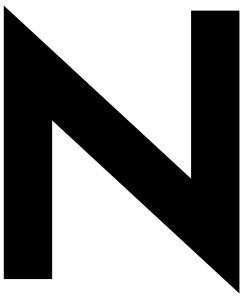
Novell® iManager:

Planning Security for Delegated Administration

www.novell.com

TECHNICAL WHITE PAPER



Novell®

Table of Contents

Synopsis

This paper discusses important security considerations in using Novell® iManager, the Novell eDirectory™ management console, including how to implement delegated administration securely. Also briefly discussed is the importance of implementing an eDirectory auditing solution whether an organization is using iManager for delegated administration or not; and how proxy-based consoles can possibly compromise security by adding a separate access control model from eDirectory, thereby requiring separate auditing.

DELEGATED ADMINISTRATION EXPLAINED

As information services continue to become more pervasive and more important to doing business, the need for delegated administration of information services has become increasingly important. Tasks such as changing passwords, managing printers and providing appropriate access to business systems are being thought of less as *technical* functions and more as *business* functions; therefore, the people who need to perform these tasks now come from a variety of departments, both inside and outside the IT organization.

The classic network administrator performed all tasks on the network, and the utilities he or she used were designed for use by someone holding his or her highly specialized, technical role. Management applications evolved according to this model, providing a global management view with the assumption of a highly technical user.

Delegated administration enables an organization to be able to assign individuals with specific management tasks and duties. The idea of delegated administration is not new, but few administrative utilities make the delegation of administrative tasks practical.

There are two important aspects to delegating authority to perform management tasks. The first is *authorization*, which has to do with assigning the access rights that allow an individual to perform the task you wish to delegate. The second aspect is *interface*, which provides the utility from which to perform the task.

Although there are many authorization systems that make delegating access rights possible, and there are many administrative utilities that provide a reduced interface to focus on one or a few specific tasks, the actual act of assigning tasks and delegating authority to perform them has traditionally been difficult and complex.

Novell iManager greatly simplifies the delegation of management tasks by facilitating both authorization, using the native access controls in Novell eDirectory, and interface, providing a task-oriented paradigm for limiting which

management interface components the user will see according to his or her role.

HOW iMANAGER ENABLES DELEGATED ADMINISTRATION

Novell designed iManager to use a task-oriented model that allows people to quickly execute common management tasks. Each task interface is therefore targeted to the task it is supposed to help accomplish. This means that each task provides a restricted interface. Examples include: Create Printer, Unlock User Account, Change Password and Configure Driver.

Novell iManager uses this task-oriented model to make delegated administration possible through an architectural feature called Role-Based Services (RBS). RBS allows the grouping of similar tasks into roles that can then be delegated to users. Custom roles can be quickly created as needed for any organization, and those custom roles can then be used to organize both existing and custom tasks. In this way, RBS follows a task-oriented, role-based management model that can be specifically tuned for delegated administration in any business environment.

Role-Based Services simplifies the typically complex interface of a comprehensive management tool and simultaneously simplifies the traditional complexity of assigning access rights required to perform management functions.

ACCESS RIGHTS AND NOVELL eDIRECTORY

For years Novell eDirectory has led the industry in directory scalability, performance and most important security. *Authorization*, which is also known as access control, is the security concept concerned with who or what can access, interact with, modify, create or delete other entries within the directory (and in some cases outside the directory).

Novell eDirectory, like its competitors, uses Access Control Lists (ACLs) to track and enforce access rights. Each object in the directory has an ACL, making granular security possible for each object. In addition to the static Access Control List for each object, eDirectory also provides dynamic inheritance of rights, allowing rights assigned on a container object's ACL to apply to all objects within that container. This feature, which is unique to and patented for Novell eDirectory, simplifies rights management and greatly reduces both replication traffic and directory storage requirements.

