Print Article Close Window

From: www.cio.com

Why CRM Security Is Always a 'Role'-Your-Own Project

– David Taber, CIO

April 03, 2013

CRM systems are the most political of all enterprise software applications. This is partly because of the user community; sales and marketing play politics for a living. It's also due to the data in the system, as a single CRM table—the forecast—can make or break careers. Exactly who sees what, and when, is a key issue in organizations of any real size.

Tips: How to Get Smarter About CRM Security

Every CRM system has a slightly different security model, but there are general principles that apply everywhere. This article focuses on the Salesforce.com access control and privilege model. It's complicated enough to be frequently misunderstood—which makes it a good example for illustrating key issues.

Core Privileges: Start With Exclusions, Then Add Exceptions

Salesforce security has several overlapping layers that focus first on exclusion, then on adding access as exceptions to the exclusion. In small systems, most of the filtering is turned off. This can lead to some nasty consequences, so larger organizations must use every filtering mechanism available to keep data quality up and compliance issues down.

The first exclusion filter is specified by the class of user, or profile, and is applied at the object (table) level: Is a given profile allowed to see an object at all, and, if so, what operations are generally allowed? In addition to create, read, update and delete, there are privileges to "change record owner," "be allowed to see any record of this object, even if you shouldn't otherwise be able to" and "execute certain classes."

The next level of filtering is at the column level. This enforces three levels of access: Hidden, read-only and read-write. Profiles also define system-wide privileges (for example, export report data), but most aren't object-specific and therefore not a big deal.

Tips: Too Much Access? Privileged Identity Management Can Help

If you've created a bunch of customer fields and installed lots of applications in your CRM, this may mean you can set 1,000 attributes or more for each profile. Don't worry, it'll get more complicated soon enough.

To Avoid Broken Records, Sort Out Role Problems

The next layer of exclusion filtering is done by the user's role in the organization. Role hierarchy enforces a single Boolean attribute: Are you allowed to see an individual record, based on who owns the

record? You can't see what your siblings (peers) or superiors own. If you can't see a record, it's literally invisible to you and any code that's executing under your user identity.

If you turn on roles in a system that wasn't using them before, all kinds of records just aren't there any more. Why? The system is now enforcing record ownership, and many of your records now owned by users who are inactive, who no longer work for the organization, who have no defined role or who have changed roles.

How-to: Avoid 3 CRM User Identity Mistakes

Depending on the specifics, records owned by users with role problems may be visible to the wrong groups, or they may not be visible to anybody at all. This causes all kinds of fun in reporting, workflows and other automation that's supposed to make things easier. The malfunctions will continue until the record ownership is brought up to date with the current role hierarchy.

The good news: This particular data repair—fixing who owns which record—requires updating just one field per CRM record. The not-so-hot news is that some data repair needs to be done every time there's a sales reorganization or staff change. The solution: Bake some data repair into your work plan at least once a quarter.

CRM Record Sharing Can Save the Day

Because the effect of roles are binary, they must be supplemented with sharing rules that amend record visibility. For example, you don't generally want sales representatives editing each others' deals, but you *do* want to be able to do load-shedding when a sales rep is on vacation.

Sharing rules are exception handlers that simply say, "Under these special conditions, some users will be able to see records that would be otherwise hidden from them." Although each rule has only a few input parameters and a single yes/no output parameter, large organizations may have hundreds of them—and, you guessed it, they need to be updated whenever there is a significant org-chart or business-charter change.

Additional CRM Security Mechanisms Will Vary

In Salesforce, several system elements—in particular, record types—are inherently sensitive to profiles, while several other areas of the system can be configured to be sensitive to roles, groups, teams and queues. Applications and custom code added to Salesforce may add their own privilege management systems for individuals and collections.

As is the case with distributed systems, cloud administration doesn't offer a single administrative master control system. You have to build and maintain your own expertise around your system's peculiarities and expect that a major part of the system administrator's job is troubleshooting and fixing security issues on an ongoing basis.

Although lots of great tools can help inspect and manipulate many of the security attributes in your CRM complex, there is no "God's eye view" of the system. In my humble opinion, there's no hope for one, though we have seen elaborate tools purpose-built to coordinate change across multiple system elements.

The only way to know *every* area of the system that might be affected by a change, then, is to build your own administrative manual. This is best done as a crowdsourced effort, *allowing only the truly knowledgeable to write* and measuring them for frequency and length of their contributions.

David Taber is the author of the new Prentice Hall book, "Salesforce.com Secrets of Success" and is the CEO of SalesLogistix, a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel and India. Taber has more than 25 years of experience in high tech, including 10 years at the VP level or above.

Follow everything from CIO.com on Twitter @CIOonline, Facebook, Google + and LinkedIn.