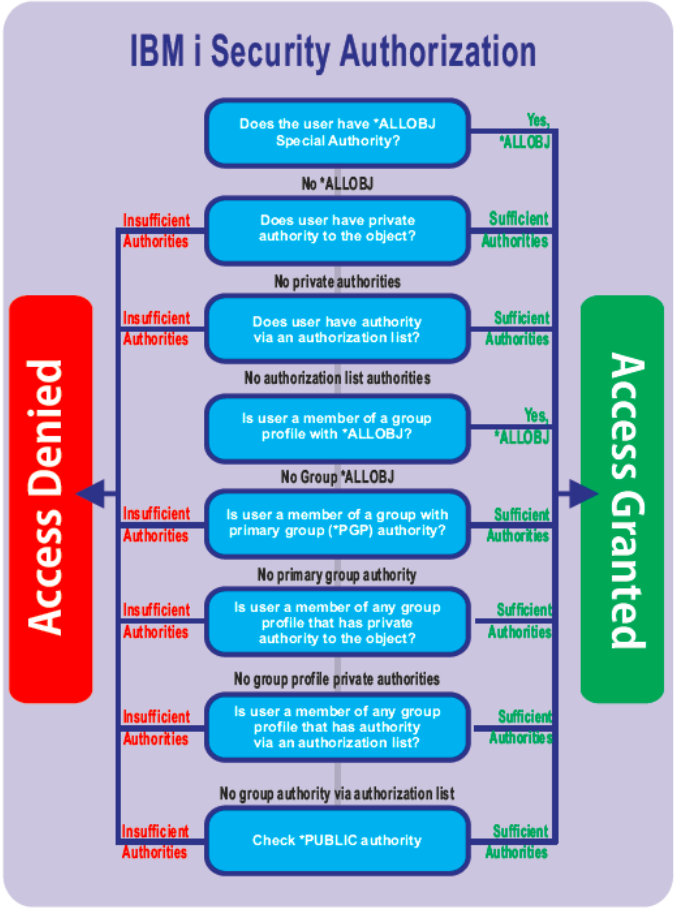
- It's honored within the IFS
- Permits up- AND down-grading authority
- Supported by non-5250 interfaces

Swapping is flexible as it can be turned on and off programmatically; however, only one swap can be active at a time.

HOT TIP!

If a program uses profile swapping, remember to swap back before presenting the user with a command line!

Authority Progression



Understanding how IBM i determines whether access will be granted or denied aids planning and troubleshooting

For a full-sized copy of this chart, email a request to robin.tatam@powertech.com

Today's Agenda

- Introductions
- Design and Documentation
- Application Ownership and Authority
- **A Simple Security Model**
- Integrity Considerations
- Resources for Security Officers
- Questions & Answers

A Simple Security Model

There are many different ways to build a secure application.

Let’s explore how to deploy a native application with authority adoption within a secure library.

A Simple Security Model

PowerTech your security expert

Create application owner profile

```
CRTUSRPRF  USRPRF(appowner)  PASSWORD(*NONE)  INLPGM(*NONE)
INLMNU(*SIGNOFF)  USRCLS(*USER)  SPCAUT(*NONE) LMTCPB(*YES)
```

Create authorization lists

```
CRTAUTL  AUTL(dataautl)  AUT(*EXCLUDE)
CHGOBJOWN  OBJ(dataautl)  OBJTYPE(*AUTL)  NEWOWN(appowner)

CRTAUTL  AUTL(pgmautl) AUT(*EXCLUDE)
CHGOBJOWN  OBJ(pgmautl)  OBJTYPE(*AUTL)  NEWOWN(appowner)
```

A Simple Security Model

PowerTech your security expert

Establish secure libraries for programs and data

```
CRTLIB  LIB(pgmlib)  AUT(*USE)  CRTAUT(*EXCLUDE)
CHGOBJOWN  OBJ(pgmlib)  OBJTYPE(*LIB)  NEWOWN(appowner)

CRTLIB  LIB(datalib)  AUT(*EXCLUDE)  CRTAUT(*EXCLUDE)
CHGOBJOWN  OBJ(datalib)  OBJTYPE(*LIB)  NEWOWN(appowner)
```

