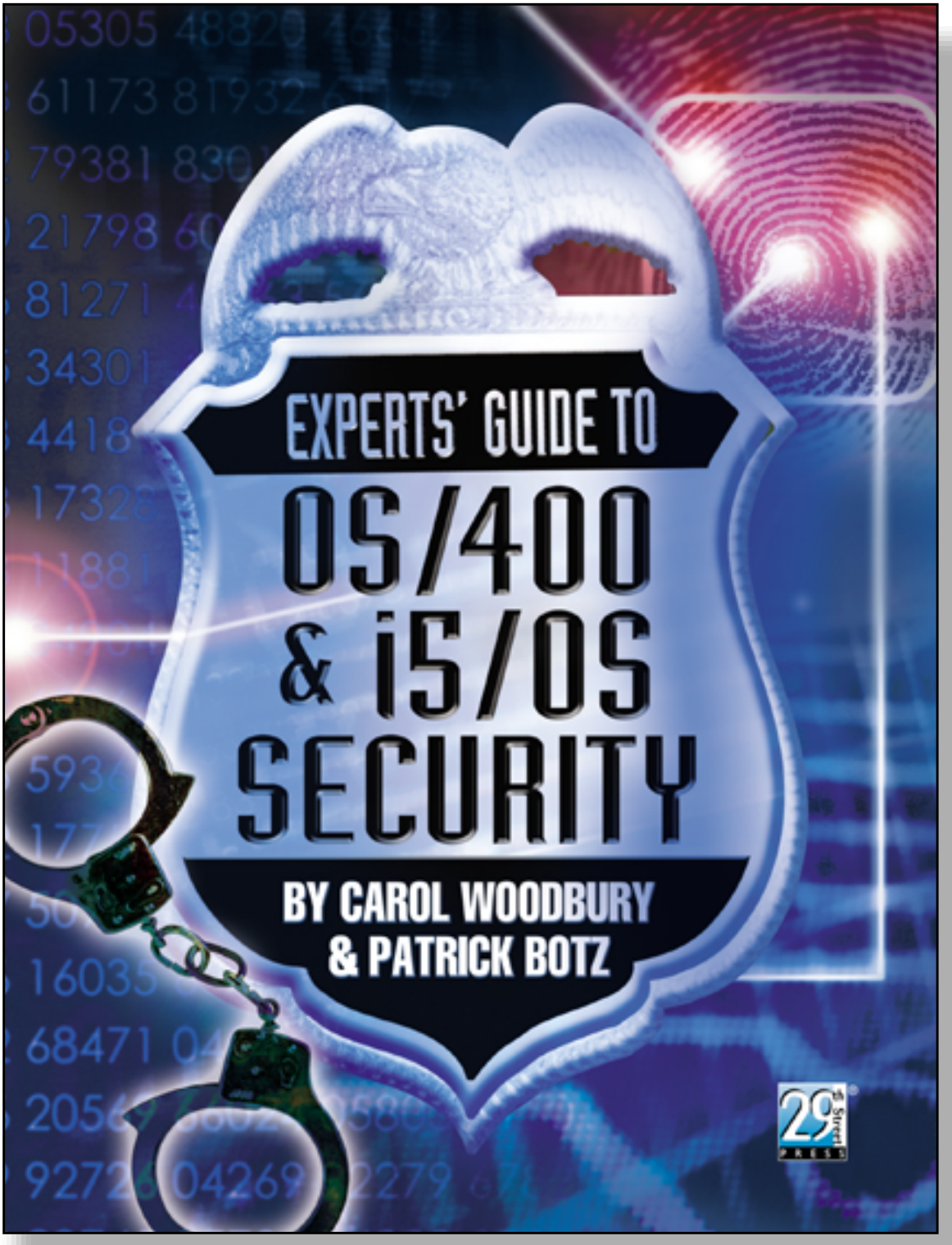
EXPERTS' GUIDE TO
OS/400
& i5/OS
SECURITY
BY CAROL WOODBURY
& PATRICK BOTZ
29th Street
PRESS

Chapter 12

Single Sign-on

DOES THE FOLLOWING SECURITY SCENARIO SOUND FAMILIAR TO YOU? YOU ARRIVE at work in the morning and sign on to your workstation using your workstation user ID and password. Then you start up your Lotus Notes client and supply your Lotus Notes password to unlock your Lotus Notes certificate, which contains your Lotus ID. A little later in the morning, you start up iSeries Access for Windows and supply your OS/400 user profile name and password. Still further into the day, you point your browser at an internal human resources Web application, and, of course, you have to supply your company ID and password.

Virtually everyone working today finds themselves in a situation similar to the one we've just described. In fact, one industry estimate suggests that the average worker has 15 user IDs and passwords with which to contend on a daily basis. In this chapter, we look at a solution provided by OS/400 to the problem of proliferating sign-ons: the OS/400 single sign-on (SSO) strategy.

The solution we describe here is intended to solve the problems associated with the multitude of user IDs and passwords that your users must remember. What causes this quagmire of user ID-and-password combinations is something Pat calls the "multiple user registry problem." By thinking of the situation in this way, you'll find OS/400's unique approach to addressing the issue easier to understand.

Before we go any further, we need to stop and define some terms. You may already know and understand these terms, but if you don't, you'll be lost when we describe the problem and its solution. So, here's a quick sign-on vocabulary lesson.

