



SECURITY PATROL

by Vincent LeVeque

The Primary Group Is Number One

QUESTION: I'd like to organize my user community into groups and have each group work under a specific group profile, but I'm concerned about the performance impact of this. Doesn't checking authority through a profile's group involve a lot of overhead? Wouldn't there be a great performance advantage to just using *PUBLIC as my default authority?

ANSWER: Long ago, it was true that validating authority through a profile's group involved additional processing steps. OS/400 could quickly verify authority for the object's owner or for *PUBLIC, as the object's header carried both authorities. Everything else required looking up private authorities, and along with that came a performance penalty.

As I said, this was long ago. With V3R1, a change came that added something called the Primary Group Profile (PGP) to the object's header. The PGP was there to support the UNIX model for file permissions, which was essential for using the AS/400 Integrated File System (AS/400 IFS). The reason why AS/400 IFS security required a PGP and not the traditional profile in QSYS had to do with how UNIX established file permissions. Unlike the AS/400, which had a very rich object authority structure, UNIX had a basic permissions structure, distinguishing between only World, Group, and Owner authorities. World corresponded to *PUBLIC owner, which was the same as on OS/400, but, aside from time-consuming private authorities, there was no corresponding OS/400 Group authority. The *PUBLIC and Owner authorities were always stored with the object header. Thus, no time-consuming lookup was required to determine whether or not the user was authorized to the object, as the system already had the object in memory to perform the access. Implementing the UNIX permissions scheme required an equivalent to Group.

The existing method for implementing UNIX-like Group permissions involved using private authority. Private authority involved looking up the specific authority in the authorized objects associated with the user

profile. To find out whether or not a user was authorized to an object, the system had to find the user profile and search for the private authority in the profile. This was a relatively time-consuming process and could degrade performance, particularly in situations such as online transaction entry, where many users and many different objects were involved. Because this slow, relatively cumbersome process for implementing such a critical part of the UNIX file system was not satisfactory, the PGP alternative was developed.

A PGP allows you to specify one and only one group profile and associated group authority for an object. Specifying a PGP requires the following:

- The user must have a group profile assigned in the AS/400 user profile.
- The user must own the object to specify the PGP.
- You must specify group authority for the object.
- You must specify *PGP authority.

Note that only one profile can be assigned

as a primary group. The PGP must have a Group ID (GID) assigned. This is the UNIX-like group identifier. It may be assigned at the time of the profile's creation via the Create User Profile (CRTUSRPRF) command or assigned to an existing profile via the Change User Profile (CHGUSRPRF) command. To get the extra parameters, be sure to press CMD 10 and scroll a few times to the last screen. GID is one of the last profile parameters listed.

There are several ways to establish PGP for an object. For the "traditional" QSYS file system, the command to change an object's PGP is Change Object PGP (CHGOBJPGP), which specifies the primary group for the object and the authority that the primary group has to the object. The corresponding command for AS/400 IFS objects is Change PGP (CHGPGP), which requires specifying the full path of the object. For DLO objects, there is the Change DLO PGP (CHGDLOPGP) command. There is also the Work with Object's PGP (WRKOBJPGP) command, which reviews every object that has a specific profile as its group profile.

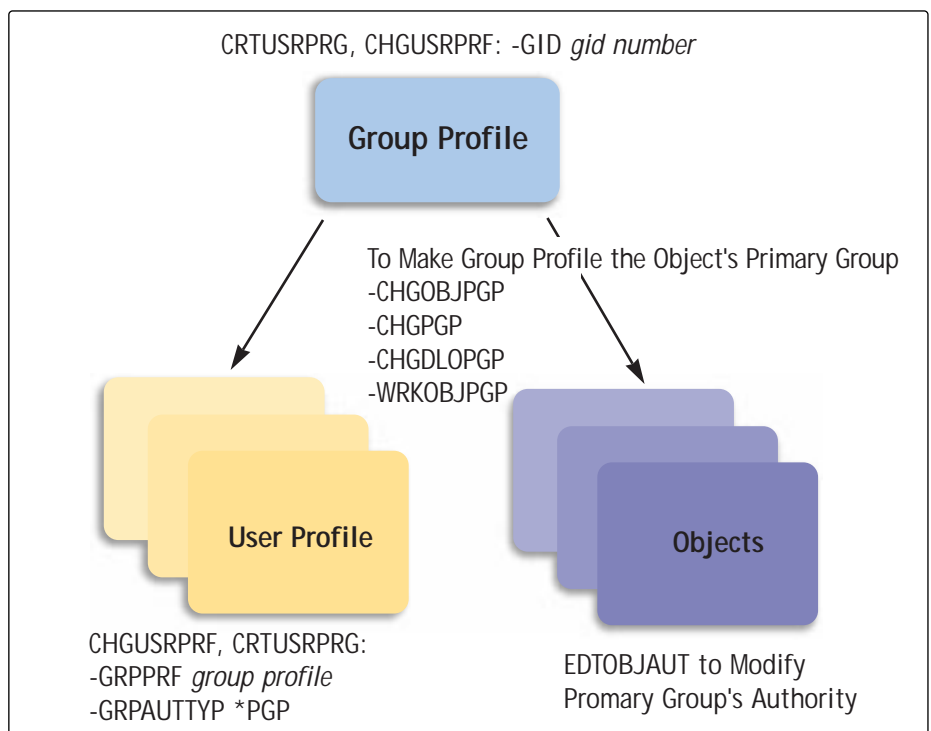


Figure 1: This is the relationship between group profiles, user profiles, and objects secured with PGPs.

