Subject: Re: [PATCH 3/7] uts namespaces: use init_utsname when appropriate
Posted by serue on Sat, 08 Apr 2006 20:27:01 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> > diff --git a/include/asm-i386/elf.h b/include/asm-i386/elf.h
> > index 4153d80..8d455e2 100644
> > --- a/include/asm-i386/elf.h
> > +++ b/include/asm-i386/elf.h
> > @@ -108,7 +108,7 @@ typedef struct user_fxsr_struct elf_fpxr
> > For the moment, we have only optimizations for the Intel generations,
> > but that could change... */
> >
> > -#define ELF_PLATFORM (system_utsname.machine)
> > +#define ELF_PLATFORM (init_utsname()->machine)
> >
> > #ifdef __KERNEL__
> > #define SET_PERSONALITY(ex, ibcs2) do { } while (0)
>
> I think this one needs to be utsname()->machine.
>
> Currently it doesn't matter. But Herbert has expressed
> the desire to make a machine appear like an older one.

Ok.

> > diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
> > index cb8a92f..81db372 100644
...
> > @@ -1479,11 +1479,11 @@ static int __init ip_auto_config_setup(c
> > case 4:
> > if ((dp = strchr(ip, '.'))) {
> > *dp++ = '\0';
> > - strlcpy(system_utsname.domainname, dp,
> > - sizeof(system_utsname.domainname));
> > + strlcpy(init_utsname()->domainname, dp,
> > + sizeof(init_utsname()->domainname));
> > }
> > - strlcpy(system_utsname.nodename, ip,
> > - sizeof(system_utsname.nodename));
> > + strlcpy(init_utsname()->nodename, ip,
> > + sizeof(init_utsname()->nodename));
> > ic_host_name_set = 1;
> > break;
> > case 5:
>

> This also probably makes sense as utsname(). It doesn't
> really matter as this is before init is executed. But logically
> this is a user space or per namespace action.

Right, I was kind of favoring using init_utsname() for anything
__init. But utsname() will of course work just as well there.

> > diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
> > index aa8965e..97c8439 100644
> > --- a/net/sunrpc/clnt.c
> > +++ b/net/sunrpc/clnt.c
> > @@ -176,10 +176,10 @@ rpc_new_client(struct rpc_xprt *xprt, ch
> > }
> >
> > /* save the nodename */
> > - clnt->cl_nodelen = strlen(system_utsname.nodename);
> > + clnt->cl_nodelen = strlen(init_utsname()->nodename);
> > if (clnt->cl_nodelen > UNX_MAXNODENAME)
> > clnt->cl_nodelen = UNX_MAXNODENAME;
> > - memcpy(clnt->cl_nodename, system_utsname.nodename, clnt->cl_nodelen);
> > + memcpy(clnt->cl_nodename, init_utsname()->nodename, clnt->cl_nodelen);
> > return clnt;
> >
> > out_no_auth:
>
> Using nodename is practically the definition of something
> that should per namespace I think. Plus it would be really inconsistent
> to use utsname() and the init_utsname for the nfs rpc calls.
>
> Unless I am missing something.

It seemed like this would be happening in any old context, so that
current->uts_ns could be any process'. Tracing it back further,
it seems like nfs+lockd should have the context available. So I'll
switch this as well.

thanks,
-serge