Sign-on Terminology

The term *identification* refers to the way a particular user is known or identified to a system or application. You can think of identification as "who someone claims to be." PAT, PSB, BOTZPS, and 93875613, for example, are all possible identifications for Patrick Botz.

Authentication is the process of proving that a user really is who he or she claims to be. The most popular form of authentication is a password or passphrase. Other forms include digital certificates, fingerprints, retinal scans, and other measures. Authentication can't happen without identification. In fact, people often use the term "authentication" alone to indicate both the identification and authentication processes. For example, you might say, "We use user IDs and passwords to authenticate users," rather than, "We use user IDs and passwords for identification and authentication."

Authorization is the process of determining which users are allowed to access which objects and for what purposes. Authorization requires a successful authentication.

A *user registry* is the set of users who are known to and/or trusted by a particular operating system or application. User registries contain the information by which people are identified and authenticated to computer systems and applications. On OS/400, the user registry is the set of all OS/400 user profiles.

User registries often, but not always, have an associated *authorization mechanism*. A system or application uses an authorization mechanism to determine whether an authenticated user is permitted the requested access to a specified resource (e.g., a data file, a program object, a directory). This is the way computer systems have worked since computer security was invented. The authorization mechanism for the OS/400 user registry is built into OS/400 itself. Not all computer systems contain this level of integrated authorization.

The Problem of Multiple User Registries

Unfortunately, every operating system — indeed, every instance of every operating system and many applications — defines and uses its own user registry and authorization mechanisms. Because operating systems and applications have had to do things this way, every entity with a heartbeat in an organization ends up with multiple user IDs and passwords! This is the multiple user registry problem.

The existence of multiple user registries severely impacts three classes of users: administrators, end users, and application programmers. Administrators have to manage (e.g., create, change, delete) all the user IDs and their associated passwords. End users must try to remember their own IDs and passwords and maintain their passwords everywhere. Programmers who create applications that run in multi-tiered environments have to write a lot of extra code just to deal with the authentication part of the problem. That's even before they start dealing with whatever business problem they're trying to solve.

The multiple user registry problem is, in our opinion, the fundamental issue limiting the widespread availability of true distributed computing. Although not the only obstacle to be overcome in distributed computing, it is the first one that needs to be addressed. The way in which this problem is approached affects the entire design and implementation of distributed applications.

In the remainder of this chapter, we describe the OS/400 single sign-on strategy, a solution to the problem of multiple user registries. We also show you how to plan an implementation of SSO in preparation for setting it up it in your environment. In the appendix at the end of this book, you'll find a step-by-step guide to the SSO configuration process.

Technical Note

If you're new to single sign-on, you might want to skip this chapter's discussions related to enhancements in i5/OS V5R3. Read the rest of the chapter and even the appendix (which covers configuration) first. Once you're familiar with the concepts, come back and read the information about the enhancements.

OS/400's Single Sign-on Strategy

The OS/400 approach to single sign-on and the multiple user registry problem is really two-pronged. The first prong is the enhancement of several important OS/400 operating system interfaces to be able to use the same authentication mechanism used in Microsoft Windows 2000 domains. The authentication mechanism used when a workstation participates in a Windows 2000 or XP domain is called *Kerberos* (after Cerberus, the three-headed guard dog of Hades in Greek mythology). The Kerberos mechanism enables transparent client/server and multi-tier authentication for the end user. Beginning in V5R2, several OS/400 system interfaces support Kerberos as an optional authentication mechanism.

The OS/400 SSO strategy requires the client system to use Kerberos. Windows 2000 and XP workstations use Kerberos by default if they participate in a Windows domain. Windows NT and standalone Windows 2000 and XP workstations don't use Kerberos by default, but they can be configured to use it. All IBM eServer platforms support Kerberos. All Linux distributions contain Kerberos and a pluggable authentication module (PAM) for exploiting it. Most Unix platforms are shipped with Kerberos. And free implementations of Kerberos are available on the Internet; the Massachusetts Institute of Technology (MIT), the inventor of Kerberos in the early 1980s, is one source.

Using the Windows domain authentication mechanism for OS/400 interfaces and OS/400 applications is great for users because each user needs to know only his or her Windows ID and password. The approach helps administrators because only the Windows ID requires a password. But there's still a problem: A person's Windows domain user name can't be used to determine which OS/400 objects the user is authorized to access! This is the problem that the second prong of the OS/400 SSO strategy is designed to solve.

The second prong of the OS/400 SSO strategy is a new technology called *Enterprise Identity Mapping (EIM)*. EIM is not a solution by itself but an enabling infrastructure for applications and operating systems. EIM enables a program to find the user ID in OS/400 that's associated with the Windows domain user name that a user provides for authentication.

In fact, EIM provides more general function: It can be used to find any associated user ID in any user registry, given a known user ID in some other user registry. Once the associated OS/400 user profile name is known, the application can swap to that user profile. When the application accesses any objects, OS/400 uses the "swapped to" user profile to determine whether the user has the appropriate authority.

Benefits of SSO

The OS/400 approach to SSO provides advantages for all affected classes of users in an enterprise. End users, of course, gain a single sign-on environment. The advantages aren't necessarily as obvious for administrators, but they are significant. In the SSO environment, the OS/400 user profile password isn't used for authentication. Users must still be authenticated to OS/400, but they use the Kerberos authentication mechanism, not OS/400. Administrators can therefore choose to set passwords to *NONE!