Link libraries to authorization lists

```
GRTOBJAUT  OBJ(datalib)  OBJTYPE(*LIB)  AUTL(dataautl)
GRTOBJAUT  OBJ(pgmlib)  OBJTYPE(*LIB)  AUTL(pgmautl)
```

A Simple Security Model

PowerTech your security expert

Create files (and data areas etc.)

```
CRTPF  FILE(datalib/myfile)  AUT(*LIBCRTAUT)

CHGOBJOWN  OBJ(datalib/myfile)  OBJTYPE(*FILE)  NEWOWN(appowner)
```

Link files to authorization lists

```
GRTOBJAUT  OBJ(datalib/myfile)  OBJTYPE(*FILE)  AUTL(dataautl)
```

A Simple Security Model

PowerTech your security expert

Create programs

```
CRTPGM  PGM(pgmlib/mypgm)  AUT(*LIBCRTAUT)

CHGOBJOWN  OBJ(pgmlib/mypgm)  OBJTYPE(*PGM)  NEWOWN(appowner)
```

Link programs to authorization list

```
GRTOBJAUT  OBJ(pgmlib/mypgm)  OBJTYPE(*PGM)  AUTL(pgmautl)
```

A Simple Security Model

PowerTech your security expert

Defer public authorities to come from AUTLs

```
GRTOBJAUT  OBJ(datalib/myfile)  OBJTYPE(*FILE)  USER(*PUBLIC)
AUTL(dataautl)

GRTOBJAUT  OBJ(pgmlib/mypgm)  OBJTYPE(*FILE)  USER(*PUBLIC)
AUTL(pgmautl)
```

A Simple Security Model

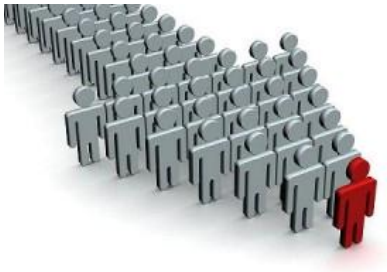
PowerTech your security expert

Set entry point program to use owner authority

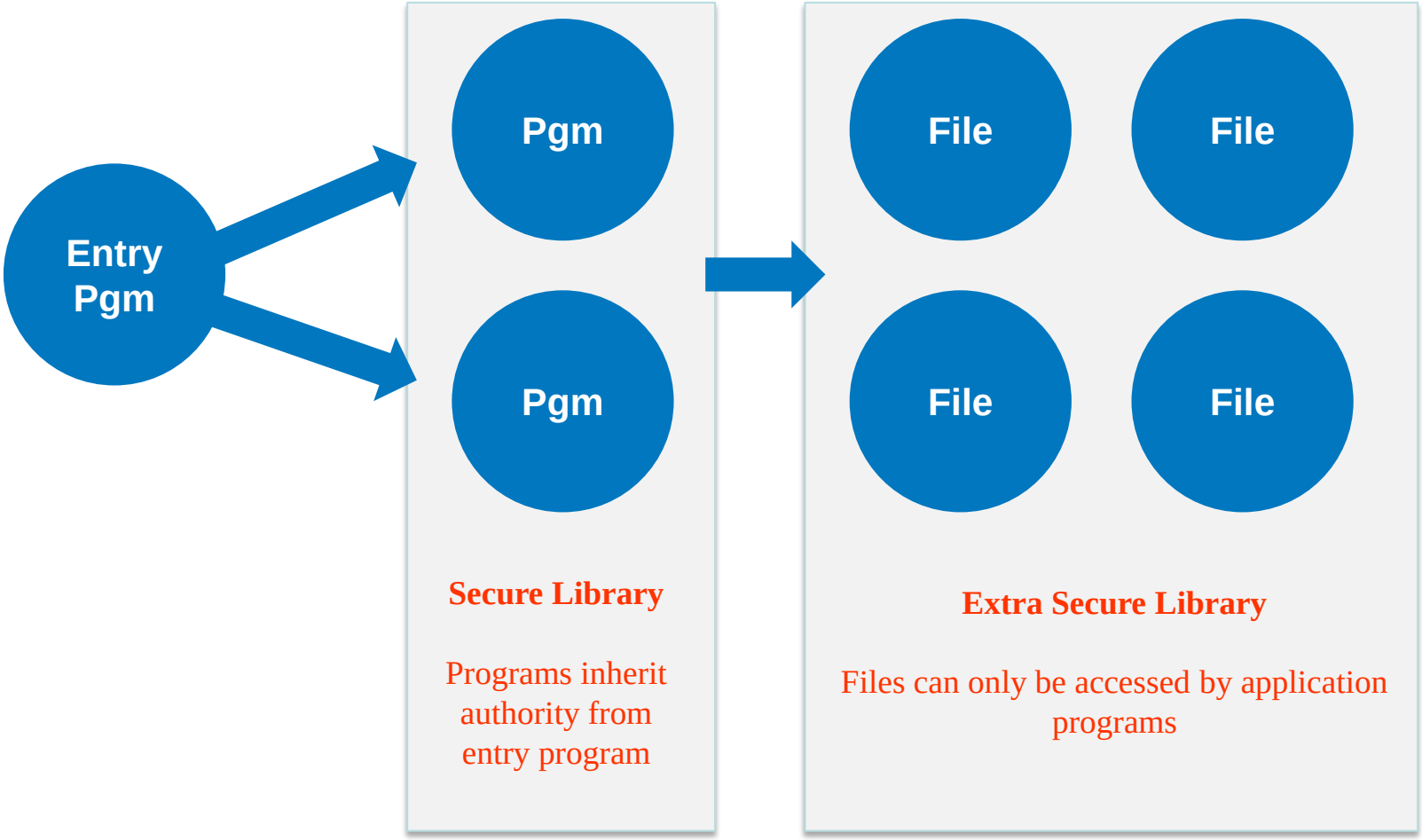
CHGPGM PGM(pgmlib/myentrypgm) USRPRF(*OWNER)

