Deploying Network Access Quarantine Control, Part 1

by Jonathan Hassell
last updated August 4, 2004

One of the easiest and arguably most prevalent ways for nefarious software or Internet users to creep onto your network is not through holes in your firewall, or brute force password attacks, or anything else that might occur at your corporate headquarters or campus. It's through your mobile users, when they try to connect to your business network while on the road.

Consider why that is the case. Most remote users are only authenticated on the basis of their identity; no effort is made to verify that their hardware and software meets a certain baseline requirement. Remote users could, and do everyday, fail any or all of the following guidelines:

- The latest service pack and the latest security hotfixes are installed.
- The corporation-standard antivirus software is installed and running, and the latest signature files are being used.
- Internet or network routing is disabled.
- Windows XP's ICF, or any other approved firewall, is installed, enabled, and actively protecting ports on the computer.

You would expect your business desktops to follow policy, but in the past, mobile users have traditionally been forgotten or grudgingly accepted as exceptions to the rule. However, Windows Server 2003 includes a new feature in its Resource Kit, called "Network Access Quarantine Control," which allows you to prevent remote users from connecting to your network with machines that aren't up-to-date and secure.

How Network Access Quarantine Works

Network Access Quarantine Control, or NAQC, prevents unhindered, free access to a network from a remote location until after the destination computer has verified the remote computer's configuration meets certain requirements and standards as outlined in a script.

To use NAQC, your remote access computers must be running any one of Windows 98 Second Edition, Windows Millennium Edition, Windows 2000, or Windows XP Home or Professional. These versions of Windows support a connectoid, containing the connection information, the baselining script, and a notifier component, that can be created using the Connection Manager Administration Kit (CMAK) in Server 2003. Additionally, you'll need at least one Windows Server 2003 machine on the backend running an approved listening component; for the purposes of our exercise, I'll assume you're running the Remote Access Quarantine Agent service (called RQS.EXE) from the Windows Server 2003 Resource Kit. Finally, you'll need a NAQC-compliant RADIUS server, such as the Internet Authentication Service in Server 2003, so that network access can be restricted.

A Step-by-Step Overview of NAQC

Here is a detailed outline of how the connection and quarantining process works, assuming you're using RQC.EXE on the client end from the CMAK and RQS.EXE on the back end from the Resource Kit.

1. The remote user connects his computer, using the quarantine CM profile, to the quarantine-enabled connection point, usually a machine running the Routing and Remote Access Service (RRAS).
2. The remote user authenticates.
3. RRAS sends a RADIUS Access-Request message to the RADIUS server-in this case, a Server 2003 machine running the Internet Authentication Service.
4. The IAS server verifies the remote user's credentials successfully and checks its remote access policies. The connection attempt matches the configured quarantine policy.
5. The connection is accepted, but with quarantine restrictions in place. The IAS server sends a RADIUS Access-Accept message, including the MS-Quarantine-IPFilter and MS-Quarantine-Session-Timeout attributes, to RRAS.
6. The remote user completes the remote access connection with the RRAS server, which includes leasing an IP address and establishing other network settings.
7. RRAS configures the MS-Quarantine-IPFilter and MS-Quarantine-Session-Timeout settings for the connection, now in quarantine mode. At this point, the remote user can only send traffic that matches the quarantine filters-all other traffic is filtered-and can only remain connected for the value, in second, of the MS-Quarantine-Session-Timeout attribute before the quarantine baselining script

must be run and the result reported back to RRAS.

8. The CMAK profile runs the quarantine script, currently defined as the "post-connect action."
9. The quarantine script runs and verifies that the remote access client computer's configuration meets a baseline. If so, the script runs RQC.EXE with its command-line parameters, including a text string representing the version of the quarantine script being used.
10. RQC.EXE sends a notification to RRAS, indicating that the script ended successfully.
11. The notification is received by RQS.EXE on the back end.
12. The listener component on the RRAS server verifies the script version string in the notification message with those configured in the registry of the RRAS and returns a message indicating that the script version was either valid or invalid.
13. If the script version was acceptable, the RQS.EXE calls the MprAdminConnectionRemoveQuarantine() API, which indicates to RRAS that it's time to remove the MS-Quarantine-IPFilter and MS-Quarantine-Session-Timeout settings from the connection and reconfigure the session for normal network access.
14. Once this is done, the remote user has normal access to the resources on the network.
15. RQS.EXE creates an event describing the quarantined connection in the System event log.

