Subject: Re: Network namespaces a path to mergable code.
Posted by ebiederm on Wed, 28 Jun 2006 06:55:25 GMT
View Forum Message <> Reply to Message

Sam Vilain <sam@vilain.net> writes:

> Eric W. Biederman wrote:
>> In general it is possible to get file descriptors opened by someone
>> else because unix domain sockets allow file descriptor passing. Similarly
>> I think there are cases in both unshare and fork that allows you to sockets
>> open before you entered a namespace.
>>
>
> This is an interesting point; it is known to be possible to do this on a
> traditional system, because with a Unix Domain socket, the other end is
> always in the same Unix Domain.
>
> However what we're doing is saying that, well, the other end of the
> socket might not be in the same Unix Domain. In fact, we've already
> smashed to pieces this monolithic concept of a Unix Domain, to the point
> where the other end might be in a different network domain, but is in
> the same filesystem domain, for instance. Does it get to pass file
> descriptors through?

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.
So except for their cookies unix domain sockets are always in the
network stack.

Which means it is a royal pain to create a unix domain socket between
namespaces. Which is the generally desired behavior.

> We would appear to be stretching the definition of "Unix Domain"
> somewhat if we allow these sockets to exist between network namespaces.
> Maybe it doesn't matter; this is just a VFS namespace feature/caveat.

Unless I am mistaken this is something that can only be created (given
my describe semantics) when you create the container. So if you want
it you got it but you can't create it if you never had it.

Eric