Authorize application users to the entry point

GRTOBJAUT OBJ(pgmlib/myentrypgm) OBJTYPE(*PGM) USER(user-or-group) AUT(*USE)



A Simple Security Model



A Simple Security Model

PowerTech. your security expert

Some additional considerations

- Objects created during runtime
- Non-5250 access
- Query access
- IFS objects
  (Reminder: no adoption in IFS)

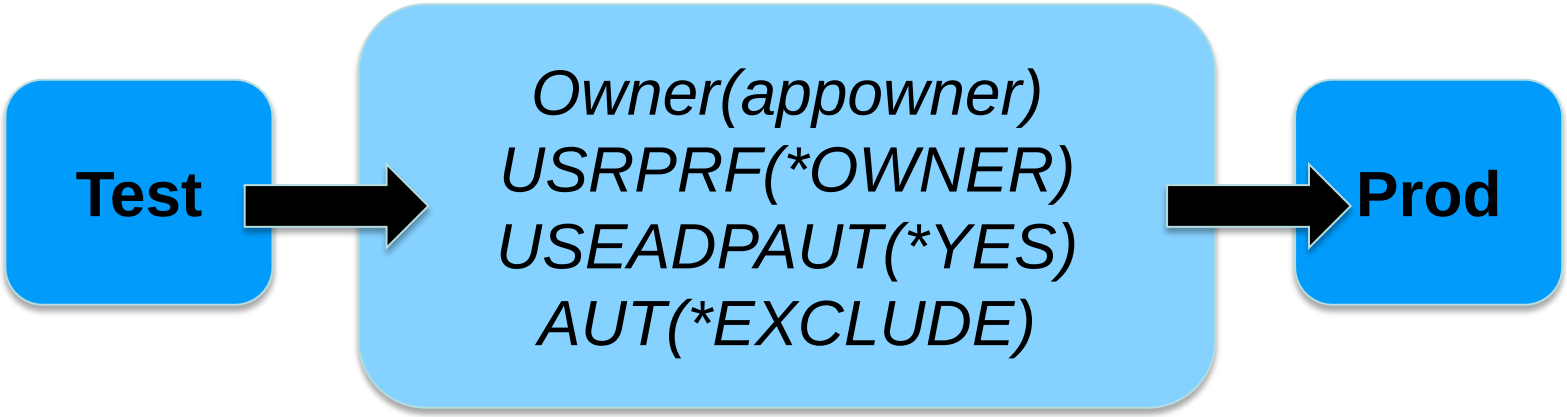
Today's Agenda

- Introductions
- Design and Documentation
- Application Ownership and Authority
- A Simple Security Model
- **Integrity Considerations**
- Resources for Security Officers
- Questions & Answers

Development & Deployment

PowerTech. your security expert

Lifecycle applications (aka change control) enable programs to be deployed into a production library—securely and consistently—and establish the correct runtime attributes.