Deploying NAQC

In this section, I'll look at the actual deployment of NQAC on your network. There are six steps, each outlined in separate subsections ahead.

Creating Quarantined Resources

The first step is to create resources that actually can be accessed while the quarantine packet filters are in place for a remote client. Examples of such resources include DNS servers and DHCP servers so IP address and connection information can be retrieved, file servers to download appropriate software to update out-of-compliance machines, and web servers that may describe the quarantining process or allow a remote user to contact IT support via e-mail if any problems occur.

There are two ways you can specify and use quarantined resources. The first is to identify certain servers on your network as quarantine resources, without regard to their physical or network location. This allows you to use existing machine to host the quarantined resources, but you also have to create individual packet filters for quarantined sessions for each of these existing machines. For performance and overhead reasons, it's best to limit the number of individual packet filters for a session.

The other approach is to limit your quarantined resources to a particular IP subnet. This way, you just need one packet filter to quarantine traffic to a remote user, but you have to re-address these machines and, in most cases, take them out of their existing service or buy new ones. Using this method, however, the packet filter requirements are much simpler. You simply need to open one for notifier traffic on destination TCP port 7250, one for DHCP traffic on source UDP port 68 and destination UDP port 67, and for all other traffic, the address range of the dedicated quarantine resource subnet. And again, you can also configure any other packet filters peculiar to your organization

Writing the Baselining Script

The next step is to actually write a baselining script that will be run on the client. This is really independent and unique to your organization, but all scripts must run RQC.EXE if the baselining compliance check was successful and include the following parameters:

```
rqc ConnName TunnelConnName TCPPort Domain Username ScriptVersion
```

The switches and arguments are explained in the following list.

- The ConnName argument is the name of the connectoid on the remote machine, most often inherited from the dial-in profile variable %DialRasEntry%.
- The TunnelConnName argument is the name of the tunnel connectoid on the remote machine, most often inherited from the dial-in profile variable %TunnelRasEntry%.
- The TCPPort argument is, obviously, the port used by the notifier to send a success message. This default is 7250.
- The Domain argument is the Windows security domain name of the remote user, most often inherited from the dial-in profile variable %Domain%.
- The Username argument is, as you might guess, the username of the remote user, most often inherited from the dial-in profile %UserName%.
- The ScriptVersion argument is a text string that contains the script version that will be matched on the RRAS server. You can use any keyboard characters except /0 in a consecutive sequence.

Installing the Listening Components

The Remote Access Quarantine Agent service, known otherwise as RQS.EXE, must be

The Remote Access Quarantine Agent service, known otherwise as RQS.EXE, must be installed on the Server 2003 machines accepting incoming calls using RRAS. RQS is found in the Windows Server 2003 Resource Kit Tools download, as found on the Microsoft web site. Once you've run the installer for the tools, select the Command Shell option from the program group on the Start menu, and run RQS_SETUP /INSTALL from that shell. This batch file will copy the appropriate binaries to the WindowsRoot\System32\RAS folder on your system and modify service and registry settings so that the listener starts automatically when the server boots up.

There is a bit of manual intervention required, however: you need to specify the version string for the baselining script. The listener service will match the version reported by the remote computer to the value stored on the RRAS computer to make sure the client is using the latest acceptable version of a script. To make this change manually after you've run RQS_SETUP from the Tools download:

1. Open the Registry Editor.
2. Navigate to the HKEY_LOCAL_MACHINE/System/CurrentControlSet/Services/Rqs key.
3. Right-click in the right pane, and select New String.
4. Name the string AllowedValue.
5. Then, double-click on the new entry, and enter the string that refers to an acceptable version of the script.

