
Subject: Re: [RFC PATCH 0/5] Resend - Use procs to change a syscall behavior
Posted by [Dave Hansen](#) on Thu, 10 Jul 2008 19:04:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Thu, 2008-07-10 at 20:45 +0200, Pavel Machek wrote:

> On Thu 2008-07-10 10:53:35, Dave Hansen wrote:

> > On Thu, 2008-07-10 at 10:54 +0200, Pavel Machek wrote:

> > >

> > > If you don't see a backward compatibility problem here, perhaps you
> > > should not be hacking kernel...? The way ids are assigned is certainly
> > > part of syscall semantics (applications rely on), at least for open.

> >

> > We also used to have a pretty defined ordering for handing out address
> > space with mmap(). That all changed with address space randomization.

> > Are file descriptors different somehow?

> >

> > Anyway, it's not like we're actually changing existing behavior. An
> > application has to do something special and new to trigger this new
> > behavior. Nobody is going to stumble over it, and it will *not* break
> > backward compatibility.

>

> It will break compatibility, but not in a way you expect. There's
> application called "subterfuge" that monitors other applications
> using ptrace and enforces security policy (or does other stuff). Such
> hacks depend on existing syscalls behaving in a way they are
> specified...

>

> Then you'll have to update open.2 man page:

>

> DESCRIPTION

> Given a pathname for a file, open() returns a file descriptor,
> a small, non-

> negative integer for use in subsequent system calls

> (read(2), write(2),

> lseek(2), fcntl(2), etc.). The file descriptor returned by

> a successful

> call will be the lowest-numbered file descriptor not currently

> open for the

> process.

>

> ...you'll need to add "unless someone write some number in file in

> /proc somewhere"... hmm... is new behaviour even POSIX compliant?

> open() is specified in POSIX...

Yup, that's true. Good point.

> Ok, so it will not break too many apps... but echo "123 >

> /proc/something" breaking bash (etc) is not nice.

>
> (Plus proposed interface is so ugly that this discussion is moot.)

Yes, I agree that the current proposed interface is too ugly to live. :)

-- Dave

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