Save & Restore

When restoring the application:

- Ensure that the owner profile exists prior to restoring the application objects.
- If private authority is used, restore the user profiles (including authorization lists) and then restore the application objects.
- Finally, restore the user's private authorities using the RSTAUT command.

An application "builder" (CL program) greatly simplifies the security configuration process

HSP

PowerTech. your security expert

At security level 40 or 50, integrity is enforced and user programs must use APIs and approved interfaces to access to system objects.

IBM i performs **Hardware Storage Protection** and **Domain Validation** to prevent system objects being accessed directly via memory pointers.

HOT TIP!

QSECURITY levels below 40 have well-known security vulnerabilities. Do NOT run below level 40!

Domain Checking

```
                          Display Object Description - Full
                                                                  Library 1 of 1
 Object . . . . . . . :   SIGNOFF          Attribute  . . . . . :
   Library  . . . . .                                . . . . . . :   QSYS
 Library ASP device .                        ASP group  . . :   *SYSBAS
 Type . . . . . . . .                        group  . . . . :   *NONE

 User-defined informat
   Attribute  . . . .
   Text . . . . . . .                                   off

 Creation information:
   Creation date/time . . . . . . . . . . :   06  6/05  21:37:17
   Created by user  . . . . . . . . . . . :   *I
   System created on                      :   00000000
   Object domain  . . . . . . . . . . . . :   *SYSTEM


                                                                  More...
 Press Enter to continue.

 F3=Exit   F12=Cancel
 (C) COPYRIGHT IBM CORP. 1980, 2005.
```

Every object has a domain, *SYSTEM or *USER

Domain Checking

```
                          Display Program Information

 Program  . . . . . . . . :   CFG0                          . . . :   UTLCNFIG
 Owner  . . . . . . . . . :   QSEC
 Program attribute  . . :   CLP

 Program statistics:
   Number of parameters . . . .
   Program size (bytes) . . . .                                   728
   Associated space size (bytes)                                 12880
   Static storage size (bytes)  . . . . . . . . . . :
   Automatic storage size (bytes) . . . . . . . . . :            024
   Number of MI instructions  . . . . . . . . . . . :            79
   Number of ODT entries                                         2615
   Program state  . . . . . . . . . . . . . . . . . :   *USER
   Program domain . . . . . . . . . . . . . . . . . :   *USER
   Compiler . . . . . . . . . . . . . . . . . . . . :           V5R4M0
   Earliest release that program can run  . . . . . :   V5R2M0
   Conversion required  . . . . . . . . . . . . . . :   *NO
                                                                   More...

 Press Enter to continue.

 F3=Exit   F12=Cancel
```

Every program has a state, *SYSTEM or *USER

Domain Checking

- Programs running *SYSTEM state can access both *USER and *SYSTEM domain objects.
- Programs running *USER state can only access *USER domain objects.

*USER domain user objects (QUSRxxx) can be created in QTEMP plus anywhere listed in the QALWOBJDMN system value.

HOT TIP!

Domain and State compatibility is only enforced at security levels 40 and 50.

Trojan Horses

Contrary to what many of us were taught, *LIBL increases the risk that an application can be compromised.

Although hard-coding a library is often not desired, consider using soft-coding library names in a data area or file.

Menu Security

Menus are a beneficial application interface but they are NOT considered adequately secure.

The problem is that:

- Menus are often used as the only form of access control
- Not all access comes via legacy native 5250 (telnet)
- Object security is often deemed unnecessary

Exit programs can provide a **compensating control**; however, best security practices should still be used.