Alternatively, you can modify the RQS_SETUP batch file, so this step can be automated for future deployments. To do so:

1. Open the RQS_SETUP.BAT file in Notepad.
2. Select Find from the Edit menu.
3. In Find what, enter Version1\0, and click OK. The text cursor should be on a line that says:

   ```
   REM REG ADD %ServicePath% /v AllowedSet /t REG_MULTI_SZ /d Version1\0Version1a\0Test
   ```

4. To add just one acceptable version, delete "REM" from the beginning of the line.
5. Now, replace the text "Version1\0Version1a\0Test" with the script version string you want to be passed by RQC.EXE.
6. If you want to add more than one acceptable version, replace the text "Version1\0Version1a\0Test" with the acceptable version strings, each separated by a "\0" line.
7. Save the file, and then exit Notepad.

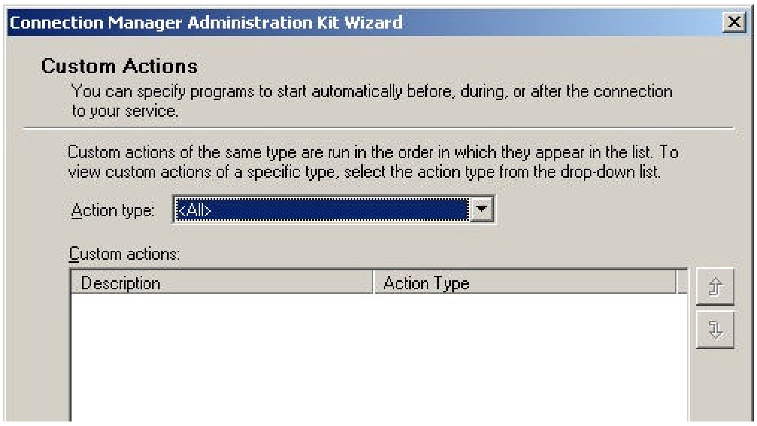
Two notes: RQS is set as a dependency of RRAS. However, when RRAS is restarted, RQS doesn't automatically restart, so you'll need to manually restart it if you ever stop RRAS manually. Also, by default, RQS.EXE listens on TCP port 7250. To change the default TCP port, navigate to the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\rqs\ key, create a new REG_DWORD value called Port and set it to the desired port.

Creating a Quarantined Connection Profile

The next step is to create a quarantine Connection Manager profile, which happens to be a plain-vanilla profile with a few modifications. For one, you need to add a post-connect action so that your baselining script will run and return a success or failure message to the RRAS machine. You also need to add the notifier to the profile as well.

In this section, I'll assume you're familiar with creating custom connectoids with the CMAK Wizard, since the whole process is beyond the scope of this article. Where the process begins to differ is at the Custom Actions screen, and I'll begin this procedural outline there.

1. Navigate to the Custom Actions screen, filling in previous screens as appropriate along the way.



Connection Manager Administration Kit Wizard

Custom Actions

You can specify programs to start automatically before, during, or after the connection to your service.

Custom actions of the same type are run in the order in which they appear in the list. To view custom actions of a specific type, select the action type from the drop-down list.

Action type: <All>

Custom actions:

Description	Action Type

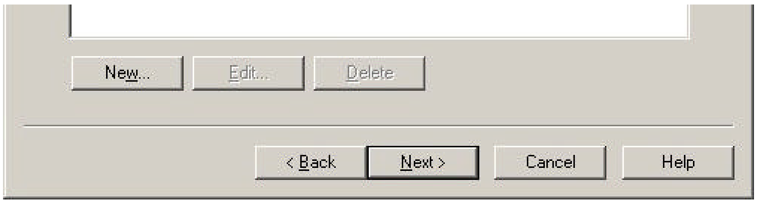
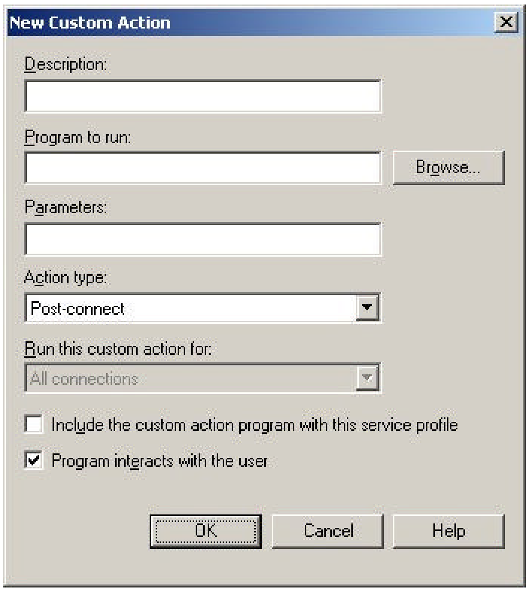