Of interest to the topic of delegated administration are the *object rights* used in eDirectory, and how they relate to inheritance. Object rights allow or prohibit specific actions to be performed on directory objects. The eDirectory object rights set is: Create, Delete, Browse, Rename and Supervisor. When a user has been granted the Create right to a container within the directory, he or she may create new objects within that container. Likewise, the Delete right will allow him or her to permanently remove objects from the container.

Note: *A more complete explanation of eDirectory access control and rights management, can be reviewed online at:* **http://www.novell.com/documentation/lg/edir87/edir87/data/fbachifb.html**.

Despite having one of the best directory security models in the industry, there are still important limitations to understand about access control capabilities within eDirectory. For example, the Create right does not restrict *what* can be created. A user with the Create right in a container can create *any* type of object that has been defined in the directory's schema. (*Schema* defines the types of objects that can be created in your tree—such as Users, Printers and Groups—and what information is required or optional at the time the object is created.) The generic nature of object rights such as create and delete becomes very important in managing delegated administration.

DELEGATION OF AUTHORITY FOR DELEGATED ADMINISTRATION

Through RBS, iManager consistently adheres to the eDirectory security model for enabling delegated administration. Each time a user is assigned to occupy a given role, RBS assigns the user the appropriate rights required to accomplish each task associated with that role. For example, a role called User Management may contain tasks to create, delete and manage user objects. When you assign a user to occupy the User Management role, you also have to set a *scope* for the assignment. The scope defines that the user can only perform the role's tasks in a specific part of the directory. Behind the scenes, RBS grants the appropriate access rights for the user using standard eDirectory Access Control Lists. This means that iManager uses eDirectory access rights consistently with other applications for determining who can perform specific management tasks and where they can perform them. By coupling together the delegation of access rights with the delegation of tasks, iManager brings together both the authorization and interface aspects of delegated administration, making it practical within the enterprise.

SECURITY RISK BY INTERFACE IMPLICATION

Without a clear understanding of how to practically use delegated administration within iManager, there is a potential security risk.

As stated in the previous section, iManager can automate the assignment of appropriate access rights (this is made optional in version 2.0 of iManager). This feature simplifies the chore of determining and assigning the access rights required to perform a specific task. Another function of iManager's Role-Based Services is to restrict the management options presented to a user, thereby simplifying the user's experience and constraining the user's rights in working with iManager. It is at this intersection of two simplifications—the interface and authorization systems—that it is possible to expose your organization to a security risk.

Consider delegating to a user a single role containing the tasks for creating, managing and deleting users. Among the access rights granted with this role, the assigned user will gain the Create and Delete object rights. The user will inherit these assigned rights, starting from the container where the assignment's scope begins, down to all objects within that container and further to any subordinate containers. If the user

has only been assigned to this one User Management role, the iManager interface will show him only the tasks contained in that role. However, i*Manager is not the only utility for managing eDirectory*. And because the Create and Delete rights granted by the above role assignment are generic, the user can create or delete *any* object with other utilities.

So what are those other utilities? For starters, Novell eDirectory can be managed through a variety of freely available LDAP utilities and simple LDAP command lines. There are also older console utilities from Novell, such as ConsoleOne®, NetWare® Administrator and the old text-based NetAdmin. Novell eDirectory will faithfully enforce access according to the rights given to the user performing the action—but outside of iManager, the interface is no longer restricted. In the example we described above, anyone assigned to occupy the User Management role could use a LDAP utility or ConsoleOne and create or delete *any* object type below where his rights were scoped by the role assignment.

Let's summarize all that and try to state it as simply as possible: Although iManager presents a restricted interface, *it is the access rights assigned to a user that determine what a user can do in the directory*.

Again, this is completely consistent with the historical access control capabilities of eDirectory. (In fact, those who have extensive experience with eDirectory will be saying "No Duh" to this lengthy explanation.) *But the restricted interface that iManager facilitates implies that there is a greater security restriction than what eDirectory is actually capable of providing*. (And for that matter, no mainstream directory service available today actually provides such granular security.)

