

White Paper  
Does a firewall provide access control to the iSeries  
servers?  
By Boris Breslav – Senior Security Specialist at Bsafe  
Software Solutions  
October 2003

Today no one questions the essential need of the firewalls in server protection. Even when the server has it's own well-configured security, the primary role of the firewall is clear - prevention of unauthorized access. But is the firewall the whole solution? Is it adequate to guarantee a secure access to the system?

Let's first see what access control measures are available in the OS/400 itself. By access control I mean control over and above object authorizations, whether it is a lock on a console or sign on screen to the telnet.

OS/400 access control, an approach based on a combination of menus, initial programs and command line restrictions was for years a good solution. The OS/400 provided a sophisticated mechanism of system authorizations to enhance access control. However, maintaining OS/400 system authorizations was a complicated undertaking that many companies simply did not have the resources to handle. They were forced to rely on menus, initial programs and command line restrictions as the only security measure.

For such organizations, the situation is largely unchanged. OS/400 security is still one of the best, but most companies still cannot afford security specialists. The problem is that the security architecture not only has to be built, but also maintained properly afterwards. I'm not saying they should ignore the OS/400 security. On the contrary, they should make their best effort to build and maintain it. But, if they use only few host servers and a few TCP/IP services, do they really need to reconfigure, or in some cases implement from scratch, the OS/400 security or they can still mainly rely on the access control?

There are other situations when the access control can be very useful as an additional level of security. The value of OS/400 security without proper application design is severely limited. A poorly designed application can ruin any security effort - and I'm not even talking about applications that require \*ALLOBJ authority for all of it's users.

Take a more subtle example – an ODBC application using a file containing customer information. Sales representatives use this application to access information like customer phone numbers. This requires at least the \*USE authority to the customer file for the sales reps to access the file. A major problem is that it allows the rep to use a simple Receive File From Host function of the telnet emulation to transfer the customer file to his PC. He then has full access to the file's contents, including confidential data such as credit card information. The \*USE authority doesn't appear to distinguish between the two ODBC functions: the PREPARE AND EXECUTE, which is usually used by ODBC applications to run the SELECT SQL command, and the STREAM FETCH, which is used to transfer files.

There are several solutions to this problem. For example you can:

- redesign the ODBC application to use adopted authority, so the user operating the file transfer will not have the \*USE authority to the customer file
- redesign the ODBC application to use views or logical files, so the user operating the file transfer will be only authorized to use views and logical files and not the customer file, and therefore will have access to the Telephone field only
- database triggers can be used in certain situations to define authorized requests.

The only problem with the above solutions is that they require additional resources and development. You're lucky if you can still find this company that sold you the ODBC application and require them to comply with the security standards.

Even if someone is allowed to use or change the customer file, how can you ensure the file will never be transferred, copied or stolen by any other means from your iSeries. After all, the only protection for this file is user and password, which important as it is, is information that can be overheard, seen or otherwise delivered into the wrong hands. In case where the QSECOFR user profile is enabled, for example, all your valuable information is protected by this user's password! A string

of letters, let's say WER123Y, is the only key to the customer credit card information, no matter how comprehensive the OS/400 security authorizations are!

Social engineering attacks are treated by many intruders as the most efficient. All the intruder needs to do is call someone who has access to the system, pretend to be a new employee or some technician running tests on the new production application and ask for valid user and password. In this case, you still may want to make sure that the QSECOFR, or any other powerful user profiles, are not allowed to access the iSeries from the network. You can use access control as an additional level of security.

With this thought in mind I put OS/400 access control to the test. I created a user of class \*USER with limited capabilities. The Limited Capabilities is one of the access prevention controls supplied with OS/400. It prevents the user from running commands from the command line. However, some major OS/400 services appeared to simply ignore that rule. I was able to

1. Run an FTP command: quote crtlib lib(remote)
2. Use other FTP functions like get, put etc.
3. Use Client Access file transfer (database service)
4. Run a remote command : rmtcmd dltlib lib(remote)

I then tried another OS/400 access control - the QLMTSECOFR system value. I set this system value to 1 to restrict users with \*ALLOBJ and \*SERVICE authorities to sign on from various devices. I was able to logon to FTP and ODBC using the QSECOFR user profile. Of course, I could disable the QSECOFR user profile to allow this user to sign on only from the console. But how practical is that - and what about all the other users with powerful special authorities?

There is a huge extension to the access control in the OS/400 5.2. That is the FTP access control in the Operation Navigator, although it can be quite limited, in particular if you want more detailed control at the object level.

The Operation Navigator also provides Packet Filtering, which is very similar to the firewall.

That brings me back to the questions about the firewall. Can a firewall help to prevent unauthorized access? In some cases yes, in others no. You can use a firewall to restrict ports and IP address, but one problem with the firewall is that it cannot

maintain access control at the iSeries object level – user profiles, devices, jobs, system values etc. No matter whether it is a network, router or Linux firewall, there is no way it can recognize the QSECOFR user profile trying to logon to the ODBC application, unless it uses some sort of VPN connection, which is usually used for external communication. There is no way a firewall can recognize an FTP CL command request or an ODBC STREAM FETCH request trying to access the customer file, as in the above example.

However there is a second line of defense in the access control. OS/400 provides a number of points in the processing of different network requests, called exit points, at which custom-built programs may be inserted. Only here can you build in the necessary algorithms to protect your iSeries against unauthorized access of the type described in the above examples. In some cases customizing exit points seems to be the only solution, in others it is an essential additional security measure. It can enhance access control on the iSeries by:

1. “Preventing unauthorized access to iSeries services and functions”

If you have relied on menus, initial programs and command line restrictions and now opened services in your iSeries, like Telnet, Database or File Server, in most of the cases you can use the exit point solution to apply access control restrictions, based on user profiles, devices and IP addresses. A firewall can only partially help you do that.

If you want to allow the ODBC application to run SELECT commands on the customer file and not allow the customer file to be transferred to the PC, or if you want with one rule to block DELETE, INSERT, UPDATE and DROP ODBC requests, you have to use the exit point solution. A firewall cannot help you do that.

2. “Limiting access of those services and functions to specific iSeries resources like, libraries, files, devices and IFS paths”

If you want to restrict Telnet connections by a device name or FTP request to a single library, you have to use the exit point solution. A firewall cannot help you do that.

If you want to prevent the QSECOFR and other powerful users from accessing iSeries from the network, you have to use the exit point solution. And a firewall cannot help you do that, unless you use VPN access to the iSeries.

3. “Monitoring access to the iSeries resources”

If you want to maintain a meaningful log of the network activity, showing ODBC, FTP, File Server and other services functions or more specific information like ODBC application SQL statements, you have to use the exit point solution. A firewall cannot help you do that.

4. “Sending alerts based on the transaction level information”

If you want to receive an email saying that someone is trying to delete files using ODBC or to run the FTP CL command, you have to use the exit point solution. A firewall cannot help you do that.

In conclusion, there is no single, all-inclusive solution for iSeries security. The best possible solution would consist of a combination of OS/400 authorizations, customization of the exit points and the use of a firewall.