Subject: Re: Network namespaces a path to mergable code.
Posted by ebiederm on Wed, 28 Jun 2006 14:03:41 GMT
View Forum Message <> Reply to Message

Cedric Le Goater <clg@fr.ibm.com> writes:

> Eric W. Biederman wrote:
>
>> Despite what it might look like unix domain sockets do not live in the
>> filesystem. They store a cookie in the filesystem that roughly
>> corresponds to the port number of an AF_INET socket. When you open a
>> socket the lookup is done by the cookie retrieved from the filesystem.
>
> unix domain socket lookup uses a path_lookup for sockets in the filesystem
> namespace and a find_by_name for socket in the abstract namespace.

Right. And the abstract namespace does nothing with the current
filesystem.

>> So except for their cookies unix domain sockets are always in the
>> network stack.
>
> what is that cookie ? the file dentry and mnt ref ?

The socket entry in the filesystem but really the socket
inode number in that entry. This entry has nothing to with dentry's
or mount refs so if I read the correctly every path to that socket
should yield the same entry.

> so, ok, the resulting struct sock is part of the network namespace but
> there is a bridge with the filesystem namespace which does not prevent
> other namespaces to do a lookup. the lookup routine needs to be changed,
> this is any way necessary for the abstract namespace.

Yep.

> I think we're reaching the limits of namespaces. It would be much easier
> with a container id in each kernel object we want to isolate.

Nope. Except for the fact that names are peculiar (sockets, network
device names, IP address, routes...) the network stack splits quite cleanly.

I did all of this in a proof of concept mode several months ago and
the code is still sitting in my git tree on kernel.org. I even got
the generic stack reference counting fixed.

Eric