Another way to look at this is that advances in the iManager interface have outpaced the eDirectory security model; therefore, it is very important to understand the access rights granted by any task you wish to delegate. Fortunately, there are some easy ways you can implement delegated administration to avoid any such security exposure.

SECURELY DEPLOYING DELEGATED ADMINISTRATION WITH iMANAGER

There are several ways to ensure that your implementation of delegated administration through Novell iManager does not create a hole in the security of your directory. In this paper, we will explore three possible solutions: using containers to limit security risk, manually assigning access rights and using a separate directory tree for administration while synchronizing it to your primary directory tree.

Using Containers to Restrict Access Rights

The easiest way to leverage the delegated administration capabilities of Novell iManager without granting excessive rights is to create specialized directory containers for delegated management. The good news is that you are quite possibly already doing this. Let's use a couple of examples to explain this idea.

Suppose you want to have a role for printer administrators and another for user administrators,

but you want to ensure that when you assign users to the two roles the users do not get rights to manage objects they should not be authorized to manage. First we'll discuss how to do this the wrong way; then we'll cover how to do it the right way.

Wrong way: Put both printer objects and user objects in the same container. By doing this, you multiply your security management by the combined number of objects and create a logistical nightmare for yourself. Why? Because your role for user administrators grants rights to users starting at the container level and allows Create and Delete, among other rights. Your role for printer administrators will do pretty much the same, despite the fact that it restricts the iManager interface to show only the printer management tasks.

User administrators can now use another management utility such as ConsoleOne to delete or modify printer objects. Printer administrators can do the same to delete or modify user objects.

Right way: Create a specialized container for printers. Do the same for users. (Many administrators who use products like Novell ZENworks® already practice this, creating separate containers for such objects as applications, policy packages and workstations.)

With such containment, you can scope your role assignments to the proper containers, so rights are restricted according to each container's purpose.

With this implementation, a potentially malevolent user who occupies one of the roles could still use another utility besides iManager, but the worst he or she can do is create an object type that does not belong in the container. For example, a printer administrator could create a user or group object inside the container for printers. Since he cannot assign that object any more rights than he or she already has, the object may present an annoyance, but presents no serious security risk.

Manually Assigning Access Rights

The second option for controlling security when delegating administrative tasks is not to use Role Based Services for assigning rights. By delegating a role and clearing the "Assign Rights" checkbox option, the assignment will not grant any access rights to the assignee. You can then manually assign access control rights on the appropriate objects for the users or groups that you want to be able to manage those objects.

Although this makes management of rights a bit more tedious, it allows you to take full advantage of the traditional granularity in managing access control in eDirectory. The "Assign Rights" checkbox is only available in Novell iManager 2.0 and later.

Implementing a Separate Directory Tree for Delegated Administration

If you need to step up the security of delegated administration further, you might consider implementing a separate eDirectory tree for delegated administration or even multiple trees. Novell Nsure™ solutions, and more specifically DirXML®, can synchronize appropriate changes to your primary eDirectory tree.

In such an implementation, you do not delegate any rights in your primary eDirectory tree. All roles are assigned in a separate tree created specifically for delegated administration. This also allows you to define a completely different hierarchy from your primary tree. Because DirXML allows you to define rules for what data is synchronized—down to individual object attributes—you can use DirXML as a rules enforcement engine for delegated administration.

For example, perhaps you want to allow helpdesk administrators to reset account lockouts and passwords and disable accounts just for users who are the finance department. But unfortunately all your user accounts from all departments are in a single container in your primary tree. If the only thing to distinguish a finance department employee from all the others is the Department attribute on the user account, then DirXML could synchronize only accounts whose Department equals "Finance" into a container in a separate administration tree. Now the helpdesk users can be assigned a role that allows them to perform the desired tasks only in the container of finance users in the administration tree.

In the previous example, you can even delegate tasks that include more permissive object rights such as Create and Delete; because even if a user creates an object that is not allowed, DirXML will not synchronize changes that it has not been explicitly configured to. This implementation therefore completely filters any erroneous object classes and out-of-policy attribute changes.