Today's Agenda

- Introductions
- Design and Documentation
- Application Ownership and Authority
- Integrity Considerations
- **Resources for Security Officers**
- Questions & Answers

Comprehensive Security Solutions for Power Systems

PowerTech
your security expert

Compliance Monitor
Custom auditing and reporting

Authority Broker
Management of privileged users

DataThread
Real-time database monitoring

PowerAdmin
Centralized user profile management

Agent for RSA SecurID
Two-factor authentication for IBM i

Network Security
Access control

Command Security
Command monitoring

Interact
Real-time SIEM integration

StandGuard Anti-Virus
Advanced native virus detection

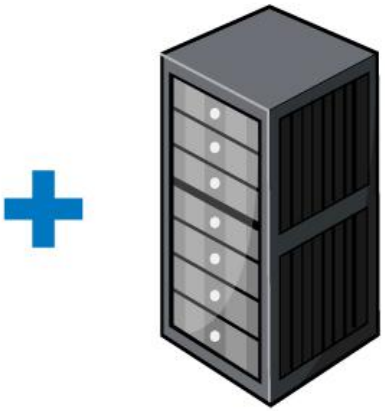
Password Self Help
Self-service password reset

Compliance Assessment
Free audit of system-wide security

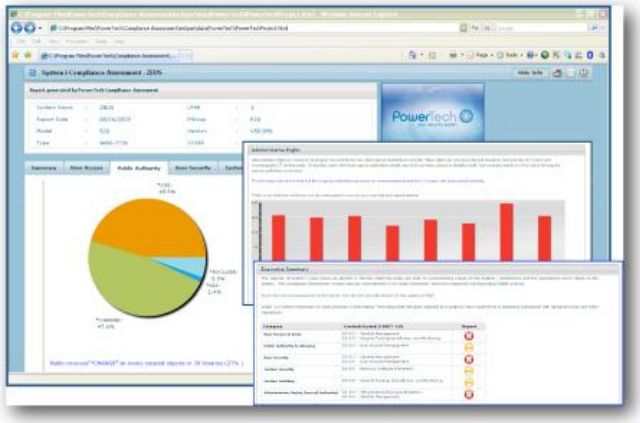
I can perform a Risk Assessment of your IBM i

