Subject: Re: [PATCH 4/9] namespaces: utsname: switch to using uts namespaces
Posted by rdunlap on Fri, 19 May 2006 02:42:50 GMT
View Forum Message <> Reply to Message

On Thu, 18 May 2006 21:21:14 -0500 Serge E. Hallyn wrote:

> Quoting Randy.Dunlap (rdunlap@xenotime.net):
> > > --- a/arch/i386/kernel/sys_i386.c
> > > +++ b/arch/i386/kernel/sys_i386.c
> > > @@ -210,7 +210,7 @@ asmlinkage int sys_uname(struct old_utsn
> > > if (!name)
> > > return -EFAULT;
> > > down_read(&uts_sem);
> > > - err=copy_to_user(name, &system_utsname, sizeof (*name));
> > > + err=copy_to_user(name, utsname(), sizeof (*name));
> >
> > It would be really nice if you would fix spacing while you are here,
> > like a space a each side of '='.
> >
> > and a space after ',' in the function calls below.
>
> Ok. Then in blocks like the following:
>
> > > - error =
__copy_to_user(&name->sysname,&system_utsname.sysname,__OLD_UTS_LEN);
> > > + error = __copy_to_user(&name->sysname,&utsname()->sysname,__OLD_UTS_LEN);
> > > error |= __put_user(0,name->sysname+__OLD_UTS_LEN);
> > > - error |=
__copy_to_user(&name->nodename,&system_utsname.nodename,__OLD_UTS_LEN);
> > > + error |= __copy_to_user(&name->nodename,&utsname()->nodename,__OLD_UTS_LEN);
> > > error |= __put_user(0,name->nodename+__OLD_UTS_LEN);
> > > - error |= __copy_to_user(&name->release,&system_utsname.release,__OLD_UTS_LEN);
> > > + error |= __copy_to_user(&name->release,&utsname()->release,__OLD_UTS_LEN);
> > > error |= __put_user(0,name->release+__OLD_UTS_LEN);
> > > - error |= __copy_to_user(&name->version,&system_utsname.version,__OLD_UTS_LEN);
> > > + error |= __copy_to_user(&name->version,&utsname()->version,__OLD_UTS_LEN);
> > > error |= __put_user(0,name->version+__OLD_UTS_LEN);
> > > - error |=
__copy_to_user(&name->machine,&system_utsname.machine,__OLD_UTS_LEN);
> > > + error |= __copy_to_user(&name->machine,&utsname()->machine,__OLD_UTS_LEN);
> > > error |= __put_user(0,name->machine+__OLD_UTS_LEN);
>
> Should I leave it as is, to keep the consistent look? Change just the
> lines I'm editing, making it inconsistent? Or change the whole block,
> making my patch seem a bit larger than it really is, but giving the
> nicest end result?

I'd go for the latter, along with my other comment of breaking them

to fit into 80 columns also.

> I suppose I could insert a separate patchset fixing up the spacing in
> those blocks but making no real changes at all, then apply my patch on
> top of that...?
>
> > > --- a/arch/mips/kernel/syscall.c
> > > +++ b/arch/mips/kernel/syscall.c
> > > @@ -232,7 +232,7 @@ out:
> > > */
> > > asmlinkage int sys_uname(struct old_utsname __user * name)
> > > {
> > > - if (name && !copy_to_user(name, &system_utsname, sizeof (*name)))
> > > + if (name && !copy_to_user(name, utsname(), sizeof (*name)))
> >
> >
> > OK, here's my big comment/question. I want to see <nodename> increased to
> > 256 bytes (per current POSIX), so each field of struct <variant>_utsname
> > needs be copied individually (I think) instead of doing a single
> > struct copy.
> >
> > I've been working on this for the past few weeks (among other
> > things). Sorry about the timing.
> > I could send patches for this against mainline in a few days,
> > but I'll be glad to listen to how it would be easiest for all of us
> > to handle.
> >
> > I'm probably a little over half done with my patches.
> > They will end up adding a lib/utsname.c that has functions for:
> > put_oldold_unmame() // to user
> > put_old_uname() // to user
> > put_new_uname() // to user
> > put_posix_uname() // to user
>
> Ok, so long as these functions accept a utsname, we should be able to
> just change what we pass in to these functions to being the namespace's
> utsname, right? Or am I missing the really nasty part?

The nodename field changes from 65 chars (struct new_utsname) to 256 chars
(struct posix_utsname), and nodename is not the final field in the
struct, so it's no longer safe to do a simple struct copy. Each
field in the struct needs to be copied individually if the target is
not a struct posix_utsname. It's not rocket science.

~Randy