AUDITING AND SECURITY: ENFORCING TRUST FOR DELEGATED ADMINISTRATION

There is an upper limit for any security model: A system is only as secure as the trustworthiness of the empowered users of that system. That might be system administrators, helpdesk employees, resource managers or secretaries.

The three A's of security have long been known as authentication, authorization and administration. Authentication is the verification that someone is who they claim to be. Authorization is the process of allowing or restricting someone to perform an action according to security policies. Administation involves the management of those security policies and access rights. However, another critical aspect of security, and what might be considered a fourth A, is auditing. The process of monitoring and tracking users' activity in internal systems and applications.

In the world of delegated administration, where management rights are distributed to a much wider group of users, auditing is key. That is to say, the more people to whom you provide administrative access, the more critical it becomes to know who has done what, where did they did it and when.

A number of solutions are available for auditing eDirectory. As part of the Novell Nsure solution, Novell Nsure Audit offers an extensive auditing system for your eDirectory environment.

POSSIBLE ALTERNATIVE: PROXY-BASED ARCHITECTURES

To effectively audit Novell eDirectory, audit solutions must be implemented within the directory itself,

as opposed to within an application that accesses the directory. Because there are so many applications that can access the directory, the only practical way to effectively audit eDirectory is to consistently follow the directorys' security model. (This point can be debated around and around, but the arguments fall outside the scope of this paper; suffice it to say that, ultimately, the debator who holds the position that audit must happen within the directory itself will win the Tightest Security Model award.)

Many applications for delegated administration use a proxy-based model. A proxy-based system uses 'alternate identities' when modifying the directory. By doing so, such tools implement their own security model at the application level, obviating the built-in security of the directory. While this can provide a method to quickly enable delegated administration, it introduces a separate security model; so it also introduces another point for auditing, separate from eDirectory. If effective auditing is a concern for an organization, then introducing yet another point from which to monitor security would be an undesirable option.

For this reason, iManager does not use a proxy-based model today; however, Novell is researching how to allow organizations to implement proxy-based delegated administration as an optional alternative. This alternative, though, would require the integration of a common auditing system between iManager and the underlying directory, in order to eliminate introduction of another security point to monitor. Such a solution would also require configurable proxy access control that does not rely on having a single "super user" proxy account.

A LONG-TERM VIEW OF DELEGATED ADMINISTRATION AND NOVELL eDIRECTORY

This paper has covered an overview of the security models used by eDirectory and iManager, discussed considerations for delegated administration and examined some practices that can provide a more secure environment for delegated administration.

As stated previously in this paper, Novell iManager has outpaced the Novell eDirectory authorization model. Without a proper understanding of the interplay between eDirectory security and the iManager console, it is possible to compromise the security of an organization's eDirectory implementation. The intersection of simplifying access control rights and administrative interfaces have brought to light some enhancements that need to be made in the eDirectory security model, despite its standing as the industry's best.

Because interest in delegated administration continues to grow, Novell is actively planning the future of both iManager and eDirectory, wherein the security model of eDirectory will allow the finest granularity of access control, based on rich policy that includes conditional factors such as time or location of access, and features application-specific access policies in addition to traditional user-based Access Control Lists.

Ultimately, there is no "one-size-fits-all" plan for IT security. Any organization planning to implement delegated administration must carefully

plan its security architecture, regardless of the tools it will use. With a clear understanding of the Novell eDirectory access control model, organizations can effectively plan and implement a solid system to provide delegated administration with Novell iManager.

Novell Product Training and Support Services

For more information about Novell's worldwide product training, certification programs, consulting and technical support services, please visit:
www.novell.com/ngage

For More Information

Contact your local Novell Solutions Provider, or visit the Novell Web site at:
www.novell.com/products/edirectory

You may also call Novell at:

1 888 321 4272 US/Canada
1 801 861 4272 Worldwide
1 801 861 8473 Facsimile

Novell, Inc.
1800 South Novell Place
Provo, Utah 84606 USA

www.novell.com

Novell

462-001360-002