IFS in Review

by Carol Woodbury, President, SkyView Partners, Inc

Many of you are daunted by the security aspects of the Integrated File System (IFS), but there are a few things that—whether you're comfortable or not—you should not ignore.

Sharing Root

One function that's available in the IFS is the ability to create a share. A file share allows you to map a drive and make what is shared available directly as a mapped drive on your PC or on a network server. File shares are not typically a security risk as long as the share is mapped at the directory whose contents are being shared. What is a risk is when root is shared. That's because, when root is shared, all of the /QSYS.LIB file system is also shared; in other words, all libraries are also shared and available for manipulation in Windows Explorer. This becomes a huge security risk, especially if you have not implemented good access controls on your database files. Why? Because the files can easily be overwritten with garbage or deleted by dragging and dropping them into the trash bin.

You can somewhat reduce the risk of sharing root by adding a dollar sign ($) to the end of the share name. This prevents the share name from being broadcast. Unfortunately, most people just add $ to the word root, as in root$ is the share name. Obviously, this is totally unimaginative and very easily guessed. If you're going to attempt to hide the share name, use a non-obvious name! Another way to add some protection is to use the QPWFSERVER authorization list that's shipped with the operating system. Users with authority to this list are able to see libraries in iNavigator as well as lists such as those presented in Windows Explorer. However, if the user has no authority to the list, then the QSYS.LIB file system (that is, libraries) will be hidden from these views. This has no effect on the actual authority the users have to these libraries; it's just a control for who can see them in this "list" view. The default *PUBLIC authority of the QPWFSERVER authorization list is *USE. Change it to *EXCLUDE to eliminate non-*ALLOBJ users' view of libraries. Authorize users or groups to the list if they have a business need.

One last thought on sharing root. A few weeks ago, someone emailed me about a client who had been infected with the CryptoLocker malware. It had infected not only a user's PC, but because the user was mapped to a portion of the IFS, it also encrypted those images. They were able to recover by restoring what had been encrypted from backup media. But what sends shivers down my spine is the thought of what could have happened if the user had been mapped to root. I know that many of you have mapped to root out of share convenience. It's easy, and that way you have access to everything that you might need. But I'm hoping that this true story causes you to consider why that might not be such a good idea.

Root Left at the Default Public Authority

When IBM installs the operating system, it sets the *PUBLIC authority to root to have data authorities (DTAAUT) of *RWX and object authorities of *ALL. This is the equivalent of *PUBLIC *ALL. The effect of this *PUBLIC authority setting allows your users and vendors to create a new subdirectory whose *PUBLIC authority will also be *ALL. In addition, users can add files directly to root instead of putting them in their own subdirectory. If you're still not getting the importance of this setting, think of it like this: it would be similar to allowing anyone to create a library with a *PUBLIC authority setting of *ALL and allowing people to create files into the QSYS library. User- and vendor-created objects should be in their own library, right? It's the same with IFS objects.

The recommended *PUBLIC authority setting for root is DTAAUT(*RX) OBJAUT(*NONE), which is the equivalent of *PUBLIC *USE. This allows users and processes to traverse through the root directory to the appropriate subdirectory, but they cannot create a new subdirectory or place an object directly into the root directory. Before changing the *PUBLIC authority, however, you'll want to check for existing processes that may already be creating objects directly into root. The best way to detect that is to examine the audit journal for create of object (CO) entries. You'll have to look in the pathname field to see the IFS path the object is being created into.

Guest Profile Assigned to the NetServer

The NetServer on IBM i allows the system to be used as a file server. Great function, but there's one feature that should be avoided and that's assigning a guest profile. Assigning a guest profile to the NetServer allows anyone to connect to the system without having a profile. They connect with the authority of the guest profile. Again, if you've never implemented object-level security, and users connect to the root share, they will be able to gain access to not only information in directories, but also objects in libraries (yet another reason not to share root). The other issue is that when multiple people connect this way, accountability is lost. All entries in the audit and database journals will be logged as the guest profile. The order in which you define user profiles is important only from a performance point of view. The system checks the authority in the order the groups are defined in the user's profile. Put the most-used group at the top of the list for the speediest authority check.

To determine if a guest profile has been assigned to the NetServer, open iNavigator, open the system, then go to Network > Servers > TCP/IP. Right-click on NetServer, choose Properties, and click on the Security tab. If the Guest user ID field is blank, there is no guest profile. If there's a profile named, that's the guest profile. To remove it, click Next Start and blank out the field. The guest profile has now been removed and is no longer in effect. If, when you right-click on NetServer you choose Open instead of Properties, you can click through the connections and see if there are any users currently connected using the guest profile. Using the IP address provided, you can contact the users and educate them on mapping to the system using their own profile rather than the guest profile.

Ownership and Private Authorities

Mismanaged ownership of directories can cause excessive private authorities. This will, in turn, cause your Save Security Data (SAVSECDTA) to run longer and longer. (SAVSECDTA is what saves private authorities.) Making sure that the same profile owns the directory and subdirectories in a path where many objects are being created will help eliminate excessive private authorities. You can also have the profile that runs the process of creating objects into a file own the subdirectory the objects are being created into to help avoid excessive private authorities in the IFS.

Final Thoughts

I hope that, if you have been avoiding the security aspects of the IFS, this article has made you realize that you need to take action. If nothing else, determine if your systems have any of the issues described. For those of you actively securing the IFS, I hope this article has served as a good refresher for some of the items you need to be working on.

How SkyView Partners can Help

If you know that you should be checking your security configuration more often to make sure that inadvertent changes haven't slipped through the cracks but you just don't have the time, SkyView's Managed Security Services will monitor your configuration for you.

SkyView Partners' Security Check-up Service examines your IBM i, AIX or Linux partitions and provides the factual, unbiased analysis needed to determine risk to your data and to provide best practice recommendations and guidance.

SkyView's Services will help you address the list of vulnerabilities and reduce the risk to your data. Whether you want to address one specific issue (such as configuring SSL or moving your systems to a higher password level), work on issues over time or attack them all at once, the SkyView Services team will bring their security expertise and experience to ensure your projects are accomplished in a timely and successful manner.

Contact us for more information.

Carol Woodbury, CRISC
President and co-founder, SkyView Partners