

Re: ssh hangs when restoring

Source: <http://www.derkeiler.com/Newsgroups/comp.security.ssh/2003-07/0363.html>

From: Richard E. Silverman (*res_at_goxp.net*)

Date: 07/28/03

Date: 28 Jul 2003 16:15:22 -0400

>>>> "NE" == Nicolas Ecartot <nicolas.ecartot@alussinan.org> writes:

NE> Hello, I'm using dump and restore to backups my systems. I dump a
NE> filesystem on host1 in a file /tmp/myFile.dump with the command :
NE> dump -af /tmp/myFile.dump /var (for example)

NE> Then, I try to restore it on another host (host2). From host2, I
NE> use :

NE> ssh someUser@host1 cat /tmp/myFile.dump | restore -tvf -

NE> All this works, restore actually displays the listing, but at the
NE> end, it hangs.

NE> Some people told me the problem came from ssh not closing a
NE> 'buffer', and restore waiting ssh to close this stream.

The problem these people are alluding to is probably this one:

<http://www.snailbook.com/faq/background-jobs.auto.html>

... however, that's not the problem you have here.

The first thing to notice, is that the command you've chosen is not a good way to accomplish the remote dump listing. This:

NE> ssh someUser@host1 cat /tmp/myFile.dump | restore -tvf -

causes the **entire** dump file — perhaps gigabytes of data — to be transmitted over the network link back to the client host, where it is fed to "restore" in order to produce the file listing. If this isn't clear, think for a moment about the way the local shell is interpreting your command; the pipe is done on the client host, not on the server.

It would be much more efficient and faster to run the listing on the other side, and only send that back, since that's all you care about:

comp.security.ssh: Re: ssh hangs when restoring

```
ssh someUser@host1 restore -tvf /tmp/myFile.dump
```

Now, the reason why your version is hanging has to do with the behavior of "restore", and your local shell. Your solution makes the assumption that "restore" will read to end of file — but it doesn't. Since you've told it to only do a listing, not an actual restore, it doesn't need all the data. It reads enough of the file (stdin in this case) to be able to complete the listing, and then exits. This leaves the entire contents of the last file in the dump still waiting to cross the SSH connection. The last piece of that path is a pipe from the SSH client to "restore", created by the shell in response to your pipe command. When "restore" exists, pipe does **not** get closed — because the shell still has an open file descriptor referencing that pipe, and pipes do not return eof until all fd's that could read from the other side, are closed. So, the pipe blocks the flow of data indefinitely.

You can demonstrate this by modifying your command like so:

```
ssh someUser@host1 cat /tmp/myFile.dump | (restore -tvf -; cat > /dev/null)
```

This transforms the last command into a sink — it will read to eof, because "cat" will take over after "restore" exits and read the rest of stdin. This command will exit, although it may appear to hang for some time if the last dump file is large, since it has to wait for all that data to be transmitted over SSH.

```
--  
Richard Silverman  
res@qoxp.net
```