PowerTech Compliance Assessment

YOUR PC



YOUR IBM i SERVER

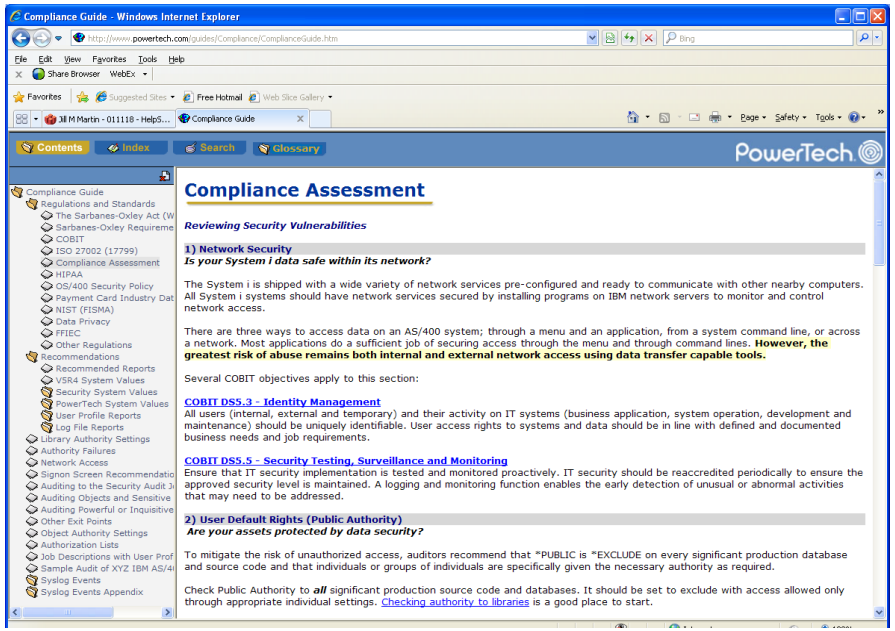


YOUR VULNERABILITIES

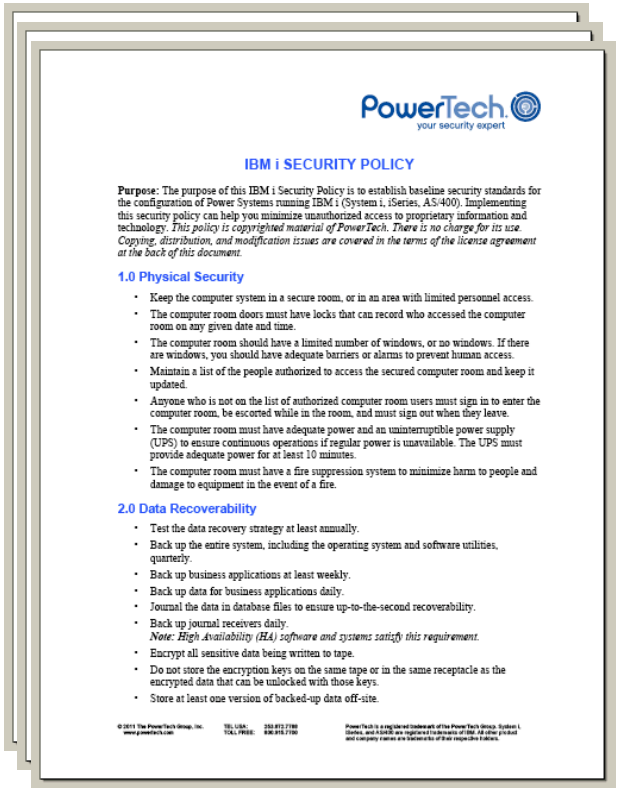
Compliance Resources

PowerTech. your security expert

Online Compliance Guide



Security Policy



PowerTech. your security expert

IBM i SECURITY POLICY

Purpose: The purpose of this IBM i Security Policy is to establish baseline security standards for the configuration of Power Systems running IBM i (System i, iSeries, AS/400). Implementing this security policy can help you minimize unauthorized access to proprietary information and technology. *This policy is copyrighted material of PowerTech. There is no charge for its use. Copying, distribution, and modification issues are covered in the terms of the license agreement at the back of this document.*

1.0 Physical Security

- Keep the computer system in a secure room, or in an area with limited personnel access.
- The computer room doors must have locks that can record who accessed the computer room on any given date and time.
- The computer room should have a limited number of windows, or no windows. If there are windows, you should have adequate barriers or alarms to prevent human access.
- Maintain a list of the people authorized to access the secured computer room and keep it updated.
- Anyone who is not on the list of authorized computer room users must sign in to enter the computer room, be escorted while in the room, and must sign out when they leave.
- The computer room must have adequate power and an uninterruptible power supply (UPS) to ensure continuous operations if regular power is unavailable. The UPS must provide adequate power for at least 10 minutes.
- The computer room must have a fire suppression system to minimize harm to people and damage to equipment in the event of a fire.

2.0 Data Recoverability

- Test the data recovery strategy at least annually.
- Back up the entire system, including the operating system and software utilities, quarterly.
- Back up business applications at least weekly.
- Back up data for business applications daily.
- Journal the data in database files to ensure up-to-the-second recoverability.
- Back up journal receivers daily.
  Note: High Availability (HA) software and systems satisfy this requirement.
- Encrypt all sensitive data being written to tape.
- Do not store the encryption keys on the same tape or in the same receptacle as the encrypted data that can be unlocked with those keys.
- Store at least one version of backed-up data off-site.

Thanks for your time!

PowerTech
your security expert

Please visit www.helpsystems.com/powertech to access:

- Demonstration Videos & Trial Downloads
- Product Information Datasheets
- White Papers and Technical Articles
- Case Studies
- PowerNews Newsletter Registration
- FREE Compliance Assessment

www.helpsystems.com/powertech (800) 328-1000
info.powertech@helpsystems.com

Today's Agenda

- Introductions
- Design and Documentation
- Application Ownership and Authority
- Integrity Considerations
- Resources for Security Officers
- **Questions & Answers**

Q & A

PowerTech.
your security expert

PowerTech.
your security expert

THANK YOU.

+1 253-872-7788 info.powertech@helpsystems.com
www.helpsystems.com/powertech