

Re: Microsoft FTP Server problem on W2K?

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.inetserver.iis.security/2003-12/0358.html>

From: DavidM (*scandal_123_at_cox.net*)

Date: 12/12/03

Date: Fri, 12 Dec 2003 15:54:55 -0600

It is a UNISYS ClearPath mainframe system that is trying to FTP using passive mode to a MS FTP server.

Again, it has multiple NICs and appears to be load balancing the connections. Which, in most cases, probably works fine for most TCP/IP applications. However, FTP uses two ports, which kind of makes things a bit more complicated.

Currently the mainframe FTPs in ACTIVE mode. Everything works fine when it works.

The problem I've found is, sometimes the mainframe is just too bogged down with running nightly updates and the changing of job priorities, that it cannot service incoming requests. I.E., if I try to FTP to mainframe, it sometimes just sits there without any banner or logon prompt. Eventually the connection will time-out or work after 2-3 minutes of waiting.

Since the mainframe pushes files to our customers over a WAN connection, the goobs that designed our operational environment send 1.4GB transfers over slow WAN links that take 5-8 hours a day to FTP! Not only does this hurt the WAN and impact the mainframe all day long with this active connection... when there is a communications problem, we have to start all over.

I now have them push the FTP files to a local Microsoft FTP server over a 10mb half-duplex link. Unfortunately, a 100mb full duplex card costs about \$30K! Arrghhh The clients then come into our FTP server to grab the files. Much more efficient.

The thing I'm fighting now is every once in a while, FTPs from Mainframe fail. The last time this happened, the operators tried to restart the mainframe jobs with no luck. They then went to W2K server and rebooted. They tried restarting job and it mysteriously worked.

I believe, however, that the problem is/was with ACTIVE connections. Since mainframe pushes file and is bogged down, I believe the FTP server could not connect back within a reasonable amount of time to establish the data connection.

By the time they called someone to recommend they reboot NT and tried it again, it worked. Although the mainframe gurus are blaming NT. Which I totally disagree with their assessment—especially when response is clearly horrible during our production hours. I had to explain why PASV is better than ACTIVE to the same mainframe guys. No clue.

I made the suggestion, for sake of efficiency, that they begin sending files to NT via PASV mode. This way NT won't have to connect back to mainframe and I assume the mainframe's outgoing connections will be faster than waiting for an incoming.

This is where we found out that the load balancing performed on the mainframe was the culprit. Although again, the mainframe gods refuse to believe their legacy system is the cause of our problems. These are the same goobs that decided to load balance across 3 NICs and not even consider isolating production traffic from client traffic. I.E., clients use one NIC for online activity, etc., and internally we use a production NIC for outgoing files and backend communication for our NT front-end systems.

They just implemented a firewall and we have over 500 FTPs going to/from mainframe, they just willy-nilly decided they're create a two-way rule for ports 20/21 for FTP. However, since we send more files out than we receive in, we could have simply sent all our files PASV and then there would have been no firewall change necessary on our end, except to those clients that have firewalls themselves. Which most do not.

It's crazy!

If you wish to reply to me personally, please remove the "underline" from scandal_123@cox.net. The is done to avoid SPAM!

"Alun Jones [MS MVP]" <alun@taxis.com> wrote in message news:sLpCb.789\$UI.289@newssvr22.news.prodigy.com...

> *In article <O2PGLCNwDHA.2000@TK2MSFTNGP11.phx.gbl>, "DavidM"*

> *<scandal_123@cox.net> wrote:*

> *>The below excerpts are what the mainframe vendor is telling us:*

> >

> *>Email #1 from us to vendor:*

> *>The windows FTP server does not like the mainframe sending packets from*

> *>multiple IP addresses. When we restricted the packets to come from only one*

> *>IP address, passive mode worked.*

> >

> *>So the question is, should packets be allowed to come from multiple IP*

> *>addresses when using Passive Mode. Bug or feature. A good question for*

> *>engineering. Can you follow up on this??*

>

> *Bug. Definitely. In the case that a PASV mode transfer is coming from a*

> *client (rather than another server, as in proxy transfer), the IP address*

> *(and technically, the port) the client binds to must match that address*

(and

> port) currently in use on the control connection. The port binding is
> parenthesised, because a) nobody does it and b) it would lead to problems
> with TIME_WAIT state when transferring large numbers of files in short
time
> (four minutes or less) through a restrictive port range (such as a
firewall
> might provide).
>
> >Email #2 from vendor to us:
> >
> >The following is from engineering:
> >1. Any given FTP connection – in fact, any TCP/IP connection – will
involve
> >packets being sent from a single IP address and to a single IP address.
The
> >TCP/IP connection is identified by source and destination port numbers
and
> >source and destination IP addresses.
>
> Yes – statement of fact.
>
> >2. The FTP standard (RFC 1123 section 4.1.2.12) states that, on a
> >multi-homed server, the data connection MUST use the same IP address as
the
> >control connection. It is not, however, required that the client-side
data
> >connection use the same IP address as the client-side control connection.
In
> >fact, it's not even necessary, according to RFC 959, that the client ends
of
> >the two connections be in the same machine. See RFC 959 section 2.3.
>
> Ah, but section 3.3 of RFC 959 says that there is a default port,
> client-side, that must be used unless the PORT command is used to
negotiate
> a different one. Since PASV voids PORT, the client side must use the
> default port (and hence, the IP address that it used to connect to the
> server from). As it turns out, few clients do this, but all clients
should
> use the same IP address.
>
> Also, check out section 4 of RFC 2577, "FTP Security Considerations" –
> essentially, your vendor is saying that there is a random chance that
their
> client won't work when connecting to a server that chooses to check for
the
> same IP address in control and data connections.
>
> Note that the random chance of _failure_ is 1/2 for a two-homed client,
2/3
> for a three-homed client, 3/4 for a four-homed client, etc. In other