You may also want to make sure that newly created objects are assigned primary group authority. To do this, ensure that all profiles assigned to the PGP have the Group authority type (GRPAUTTY) parameter set to *PGP.

Last, I am sure you are diligently auditing all security-relevant events on your AS/400. In your audit journal (QAUDJRN), you will see audit events designed specifically for PGPs as RZ for the primary group of the object changed during a restore operation and PG for the primary group of the object changed. The relationship between group profiles, user profiles, and objects secured with PGPs is illustrated in Figure 1.

I suggest using PGPs instead of *PUBLIC authority for your users' default access. You can follow the good security practice of *PUBLIC *EXCLUDE without the performance penalty of private authorities.

Wayne Evans touched on PGPs in an earlier Security Patrol article (MC, September 1998). You may also want to refer to this article for further information on this subject..

The Best Policy

QUESTION: As administrator of our AS/400 system, I'm responsible for security. I'd like to develop a security policy for our management, but I'm not sure exactly how to proceed. What should go into a security policy, and how detailed does it have to be?

ANSWER: There is a wide variety of opinions on the exact nature of security policies and their organization, their format, and even their size. I've seen policies that range from memo-sized informal documents to massive tomes of hundreds of pages. While I prefer the shorter policies, the best size seems to be whatever works. In fact, "whatever works" is the best advice for any security policy.

When writing the various sections of a security policy, keep in mind the three dimensions of security: confidentiality, integrity, and availability (CIA for short). *Confidentiality* describes protecting sensitive information in the sense that secrets are guarded and information is provided only to those who "need to know." *Integrity* refers to the trustworthiness of information: Does it really represent the events that it purports to? Integrity in a data warehouse, for example, means that management can rely on statistical analysis of information to make critical decisions. In a financial system, integrity means that all transactions are properly recorded per policy and accounting practices. *Availability* means having systems and data available when they are needed and with the quality required.

Typically, a security policy is expected to

include the following sections:

- *Statement of mission and objectives.* The mission and objectives should state what the policy is trying to accomplish. Keep this down to a paragraph or so and keep it nontechnical.
- *Definitions of terms.* The definitions of terms should standardize terminology so the rest of the document is entirely unambiguous.
- *Statement of corporate positions on security issues.* The corporate positions on security issues make up the meat of the policy. These may include policy statements regarding who is entitled to get accounts to various systems; user authentication; protection of identified, confidential corporate information (e.g., trade secrets and payroll); access control (i.e., who is permitted access to which resources); event auditing and review of audit logs; proper use of corporate PCs, email, and the Internet; and repair of security problems.
- *Definitions of responsibilities for executing various part of the policy.* The definitions of responsibilities state who is responsible for what aspects of the policy. Most of these will be for non-technical positions. Think about the role non-technical managers must play in educating their staff and ensuring that they follow accepted practices. Include human resources, internal auditing, and legal, as these

functions all have roles to play in defining and upholding the security policy. At the minimum, system administrators and security officers should be responsible for configuring systems securely. They are also usually responsible for monitoring audit logs, creating user profiles properly, and evaluating the security features of third-party software.

- *Provisions for maintaining and approving the policy.* As a living document, the security policy must provide for some process for changing the policy, approving the changes, and getting the most current version out to all employees.

Most of the Web sites that have policy information are slanted toward institutes of higher education. My favorite is authored by a fellow from the University of New Mexico named David D. Grisham. He has put together a comprehensive list of links to security policies at universities around the world at www.unm.edu/~dave/. USENIX, a UNIX-oriented user group, has a sample policy posted at www.usenix.org/sage/publications/policies/fr_template.html. The USENIX policy is really designed for midsize UNIX shops, so it should be pertinent to many AS/400 installations despite some technical differences. Finally, the U.S. government provides a sam-

Kisco

ple Internet use policy at csrc.nist.gov/isptg/. I like using the format of this policy as a standard; It's very clean and well-organized and distinguishes between organizational policies and technology-specific policies. The organizational policies deal with broad issues, such as business communication and publicity policy. The technology-specific policies apply the organizational policies to specific areas, such as Internet email policy. Best of all, you can download this policy in a variety of formats, including Word 97, making it easy to use this document as a template for your organization's policy.

Vincent LeVeque is a senior security engineer for Science Applications International Corporation (SAIC). He can be reached at vleveque@earthlink.net.

REFERENCES AND RELATED MATERIALS

"Security Patrol," Wayne O. Evans, MC, September 1998

WRITE IN # ON READER SERVICE CARD OR GO TO WWW.SOLUTIONSCTR.COM