

The network interoperability compatibility problem, second follow-up

devblogs.microsoft.com/oldnewthing/20070201-03

February 1, 2007



Raymond Chen

I post this entry with great reluctance, because I can feel the heat from the pilot lights of the flame throwers all the way from here. The struggle with the network interoperability problem continued for several months after I brought up the topic. In that time, a significant number of network attached storage devices were found that did not implement “fast mode” queries correctly. (Buried in this query are some of them; there are others.) Some of them were Samba-based whose vendors did not have an upgrade available that fixed the bug. But many of them used custom implementations of CIFS; consequently, any Samba-specific solutions would not have helped those devices. (Most of the auto-detection suggestions people proposed addressed only the Samba scenario. Those non-Samba devices would still not have worked.) Even worse, most of the devices are low-cost solutions which aren’t firmware-upgradable or have any vendor support. Some of the reports came from people running fully-patched well-known Linux distributions. So much for being in all the new commercially supported offerings over the next couple months. Furthermore, those buggy non-Samba implementations mishandled fast mode queries in different ways. For example, one of them I was asked to look at didn’t return any error codes at all. It just returned garbage data (most noticeably, corrupting the file name by deleting the first five characters). How do you detect that this has happened? If the server reports “I have a file called `e.txt`”, is Windows supposed to say, “Oh, I don’t think so. I bet you’re one of those buggy servers that chops off the first five letters of file names and that you really meant to say (scrunches forehead in concentration) `readme.txt`”? What if you really had a file called `e.txt`? What if the server said, “This directory has two files, `1.txt` and `2.txt`”? Is this a buggy server? Maybe the files are really `abcde1.txt` and `defgh2.txt`, or maybe the server wasn’t lying and the files really are `1.txt` and `2.txt`. One device simply crashed if asked to perform a fast mode query. Another wedged up and had to be reset. “Oh, looks like somebody brought their Vista laptop from home and plugged it into the corporate network. Our document server crashed again.” Given the much broader ways that servers mishandled fast queries, any attempt at auto-detecting them will necessarily be incomplete and fail to detect broken servers. This is fundamentally the case for servers which return perfectly formed, but incorrect, data. And even if the detection were perfect, if it left the server in a crashed or hung state, that wouldn’t

be much consolation. Given this new information, the solution that was settled on was simply to stop using “fast mode” queries for anything other than local devices. The most popular file system drivers for local devices (NTFS, FAT, CDFS, UDF) are all under Microsoft’s control and they have already been tested with fast mode queries. Such is the sad but all-too-true cost of interoperability and compatibility.

(To address other minor points: It’s not the case that the Vista developers “knew the [fast mode query] would break Samba-based devices since late 2005”. The fast mode query was added, and the incompatibility with Samba wasn’t discovered until March 2006. “Why didn’t you notify the Samba team?” Because by the time we found the problem, they had already fixed it.)

Raymond Chen

Follow