Figure 1: The Custom Actions screen of the CMAK Wizard

2. Select Post-Connect from the Action type drop-down box, and then click the New button to add an action. The New Custom Action dialog box is displayed, as shown Figure 2.



Description:

Program to run:

Parameters:

Action type: Post-connect

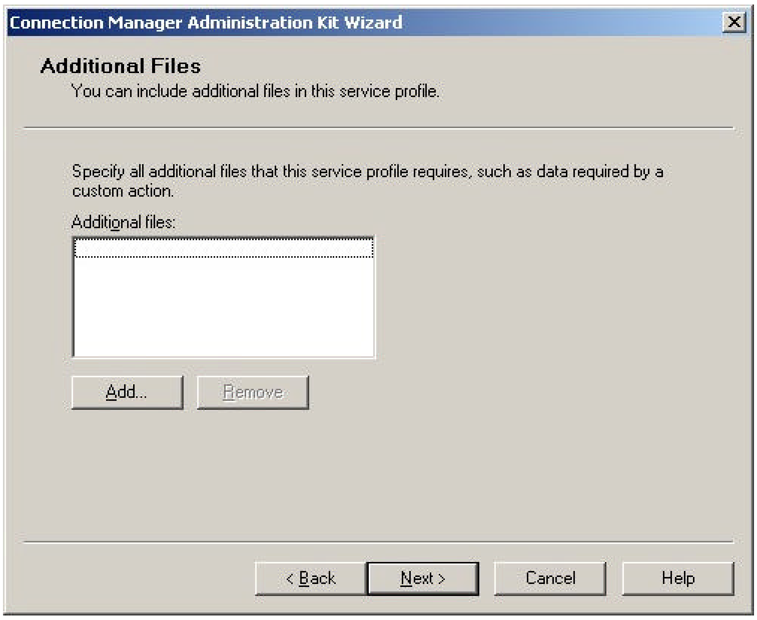
Run this custom action for: All connections

☐ Include the custom action program with this service profile

☑ Program interacts with the user

Figure 2: the New Custom Action dialog box

3. Type a descriptive title for the post-connection action in the Description box. In Program to run, enter the name of your baselining script. You can also use the Browse button to look for it. Type the command-line switches and their arguments in the Parameters box. Finally, check the two bottom boxes, "Include the custom action program with this service profile" and "Program interacts with the user."
4. Click OK, and you should return to the Custom Actions screen. Click Next. Continue filling in the wizard screens as appropriate, until you come to the Additional Files screen, depicted in Figure 3.



Additional Files

You can include additional files in this service profile.

Specify all additional files that this service profile requires, such as data required by a custom action.

Additional files:

Figure 3: the CMAK wizard Additional Files screen

5. Click Add, and then enter RQC.EXE in the dialog presented next. You can use the Browse button to search for it graphically. Once you're finished, click OK. You'll be returned to the Additional Files screen, where you'll see RQC.EXE listed. Click Next.
6. Complete the remainder of the wizard as appropriate.

Next Time

In the next installment of this article, I'll look at distributing this new profile to remote users, configuring the policy that actually performs the quarantining, how to except users from certain quarantine configurations, and how this technology is implemented in Microsoft's new ISA Server 2004. Stay tuned.

About the author

Jonathan Hassell is an author and consultant specializing in Windows administration and security. He is the author of Managing Windows Server 2003 and RADIUS, both published by O'Reilly & Associates, and Hardening Windows , published by Apress. He also holds periodic public seminars; see www.hardeningwin.com for details. He has written for Windows & .NET Magazine and WindowsITSecurity.COM and is a contributor to PC Pro, a leading computer magazine in the United Kingdom.