words,

> *the chances of failure go up the more network cards you have in the machine,*

> *and your chance of success is never better than fifty-fifty. What poor odds. Such an unreliable client. Care to name it, so we can all avoid it?*

>

> >3. *The mainframe FTP behaves in this way:*

> >

> >*In PORT mode, where the data connection establishment is FROM server TO client, our client establishes the data connection from the same IP address*

> >*as the data connection.*

>

> *Yes, this makes sense, for security's sake as much as for consistency.*

>

> >*In PASV mode, where the data connection establishment is FROM client TO server, our client allows TCP/IP to assign the IP address of the data connection. It is possible that this could be different from the IP address*

> >*of the control connection.*

>

> *This is bogus. It takes what, two, three lines of code to call getsockname*

> *and bind? The IP address used to connect to the PASV socket should be the same as used to connect to the control connection, if it comes from the same*

> *machine. Otherwise, the server is free to decide that "this is not the same*

> *client", and reject the attempt to hijack the port.*

>

> >*The following is the result of the analysis of the information you provided:*

> >*We analyzed the trace that was provided and it shows, we initiated the control connection from 10.246.1.12, and the data connection from 10.246.1.11. The data connection opened successfully, which suggests that*

> >*the remote FTP server was, at least at a TCP level, prepared to accept the*

> >*connection initiation from the second IP address. The server returned a 426*

> >*error response when we sent the STOR command over the control connection, and immediately closed the data connection.*

> >

> >*As I said previously, I believe that the behavior of our client is in accordance with the FTP protocol standard.*

>

> *So is a client that refuses to allow the user to transfer any files at all.*

>

> *At some stage, you have to say "okay, we want to make sure that our*

software

> *not only obeys the standard, but also has some useful functionality".*

>

> *>We still believe the problem is on the remote FTP server side.*

>

> *Bull–pucky. The FTP server is behaving slightly more securely than RFC 959*

> */ 1123 dictates, in a manner compliant with RFC 2577. The FTP client is*

> *behaving like its programmers have got their heads up their collective arses*

> *and haven't been aware of the last decade's developments in FTP.*

>

> *>Can someone please tell me what the correct answer is. I would imagine that*

> *>the FTP server cannot accept passive connections from two different IP*

> *>addresses. Otherwise, FTP transmissions would be failing for no apparent*

> *>reasons due to port scans, hackers, etc.*

>

> *The FTP server _could_, if it wanted to, accept connections to PASV*

> *listeners from any old IP address. But, this would open your server up to*

> *port hijacking attacks. See RFC 2577. This FTP server is configured or*

> *programmed to prevent such attacks by requiring that both control and data*

> *connections must come from the same port. This is how our own FTP*

servers,

> *WFTPD and WFTPD Pro, are configured by default, and it's a recommendation of*

> *RFC 2577, which only covers the most basic and vital of security*

> *considerations. [Its only concession to encryption, for instance, is to*

> *state "To guarantee the privacy of the information FTP transmits, a strong*

> *encryption scheme should be used whenever possible."]*

>

> *The client, and its vendor, are being stubborn–headed in insisting that*

> *they're RFC compliant. For that matter, so's a brick – it refuses all*

> *access. That doesn't make it useful. Your vendor doesn't want to add the*

> *few lines that it takes to make their client more useful in the days of an*

> *unsecured Internet. Dump your client vendor and go with one that gives*

even

> *the slightest damn about security.*

>

> *>Any comments, feedback, etc. appreciated.*

>

> *I like RFC 2577. It has my name in it.*

>

> *>Perhaps THIS IS A Microsoft bug???*

>

> *No, it's a rare occasion of Microsoft being more secure than is desired by*

> *your vendor.*

>

> *Alun.*

> *~~~~~*

>

microsoft.public.inetserver.iis.security: Re: Microsoft FTP Server problem on W2K?

- > *[Please don't email posters, if a Usenet response is appropriate.]*
- > --
- > *Texas Imperial Software | Find us at <http://www.wftpd.com> or email*
- > *1602 Harvest Moon Place | alun@texas.com.*
- > *Cedar Park TX 78613-1419 | WFTPD, WFTPD Pro are Windows FTP servers.*
- > *Fax/Voice +1(512)258-9858 | Try our NEW client software, WFTPD Explorer.*